**Group 03 – Project 4 Write Up**

This analysis was conducted by Carlos Ruiz, Daniel Purrier, Steven Madden and Amar Patil. In it we analyze Anime Recommendation Database from the Kaggle Dataset, to provide a content based recommendation and visualization. Our goal with this dataset was to build an Anime recommender that can take into account viewer preferences of type and genre, returning new anime properties the user would enjoy.

We chose this dataset based on file size, number of rows, and perceived amount of cleaning necessary. We estimated we might only lose around 3.5% of rows when cleaning was done. We attempted to answer

.

1. Could we build the basic recommender?
2. Could we refine it based on genre and type preferences?
3. Could we then publish it as a usable web app with interactive data visualizations?

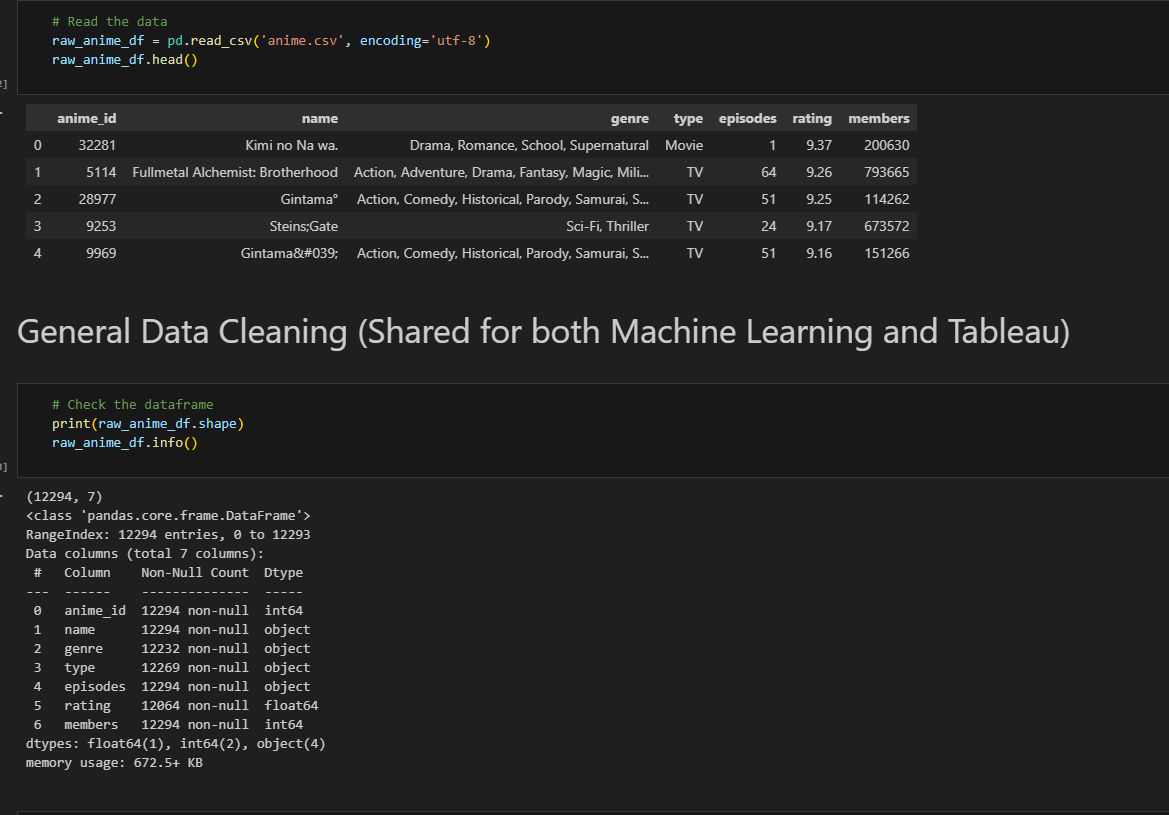
**Data Cleaning:**

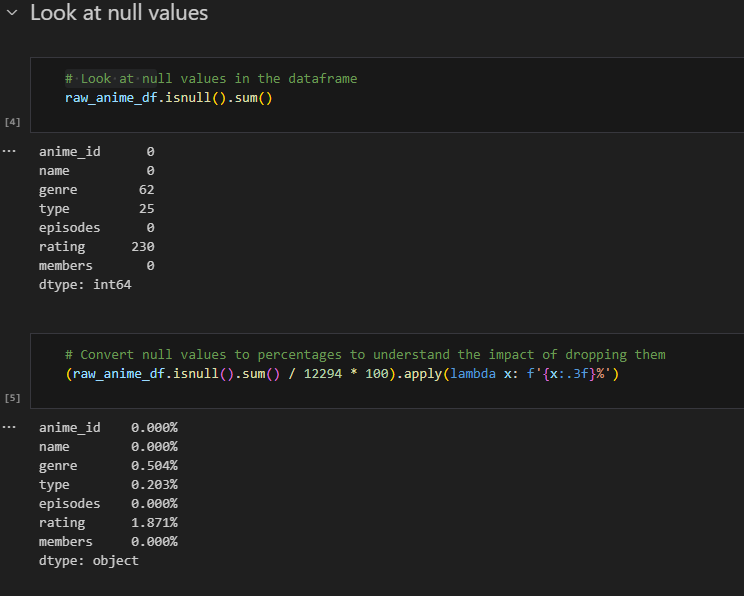
To proceed with answering these questions, we first needed to look at the dataset we are working with and modify it to better fit the questions that we are posing. Below is our raw data set, it has 8 columns and provides a good working amount of data. Reviewing the columns and their titles, we considered which columns would have the most impact in answering our questions, which columns need to be manipulated, and which can be dropped all together.

A screenshot of a computer

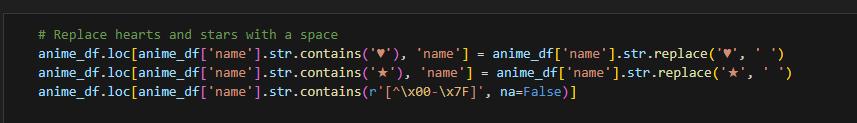
Description automatically generated

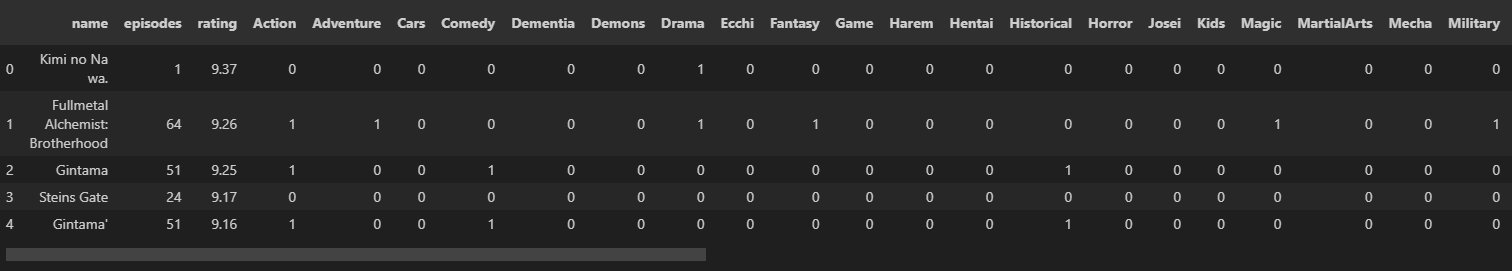
We started by reading in our dataset looking at its shape and searching for nulls. Since the impact of dropping null values is less than 1% in genre and type columns, we can drop all rows with null values. We lost another 2.5% when we found entries in the episode column containing non-numeric values.



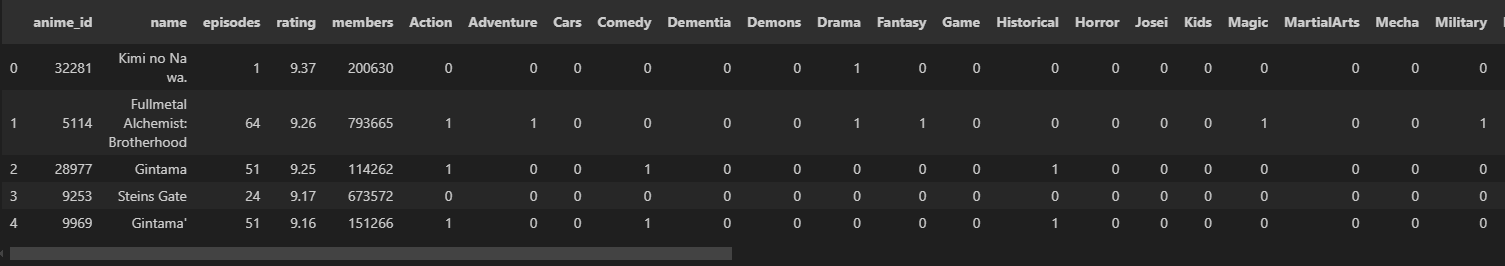


We found some strange symbols in some of the names, so we removed them. Then it was a matter of taking all the unique entries in the genre and type columns and turning them into their own Boolean columns, and then dropping the genre and type columns.



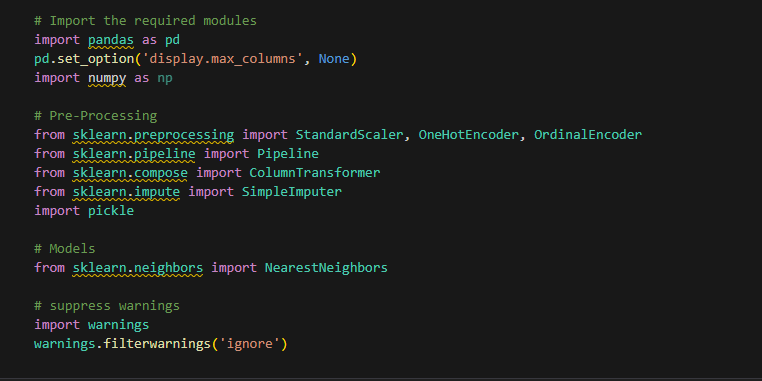


Once done, we dropped just a few more columns, when we realized several genre columns were of an 18+ plus variety, so we decided to drop them.



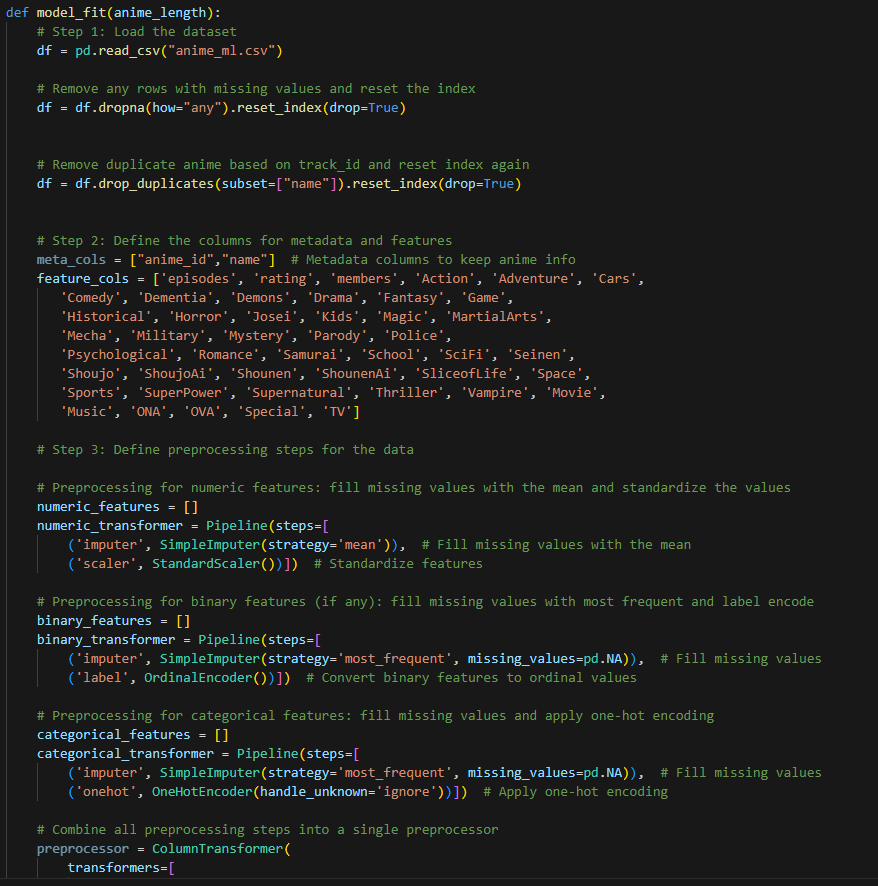
With that done, we were ready to start work

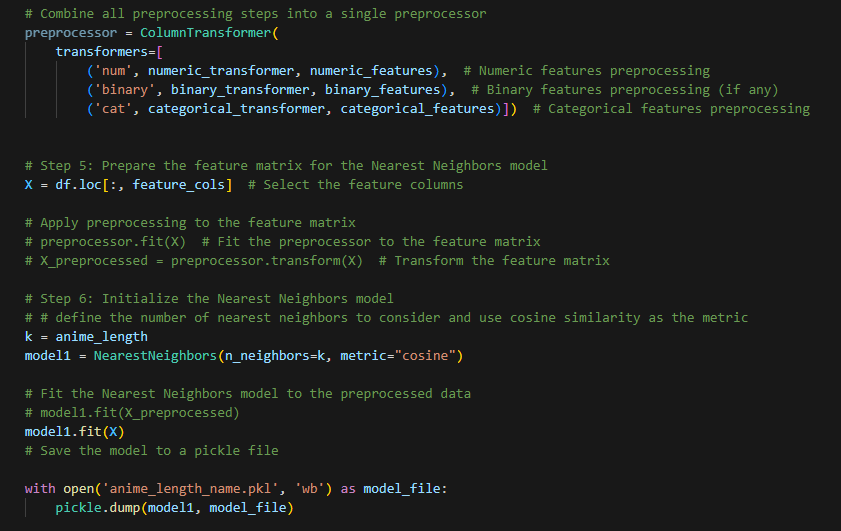
**Question 1: Could we build a basic recommender?**

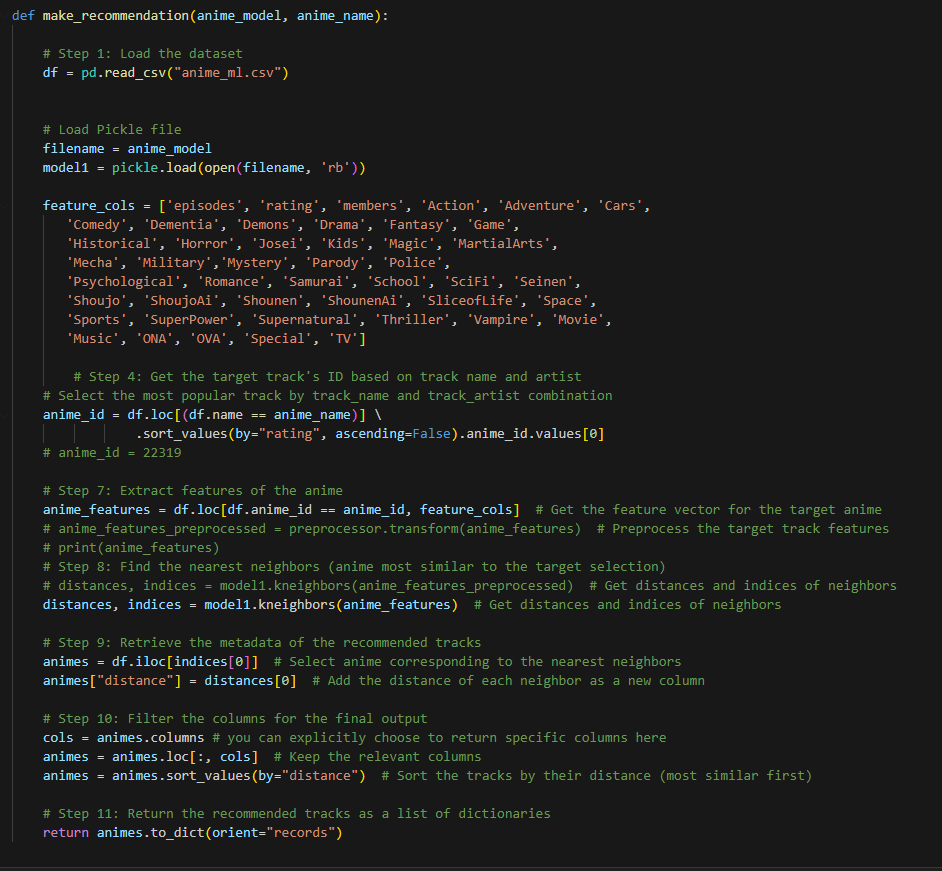
Utilizing jupyter notebook, we imported pandas, numpy, pickle, and scikit learn as sklearn (for processing)

.

Following an in class example we then built the recommender, first by programming the model\_fit: then the recommender itself.





Then the recommender itself:

**Recommender System:**

1. Candidate generation
   1. In this first state, the system starts from huge dataset and generates a smaller subset of data. Example in Youtube reduces large amount of videos down to hundreds or thousands. This is first stage of Recommendation. two common candidate generation approaches:
      1. Content-based Filtering
         1. Uses similarity between items to recommend items similar to what user likes
         2. Ex: if User watches dog videos, then the system can recommend cute animals videos to that user.
      2. Collaborative Filtering
         1. Recommends items based on the behavior or preferences of similar users.
         2. Ex: If used A is similar to user B, and user B likes video 1, then the system can recommend video 1 to user A.
      3. Hybrid Recommender
         1. Hybrid recommender systems combine collaborative filtering (analyzing user behavior and preferences) and content-based filtering (focusing on item characteristics) to provide more accurate and diverse recommendations.
2. Scoring
   1. This model scores and ranks the candidates in order to select the set of items (on the order of 10) to display to the user. This is subset of Candidate generation
3. Re-ranking
   1. This is final ranking which takes into account dislikes or likes of newer content. Re-ranking helps to ensure diversity, freshness and fairness.

To determine the degree of similarity, most recommendation systems rely on one or more of the following:

1. Manhattan

Also, know as city block distance, or taxicab geometry wherein the distance is measured between two data points in Grid-like path. As shown below. Mainly used where high dimensionality in the data.

A line drawn on a grid

Description automatically generated

1. Euclidean

This is the shortest distance between 2 data points in the plane. Mainly used for the smaller dimensionality problems.

A graph of a line with a pink pin

Description automatically generated

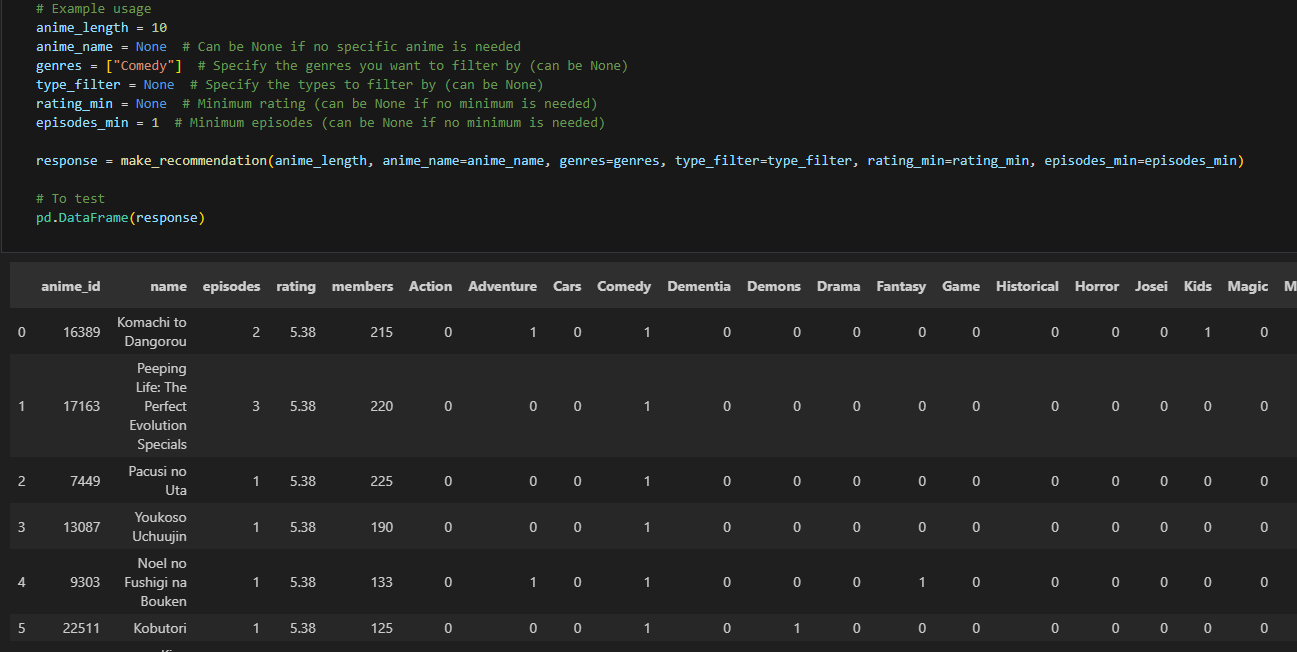
1. Cosine Distance

Also known as Cosine Similarity is used to find similarities between two data points. Cosine similarity is given by Cos θ, and cosine distance is 1- Cos θ. Mainly used in the Collaborative Filtering based recommender systems to offer future recommendations

A graph with a green arrow and a line

Description automatically generated

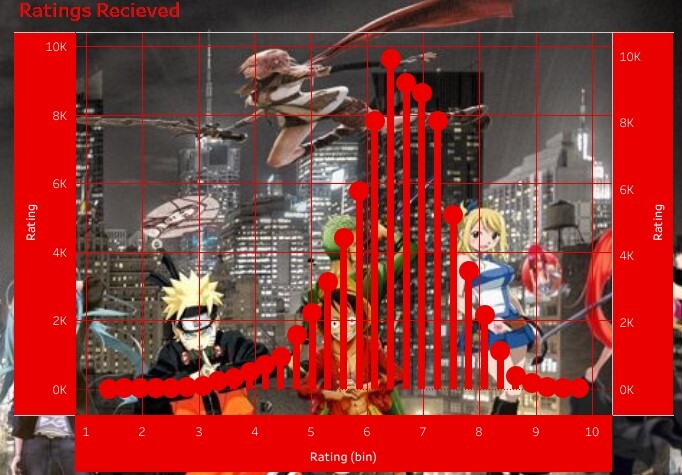
**Question 2:** Could we refine it based on genre and type preferences?

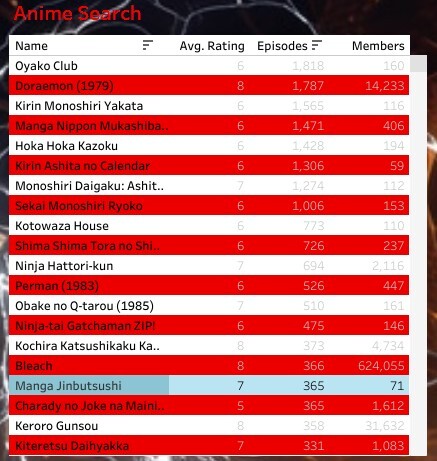
After the base code was working, several attempts were made to have the recommender offer suggestions via genre and type preferences. While the code worked within jupyter notebook, we reached a problem implementing it on the web app side.

It was decided we would try to implement the refined search through the web app itself.

**Question 3:** Could we publish it as a usable web app with interactive data visualizations?

Utilizing Tableau, we took our clean dataset and started building visualizations that could be interacted with. We wanted these visualizations to reference things in the data such as ratings of animes in the data set, the size of various genres and user feedback.



As well as having quick reference data an interested user might find helpful for their searches.

Then it was a matter of deploying all of it to the web app.

**Machine Learning Recommender Website**

Our group developed a comprehensive machine learning recommender website designed to provide list of anime recommendations based on a certain anime characters. The website is structured to offer insights into anime name, genre, ratings and episodes. The website has various sections, including a home page, interactive recommender dashboard, Tableau, an "About Us", page that introduces the team, proposal write-up page and a citation page that credits data sources used.

**Home Page**:  
 The home page sets the tone for the website by discussing

**Dashboard**:  
 The dashboard is dynamic and can be alter as per the user’s anime choice. First user selects anime name from the dropdown menu, this will trigger the ML recommender which is content based recommender to populate the list of anime names. Further the user can select additional filters like genre, type (movie, music, and TV etc.), ratings and episode.

**Tableau:**

Tableau was used to create a visualization story of the anime data. The original data did not require much cleaning, however there were a few explicit episode categories that were removed. Once cleaned multiple charts and visualization were created to explain the data. On the first page shows selections made by anime fans based off of genre. A tree map and a bubble chart were used to display this. A highlight chart can also be found here that showcases the name, rating, episode count, and members of the entire data set. This highlight chart can be searched by typing in the name of any anime a searcher can think of. Once located the highlight chart will display all findings related to the search. On page two is a highlight chart which will only display anime episodes that have a rating of 7 or greater, and can also be searched. The donut chart on this page shows the episode types of the data and finally, a lollipop chart depicts the ratings received for each episode. Multiple functions were used to create this story. The split function was used with the bubble chart as most of the episodes had multiple genre assigned to it in the original data and had to be split from them. The contain function was used to limit results when searching for anime with a rating higher than 7. The last function used, was needed to address the case sensitive issue while searching the highlight charts. Initially anime could only be located if each word type in began with a capital. After some searching it was discovered the regexp\_match function could fix this. Once implemented searches could now be completed without case sensitive issues.

**About Us Page**:  
 This page introduces the team members, providing a personal touch to the project. Each member's background is outlined, highlighting their expertise and contributions. This helps build credibility and showcases the diverse skill set within our team.

**Citation Page**:  
 The citation page is essential for maintaining academic and professional integrity. It lists all the data sources used in the project, ensuring transparency and giving credit to original authors and datasets. This section shows that the project is well-researched and relies on credible sources .

**Technical Aspects**:  
 The website likely utilizes a backend system, possibly built with Flask (as indicated using Python scripts like app.py and sqlHelper.py), to manage data retrieval and processing. The SQLite database and .csv serves as the data storage, housing information on anime, which is then used to generate the visualizations and data tables on the site.

The website is styled using Bootstrap, with the "Minty" theme providing a clean and modern look. The consistent use of this theme across all pages ensures a cohesive user experience. The use of JavaScript libraries like D3.js and Plotly.js for visualizations, along with DataTables for data management, adds a high level of interactivity to the site, allowing users to engage with the data meaningfully.

**Conclusion**:  
 The website created by our group is a well-rounded project that effectively combines data analysis, visualization, and user interaction to provide a comprehensive understanding of the pandemic's global impact. The website is not only informative but also accessible, thanks to its user-friendly design and interactive features. The inclusion of various sections, such as the tableau and dashboard, ensures that users can explore the data from multiple perspectives, making the project a valuable tool for anyone interested in high level of anime recommendation.

**Bias and Limitations**

When considering our dataset there are several biases and limitations that are found within that need to be considered when drawing conclusions. It is limited by the fact that this dataset lacks the most recent anime collection. It was last updated 8 years ago. Additionally, if a year or studio column were present, the team would have added additional layers of features to provide recommendations. And currently, you must no at least one anime for the recommender to work.

**Future Work**

Some examples of features the team would like to work on in the future are a further refining of the nearest neighbors to yield stronger results. Added search functionality with other columns, like year a project was released or the studio that produced it. Add the ability to show cover art of recommended animes. And further optimize the recommender for people who have no experience with anime

**Conclusion**

With more than two thirds of the population of the United States enjoying some kind of anime, a recommender may be a useful tool for finding new shows and movies to watch amid the wide variety available. Hopefully a project like this can take some of the guesswork out of finding new entertainment.

**Bibliography**

* [Anime Recommendations Database (kaggle.com)](https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database?select=anime.csv)  
    
  [Candidate generation overview  |  Machine Learning  |  Google for Developers](https://developers.google.com/machine-learning/recommendation/overview/candidate-generation)  
    
  [Hybrid Recommender Systems: Beginner's Guide (marketsy.ai)](https://marketsy.ai/blog/hybrid-recommender-systems-beginners-guide)
* <https://select2.org/data-sources/ajax>
* <https://makitweb.com/loading-data-remotely-in-select2-with-ajax/>
* [worldpopulationreview.com/country-rankings/anime-popularity-by-country](https://worldpopulationreview.com/country-rankings/anime-popularity-by-country)
* [git.bootcampcontent.com/boot-camp-consortium-east-coast/DATA-PT-EAST-APRIL-041524/-/tree/main/01-Lesson-Plans/23-Project-4-Week-1/3/BOOOTH\_RECOMMENDER\_EXAMPLE?ref\_type=heads](http://git.bootcampcontent.com/boot-camp-consortium-east-coast/DATA-PT-EAST-APRIL-041524/-/tree/main/01-Lesson-Plans/23-Project-4-Week-1/3/BOOOTH_RECOMMENDER_EXAMPLE?ref_type=heads)
* [public.tableau.com/app/profile/sakib.mahmud1560/viz/AnimeAnalyticsthroughStudios/AnimeAnalyticsDashboard](https://public.tableau.com/app/profile/sakib.mahmud1560/viz/AnimeAnalyticsthroughStudios/AnimeAnalyticsDashboard)