

# **Project Proposal: IMDb Anime Dataset Analysis Platform**

## **Pitch**

Our project aims to create an interactive web platform for exploring and analyzing the IMDb Anime Dataset. This platform will provide users with valuable insights into Japanese anime, including ratings, rankings, genres, and more. By simplifying the technical aspects and presenting the data in a user-friendly manner, we will cater to a wide audience of anime enthusiasts, researchers, and industry professionals. Our platform will offer a unique opportunity to dive deep into the world of Japanese animation.

## **Functionality**

The platform will offer the following key functionalities:

1. Browse Anime Titles: Users can search and browse a vast collection of Japanese anime titles from the IMDb dataset.
2. View Anime Details: Detailed information about each anime title, including user ratings, number of votes, runtime, year of release, and more, will be available.
3. User account: user can create an account and password to access the information.
4. Genre Analysis: Users can explore the popularity of different anime genres based on user ratings and the number of votes.
5. Predictive Modelling: Machine learning algorithms will predict the rating and financial success (gross earnings) of an anime based on various features such as genre, runtime, and stars that users provide.
6. Top popular: visualize the TOP 20 animes based on the Number of Votes or User Rating that users could choose to see.
7. Data Visualization: Create interactive dashboards for data visualization, allowing users to filter data by criteria such as genre, year, and rating.

## **Components**

The project will consist of the following major components:

- 1) Frontend: The frontend will be developed using modern web technologies, including HTML, CSS, JavaScript, and React.js. This component will handle user interactions and data presentation.

- 2) Backend: The backend, built using Python and the Flask framework, will manage user requests, data processing, and predictive modeling.
- 3) Database: We will use a relational database, such as PostgreSQL, to store the IMDb Anime Dataset and user-generated data, enabling efficient data retrieval.
- 4) Data Processing: Python scripts will process and clean the IMDb dataset, preparing it for analysis and visualization.
- 5) Machine Learning: Machine learning models will be developed using scikit-learn to predict anime ratings and financial success.
- 6) User Authentication: Implement user authentication to manage user profiles and personalize recommendations.
- 7) APIs: Create APIs for frontend-backend communication to enable data exchange and predictions.
- 8) Testing Methodology: Unit testing will ensure the proper behavior of backend functionalities, while integration testing will validate interactions between the frontend and backend components.
- 9) Continuous Integration: Configure GitHub Actions to automate testing, code style checks, and deployment.

### **Interactions With Other Components**

- 1) The frontend will communicate with the backend through API endpoints for data retrieval and predictions.
- 2) Data processing scripts will update the database with new IMDb dataset entries.
- 3) Machine learning models will be integrated into the backend for predictive modeling.
- 4) User authentication will ensure secure access to personalized features.

### **Schedule**

Our project timeline is as follows:

Week 1-2: Project setup, database design, and data import.

Week 3-4: Frontend development and API design.

Week 5-6: Backend development, data processing, and predictive modeling.

Week 7-8: User authentication, and content recommendation.

Week 9-10: Data visualization, bug fixing, and final testing.

### **Possible Risks**

- 1) Data Updates: The IMDb dataset may change or receive updates, requiring regular data synchronization. **Solution:** we may consider adding some ways for users to help update the data for us.
  
- 2) User Data Security: Ensuring user data security and privacy is a critical concern that needs to be addressed comprehensively. **Solution:** This could be a potential risk. So, we try to just let users create an account without personal information.
  
- 3) Scalability: If the user base grows significantly, we may need to consider scaling the infrastructure. **Solution:** Because this risk may get out of our control, we will need to limit the number of users' accounts or ask help from professors or teaching assistants.
  
- 4) Complex User Interactions: Implementing advanced features like predictive modelling and recommendation systems could be complex and may require thorough testing. **Solution:** Because of the difficulty, we could work on this part earlier. Or we may ask help from class assistants to overcome the difficulties if we cannot complete.

### **Division of Work**

We will divide responsibilities among team members based on their expertise. Responsibilities will be assigned as follows:

Frontend Development: Wentao Zhang

Backend Development: Jiayin Meng

Database Management: Wentao Zhang

Data Processing: Minghao Shi

Machine Learning: Boqin Yuan

### **Teamwork**

We will establish a standard development environment setup, including the operating system, programming language versions, and library

dependencies. This will ensure that each team member is working in a similar development environment, reducing the likelihood of code working on one machine but not on another. This standardized environment can be achieved through virtual environments, containerization, or development environment setup scripts.

This approach makes sense for our team as we are all friendly and cooperative individuals who are not prone to arguments. Additionally, we are all open to receiving input from others and have a collaborative attitude, making it easier to adopt a common development environment. The use of a standardized environment will ensure that everyone has a similar foundation to work from, facilitating smoother collaboration and project delivery.

Regarding the division of project tasks, considering our team's friendly and cooperative nature, we plan to adopt a voluntary task allocation approach. This way, each team member can choose tasks based on their interests and expertise without feeling forced or unfairly treated. We will hold regular team meetings to discuss various aspects of the project and allow team members to voluntarily propose contributions. If someone is willing to take on a specific task, we will encourage and support them, rather than assigning tasks that may not align with their skills or interests.

### **Continuous Integration**

- 1) We will utilize pytest for our testing needs. This library provides a plethora of functionalities that are suitable for our Python-based backend.
- 2) We will adhere to the PEP 8 style guide for our Python code. This guide provides a detailed overview of the coding conventions for Python code.
- 3) For our frontend (JavaScript, CSS, HTML), we will follow the Airbnb Style Guide.
- 4) For Python, we will use flake8, a tool that will check our codebase against coding style.
- 5) For JavaScript, we will utilize ESLint with the Airbnb configuration.

- 6) To measure test coverage, we'll use the coverage.py tool. It provides a detailed report on which parts of the code are tested, enabling us to identify areas that require more thorough testing.
- 7) PRs will be opened once a feature is complete and tested locally. Reviewers will be rotated among team members to ensure diverse feedback. Every member will be assigned a specific PR to review each week. If the designated reviewer fails to review within 48 hours, the PR will be passed to the next available team member.
- 8) We will keep our branches up-to-date with the main branch. Regularly pulling from the main branch will help in identifying and resolving potential conflicts early. We will communicate effectively to ensure two team members are not working on the same functionality or piece of code simultaneously. In case of unavoidable merge conflicts, the team will collaborate to resolve them before merging the PR.
- 9) With these CI strategies in place, we aim to maintain a high standard of code quality, functionality, and teamwork. This will not only streamline our development process but also ensure the delivery of a robust IMDb Anime Dataset Analysis Platform.

## **Conclusion**

Our project aims to deliver an insightful and user-friendly platform for exploring and analyzing Japanese anime using the IMDb Anime Dataset. By breaking down the project into components, ensuring data security, and following a structured development schedule, we aim to provide a valuable resource for anime enthusiasts, researchers, and industry professionals. Continuous integration and regular updates will ensure the project's stability and usefulness over time.

Let's embark on this exciting journey to make Japanese anime data accessible and meaningful to all!