# ANIME RECOMMENDATION SYSTEM

## About Dataset

**This data set contains information on user preference data from 73,516 users on 12,294 anime. Each user is able to add anime to their completed list and give it a rating and this data set is a compilation of those ratings.**

## Content

**Anime.csv**

- **anime_id - myanimelist.net's unique id identifying an anime.**
- **name - full name of anime.**
- **genre - comma separated list of genres for this anime.**
- **type - movie, TV, OVA, etc.**
- **episodes - how many episodes in this show. (1 if movie).**
- **rating - average rating out of 10 for this anime.**
- **members - number of community members that are in this anime's "group".**

**Rating.csv**

- **user_id - non identifiable randomly generated user id.**
- **anime_id - the anime that this user has rated.**
- **rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).**

## Dataset taken from myanimelist.net API

# CONTENT BASED RECOMMENDER

Content based filtering recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. A content based recommender works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). Based on that data, a user profile is generated which is then used to make suggestions for the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

**Term Frequency(TF) & Inverse Document Frequency(IDF)**

TF is simply the frequency of a word in a document. IDF is the inverse of the document frequency among the whole corpus of documents. TF-IDF is used mainly because of two reasons: Suppose we search for “the rise of analytics” on Google. It is certain that “the” will occur more frequently than “analytics” but the relative importance of analytics is higher than the search query point of view. In such cases, TF-IDF weighting negates the effect of high frequency words in determining the importance of an item (document). Here we are going to use it on the genre of animes so that we can recommend contents to the users based on genres.

Also, scikit-learn already provides pairwise metrics that work for both dense and sparse representations of vector collections. Here we need to assign 1 for recommended anime and 0 for not recommended anime. We will use sigmoid kernel here