Supervised machine learning on Kaggle's anime data

Team B1: Linh To, Carmen Cheung, Haiyuan Zhang, Song Lin, Zichen Wang

1. Business Problem

Anime is a style of film and television animation originated from Japan and has been growing more and more popular around the world. Every year, great series like "*NARUTO", "Death Note", "Attack on Titan"* would draw much attention from the audience and cause heated discussion. Other classic anime films like *"Spirited Away"* or *"Your Name"* are also known worldly with a box revenue of more than 200 million each.

The anime industry is no doubt a rapidly growing and profitable business. The reason why anime is getting more affection is because it encourages diverse viewpoints and gives the audience opportunities to share their thoughts across different cultures. However, large numbers of genres of anime make it hard to target an audience. It can be confusing for new anime viewers to select their favorites between the expansive collections of anime available.

In order to introduce and recommend different anime to new audiences, it is important to assess and characterize these anime. A recommendation system based on those who have seen and reviewed popular anime can be used in a variety of business settings, such as for recommending similar anime for users to enjoy by anime content providers or under standing current trends in anime to make decisions on what kind of anime merchandise to sell in stores. The purpose of our analysis is to use a variety of unsupervised machine learning approaches such as clustering and text analysis to drive insights that help achieve these goals.

2. Dataset

The dataset we performed on was from Kaggle: https://www.kaggle.com/hernan4444/anime-recommendation-database-2020

This is a folder of five datasets (*anime.csv, anime_with_synopsis.csv, animelist.csv, rating_complete.csv, watching_status.csv*), among which *anime.csv* and *animelist.csv* were mostly used for the analysis.

1) anime.csv:

This dataset contains 17562 rows and 35 columns, showing several features of each anime. The columns are as followed:

- **MAL_ID**: MyAnimelist ID of the anime.
- **Name**: full name of the anime
- **Score**: average score of the anime given from all users in MyAnimelist database.
- **Genres**: comma separated list of genres for this anime.
- **English name**: full name in English of the anime.
- **Japanese name**: full name in Japanese of the anime.
- **Type**: TV, movie, OVA, etc.
- **Episodes**: number of chapters.
- **Aired**: broadcast date.
- **Premiered**: season premiere.
- **Producers**: comma separated list of producers
- **Licensors**: comma separated list of licensors
- **Studios**: comma separated list of studios

- **Source**: Manga, Light novel, Book, etc.
- **Duration**: duration of the anime per episode
- **Rating**: age rate
- **Ranked**: position based in the score
- **Popularity**: position based in the the number of users who have added the anime to their list.
- **Members**: number of community members that are in this anime's "group".
- **Favorites**: number of users who have the anime as "favorites".
- **Watching**: number of users who are watching the anime.
- **Completed**: number of users who have complete the anime
- **On-Hold**: number of users who have the anime on Hold.
- **Dropped**: number of users who have dropped the anime
- **Plan to Watch**: number of users who plan to watch the anime.
- **Score-10/9/8/7/6/5/4/3/2/1**: number of users who scored 1-10.

2) animelist.csv

This dataset contains 109224747 rows and 5 columns, showing watching status and ratings of users towards several animes. The columns are as followed:

- **user_id**: non identifiable randomly generated user id.
- **anime_id**: MyAnemlist ID of the anime.
- **rating**: score between 1 to 10 given by the user. 0 if the user didn't assign a score.
- **watching_status**: state ID from this anime in the anime list of this user.
- **watched_episodes**: numbers of episodes watched by the user.

3) anime_with_sypnopsis

This dataset contains 109224747 rows and 5 columns, showing animes and their content. The columns are as followed:

- **MAL_ID:** The ID of the anime
- **Name:** The name of the anime
- **Score:** The average score that participants on the website Kiss.anime has given to each anime
- **Genres:** The genres of a anime
- **synopsis:** The content of the anime

3. Exploratory Data Analysis

Figure 1: Histogram of Ratings

To see the relationship between the ratings and number of episodes in the anime series, we plotted a histogram (right). With the rating of animes from 0 to 10, audiences usually give the scores to be around 6-8.

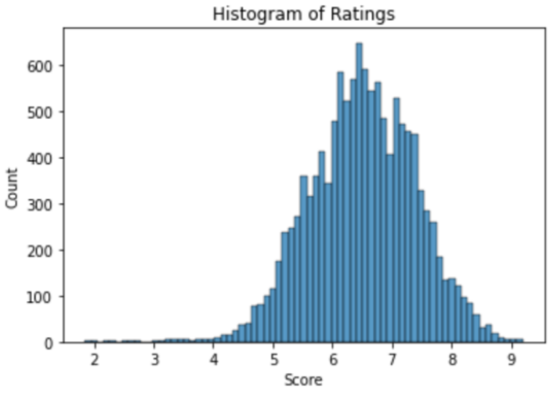
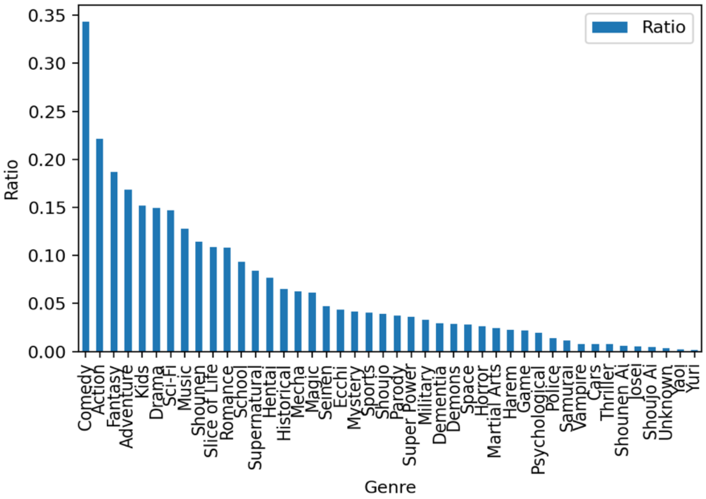


Figure 2: Ratio of Titles per Genre

In an analysis of the genre (since each title was marked under several genres), we found the proportion that the specific genre occurred amongst all the titles in the dataset. The most common genre is comedy and about ⅓ of the titles in the dataset include comedy. Some of the other most popular genres include Action, Drama, Adventure, Fantasy, and Sci-Fi.



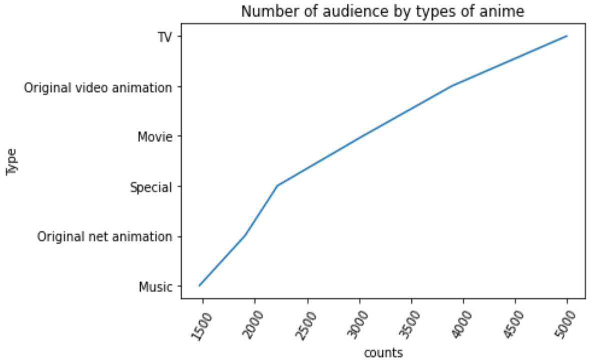


Figure 3: Number of Audience by Anime

From the line chart between number of audience and anime types, we can see that TV anime is the most popular among anime audience with 5000 people watching, followed by original video animation (4000) and movie (3250).

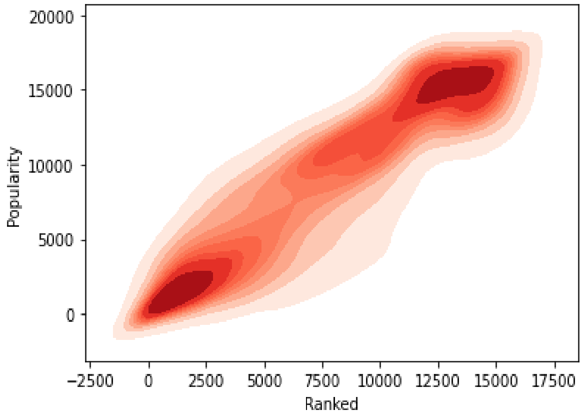


Figure 4: 2D Density Map

To avoid overplotting, which is a common issue with scatterplot, we plotted a 2D density graph to see the correlation between anime popularity and their ranks. We can see that it is almost a linear relationship, which means that the more popular the anime is, the higher they rank at myanimelist.net and vice versa. It also can be seen that the correlation will be most visible (or that there are more data points) around the high ranking and low ranking animes, rather than most of the medium ranking ones.

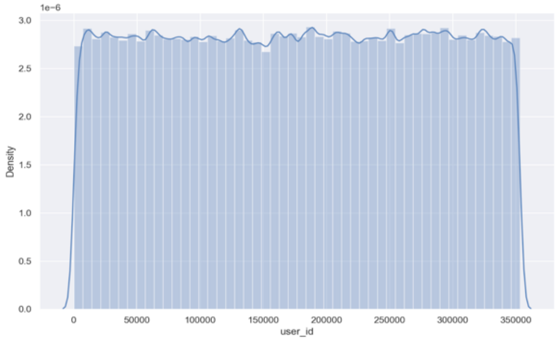


Figure 5: Density Plots of User ID

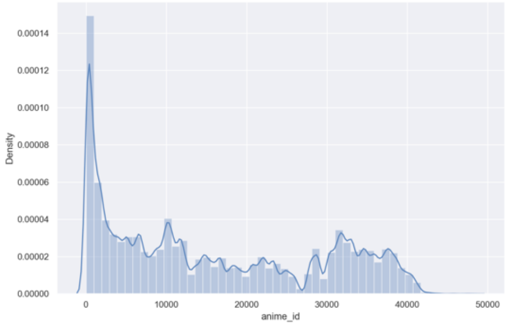


Figure 6: Density Plot of Anime ID

Furthermore, we explored the distribution of users when they rated different animes. From the plot above we can conclude that most users filled out almost the same amount of ratings for animes during the survey. But the distribution of anime seems biased because some anime are rated at high frequency while others are not.

4. Analysis and Methodology

The focus of our analysis is on clustering. We employed a variety of tools in developing unsupervised learning models such as PCA, scaling using StandardScalar, and k-means clustering with analysis of silhouette score and inertia using elbow method.

Clustering on Anime Features

Before training our models, we performed data processing. First of all, we seperate numerical and categorical data. Then we standardized our data to make sure they are on the same scale. Post-cleaning, our dataset contains 22 numeric variables we create. We try to dummify our categorical data to give more predictive power, but the number of combined columns is too large so we decide to only use numeric values. Since we have 22 columns for each observation, we decide to use PCA to reduce the dimensions and get a more accurate result.

Figure 7: Cumulative Explained Variance of PCA

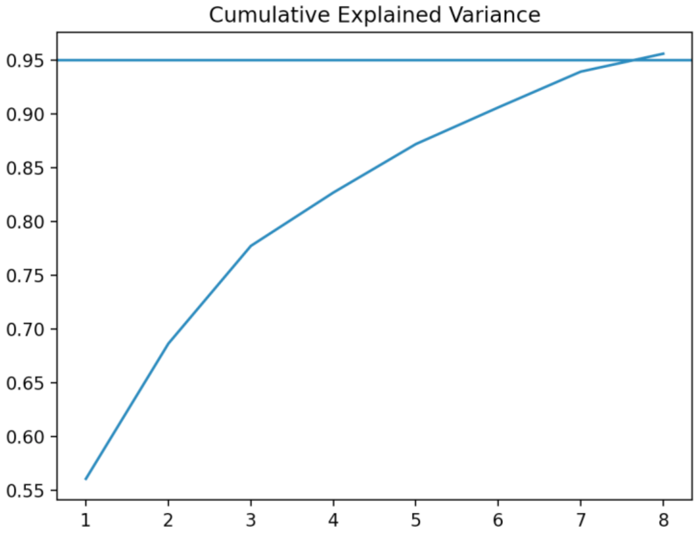


Figure 8: Inertia of Fit and k-value

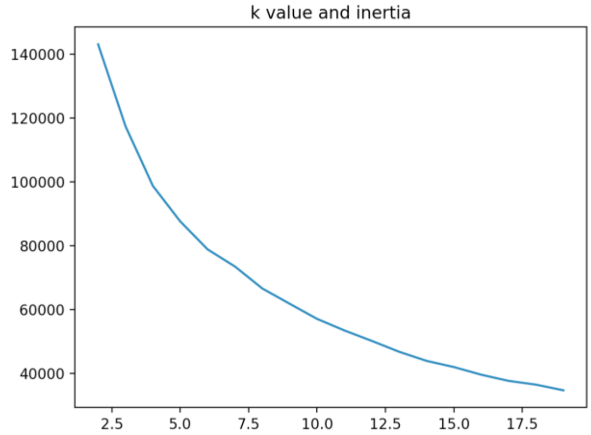
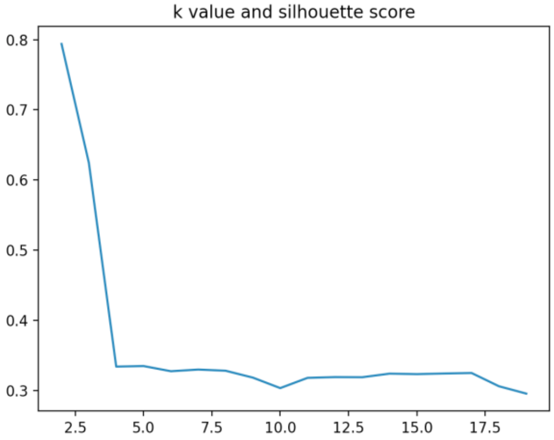


Figure 9: Silhouette Score and k-value



We decide to keep 8pcs for later clustering because it explains about 90% of the variance. Then we tried to find the proper K value based on inertia and silhouette score. Based on the graph above, k=4 is a good choice because it has low inertia and high silhouette score. After determining the K value, we fit into the K means model and get a 0.334 silhouette score. We can see from the graph that some mis-clustering exists in group 1 and 2, but the model performed well overall.

Figure 10: Silhouette Analysis with k = 3

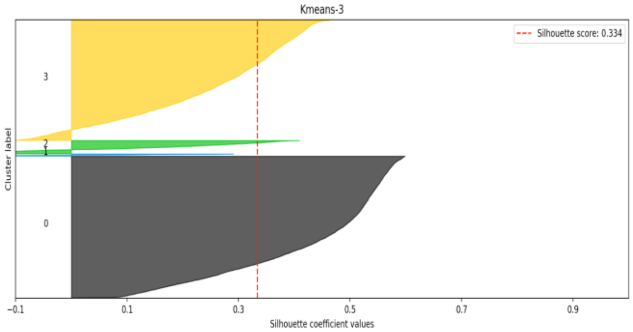
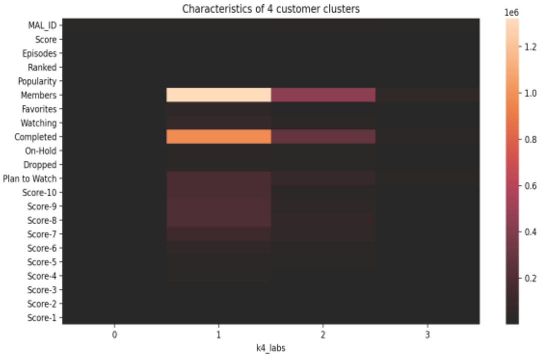


Figure 11: Heatmap of Clusters



Clustering0: There are 5481 entries in this cluster. They are relatively low at most characteristics.
Clustering1: There are 78 entries in this cluster. They have high members and high complete episodes. In this cluster, the ratings for animes above 5 is more common.
Clustering2: There are 508 entries in this cluster. They have relatively high members and high complete episodes, but the overall rating is lower than cluster1.
Clustering3: There are 4634 entries in this cluster. Similar to cluster0, they are relatively low at most characteristics, but with more members and complete episodes.

Clustering on User Ratings

Another analysis we conducted was to perform clustering analysis on users based on the ratings they gave for each anime. From the animelist we derived a dataframe with “user_id” as the index, “anime_id” as the column, and the rating as the value in each cell. Due to the size of the data (a data frame with 109,224,747 rows and 5 columns, where a majority of users did not watch overlapping anime titles, we preprocess the data to analyze on a subset of the entire dataframe.

From the figure below, a majority of the anime titles are viewed by less than 5% of viewers and even more titles (12,939 out of 17,562 or 73.7%) viewed by less than 1% of users . Therefore we focused on the anime that are the most popular by user viewing rate on this platform (titles that were viewed by at least 40% of the users). Figure 13 is a closer look at the histogram of anime count that have been rated by at least 40% of the total users including the top few anime titles above 60% (Death Note, Attack on Titan, Sword Art Online, Fullmetal Alchemist: Brotherhood, and Toradora!). The reason we conducted a form of variable reduction like this was because we are mostly interested in the anime that have a lot of ratings and were popular, recognizable titles to conduct further analysis upon to build a recommendation system. From there, we got 59 anime titles of which 2,507 users have seen at least one episode of all of these most popular anime and we will use their ratings to perform cluster analysis on these users.

Figure 12: Histogram of Percentage Watched

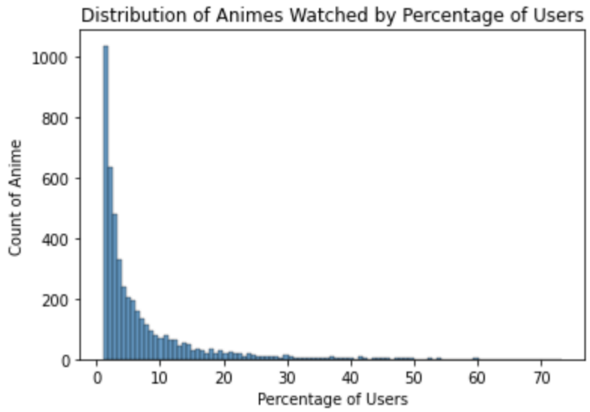
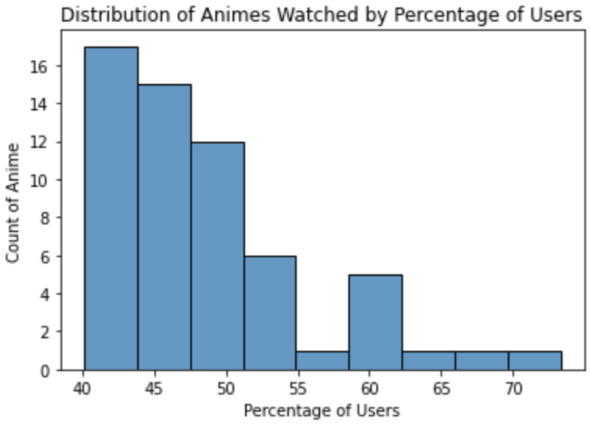
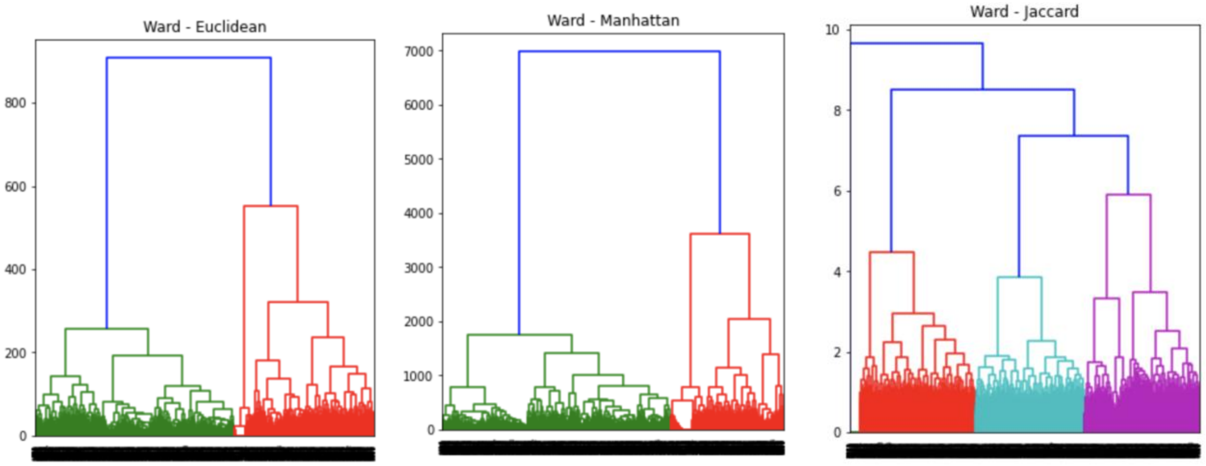


Figure 13: Histogram of Anime over 40% Watched



We observed the grouping of users using hierarchical and k-means clustering. Using hierarchical clustering, we constructed several dendrograms to visualize the clusters of users and their distance because of the large amount of users included in this analysis. We took a look at a variety of distance measurement techniques such as Euclidean, Manhattan, Jaccard, and Cosine as well as different linkage methods including complete, single, average, and ward. We compared the results between all the dendrograms generated like in Figure 14 below and determined that measuring distance using Jaccard and Ward linkage methods produced the closest clusters amongst these methods.

Figure 14: Dendrograms using Ward Linkage Method and Various Distance Calculations



We also performed k-means clustering using this dataset between users and their ratings of each anime. For this method, we looked at the inertia of fit and the silhouette scores of the clusters. Using the elbow method on inertia (Figure 15) and other considerations, we found that at about k = 5 would be optimal for the number of clusters. The silhouette score was 0.113 and some misclustering especially in cluster 0,1, and 3 (Figure 16). This could be due to the non-compliance of users to rate every single anime they have watched which resulted in a zero rating. Clustering can be improved with more ratings from these users.

Figure 15: Inertia of Fit of k-value

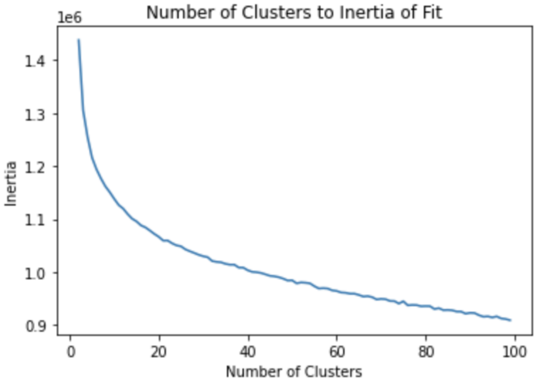
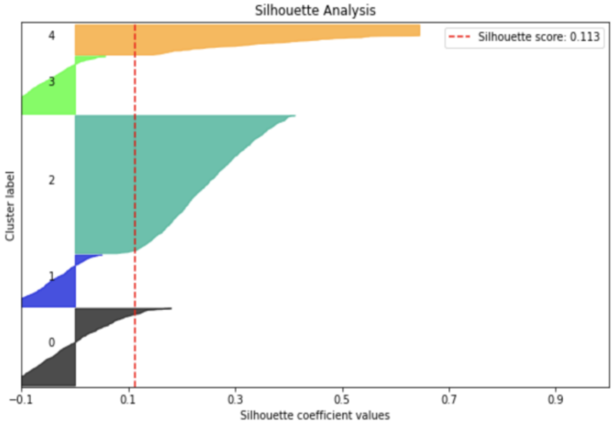


Figure 16: Silhouette Analysis Plot with k = 5



Despite not having very well clustered clusters , this is the basis of our recommendation system that we will go into in the next section. The goal is to further improve on this clustering and deciding similarity to bypass the issues with users that do not give a rating for some of the anime they have watched.

5. Recommendation System

Recommendation system using collaborative filtering (based on KNN)

Based on the above analysis, we will find that directly using some elements of the animation itself for cluster analysis is not good. This is because the variables we can measure do not well reflect the quality or style characteristics of a certain animation. For example, the number of episodes, the release time, the length of time, the type of film and television, etc., are not the content that we will pay attention to when we appreciate anime.

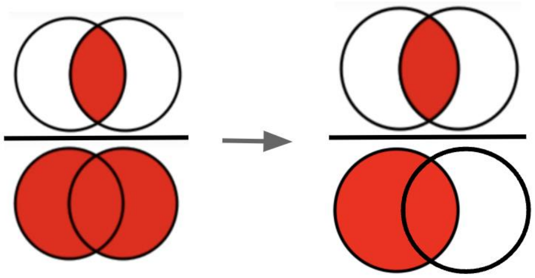
Some factors that are difficult to measure, such as painting style, story, voice actors, special effects, etc., are the most critical factors in evaluating an animation.

Therefore, we take a different approach and use collaborative filtering to form a recommendation system.

We drew inspiration from jaccard distance and knn algorithm, and made some modifications to make them more suitable for solving current problems.

- Instead of dividing the size of intersection by the size of union like Jaccard distance, we just divide the intersection by the size of the new user’s favorite list to measure the distance between the new user and the existing user (Figure 17). That is, we only hope that the favorite list of existing users covers the list of new users as comprehensively as possible

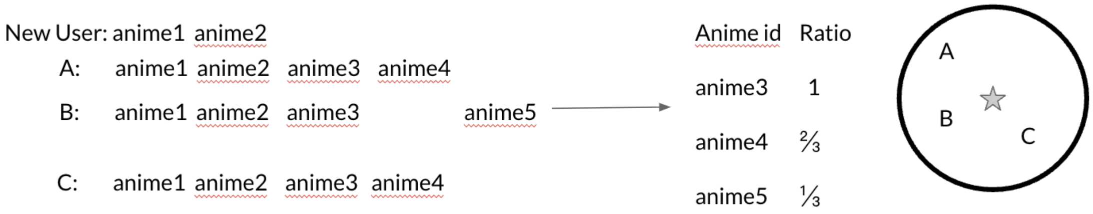
Figure 17: New distance calculate method



- Knn determines the type of new observation by voting. But there is no simple label to vote for when recommending anime. Therefore, existing nearby users will vote for every anime in their favorite list. By counting the number of votes for each anime, we can get the ranking of some anime and take it as the recommendation list.

  A simple example is shown below (Figure 18). Anime1 and anime2 is the favorite anime entered by the new user. And the ratio reflect the proportion of users 'voted' for that anime.

  Figure 18: Example of 'Voting'



New User: anime1 anime2

A: anime1 anime2 anime3 anime4

B: anime1 anime2 anime3 anime5

C: anime1 anime2 anime3 anime4

Anime id	Ratio
anime3	1
anime4	⅔
anime5	⅓

- The application of Knn needs to determine the number of k. So we came up with a new evaluation method. For each user in the test set, we only input 10 animations from his favorite list in the recommendation system. At the same time, only the top 10 animations in the recommendation system are intercepted as the recommendation list. We can determine the value of the hyperparameter k by measuring the number of animations in the recommended list that appear in the user's original complete favorite list in each test set.

The validation result is shown in Figure 19. We choose to set k as 60.

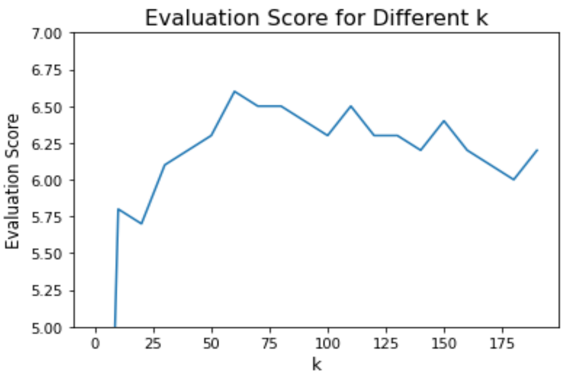


Figure 19: Validation Score for different k-value

Based on the evaluation results, if the new user enters 10 anime he likes, then at least 6 of the top 10 recommending anime will meet his tastes. This is still an underestimation, since the users in the test set may not have watched all the recommended anime. So although the recommended anime is not in his favorite list, we don't know if he doesn't like it or hasn't watched it yet.

Recommendation system using NLP on content of the anime

Dataset: We used the dataset called anime_with_synopsis.csv with the shape of 16214 rows and 5 columns of animes and their content and we performed the recommendation system with the process below:

1. Dataset cleaning
   - Subset the dataset: with more than 16k rows of animes, we thought that would be too large and unnecessary to include all of them, so we took 6k most highly rated animes by sorting the column "Score" descending.
   - Cleaning column "sypnopsis": For text preprocessing techniques we performed converting the column into lowercase, removing punctuation, removing strange characters that are not English (with regex), removing stopword (with nltk library), lemmatizing words (with spacy library).
   - We then tokenize the column "sypnopsis" into a document term matrix (with sklearn's CountVectorizer) with the total number of tokens is 28720 tokens.

2. Cosine similarity calculation
   Cosine similarity is frequently used to calculate the similarity between the documents (with sklearn library) and from then we can use these similarities to recommend the anime.

3. Recommendation
   From the cosine calculation we had a list of tuples and we did enumeration through each of them. We then created a new column to enumerate through the anime dataset and created a function to match the top 5 highest similarity scores to an anime to their respective names in the dataset, which results in a recommendation system based on the anime's content. Now the user can enter the anime's name and the recommendation system will give 5 similar animes for the user.