Anime Recommendation

...

Eswar Chand
Bhanu Prashanth
Vineel Kurma

Objective & Data

Objective: Recommending Anime to MyAnimeList.com Users

Data:

1. Kaggle dataset
2. 2 files: Anime & Rating
    a. Anime: AnimeID, Name, Genre, Type, Episodes, Rating, Members
    b. Rating: UserID, AnimeID, Rating
3. Unique Anime : 12,294 ; Unique Users: 73,315
4. Type: TV, Movie, OVA, Special, Music, ONA
5. Ratings: Anime data - average rating given by viewers to the anime; User data - rating given by the viewer to the anime

Data:

Anime Table

anime_id	name	genre	type	episodes	rating	members
32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
5114	Fullmetal Alchemist: …	Action, Adventure, Drama, Fantasy, Magic, Military, Shounen	TV	64	9.26	793665
28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	TV	51	9.25	114262
9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572

Rating / User Table

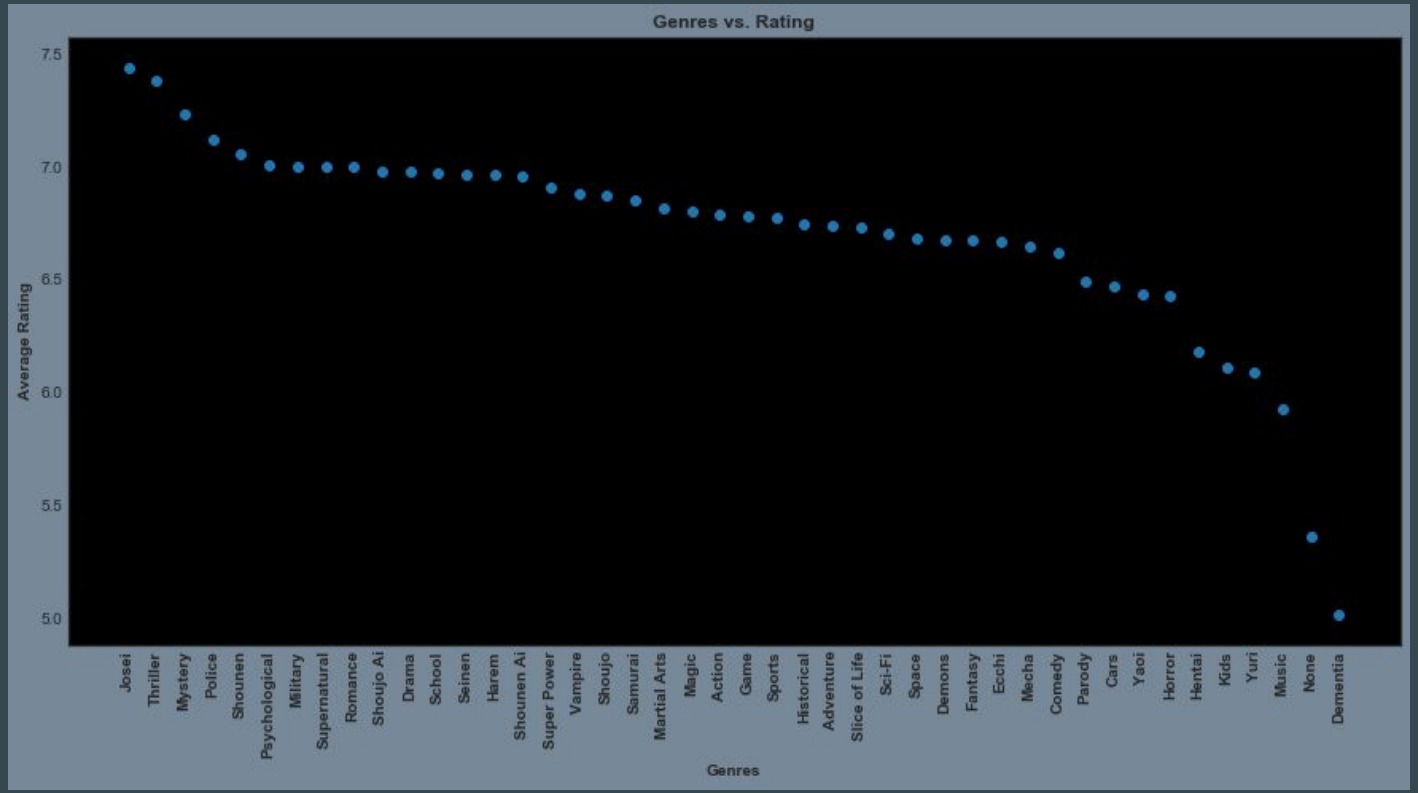
user_id	anime_id	rating
2	11771	10
2	12189	-1
2	16417	-1
3	20	8
3	154	6

Exploratory Data Analysis

Data Manipulation:

- Replaced all the records with rating as -1 to nan, as this indicates that the User hasn’t given any rating to the Anime
- Used only TV data for our analysis
- Merged both Anime and Rating data using the anime_id column

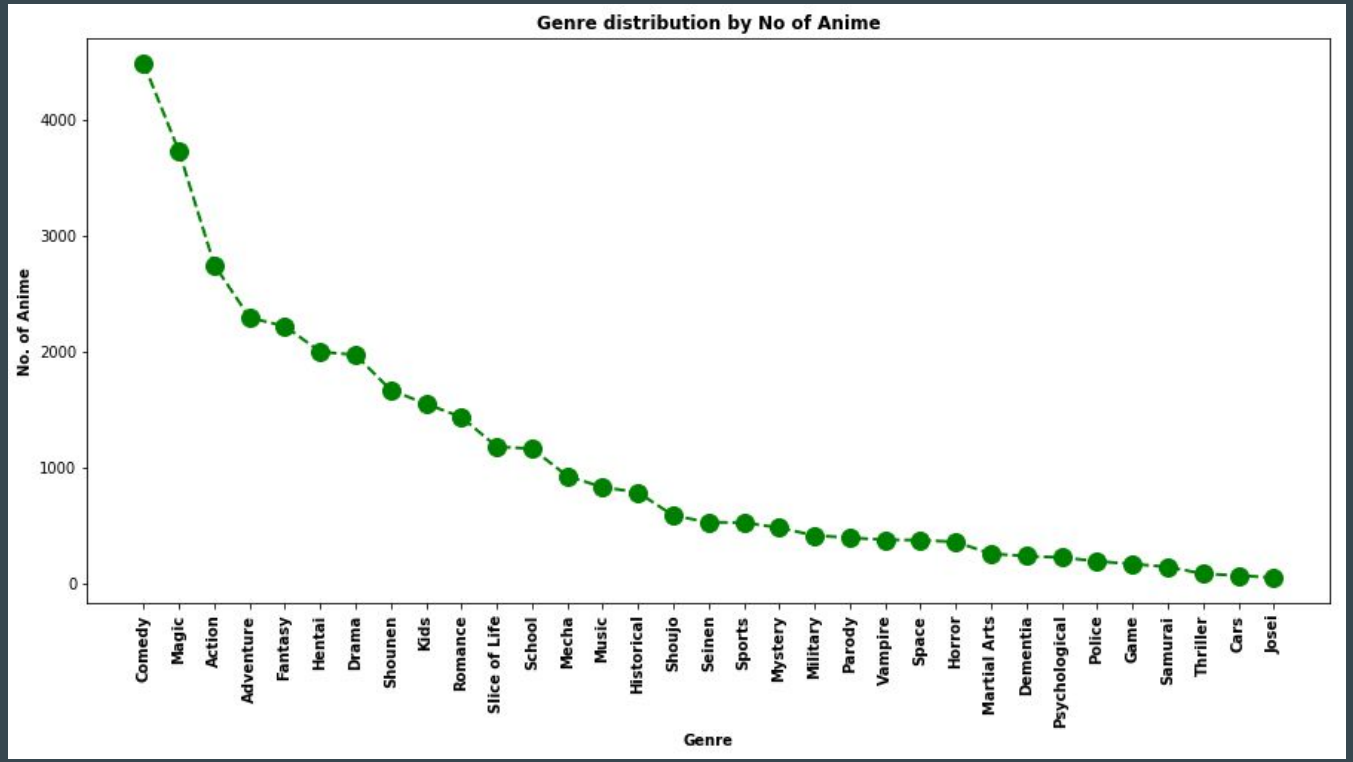
Data Exploration: Genres vs. Rating



Top 5 Genres rating wise

- Josei
- Thriller
- Mystery
- Police
- Shounen

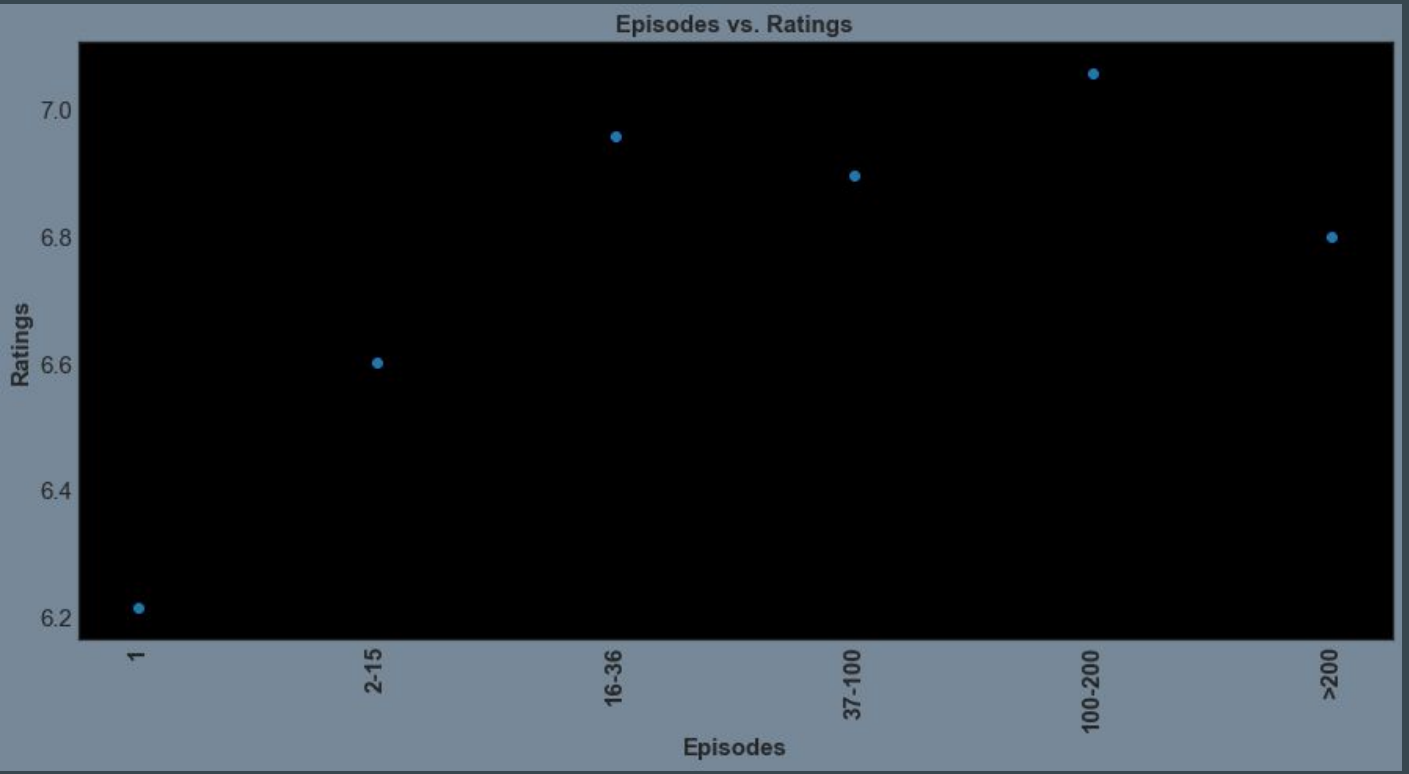
Data Exploration: Genres vs. # of Anime



Top 5 Genres by # Anime

- Comedy
- Magic
- Action
- Adventure
- Fantasy

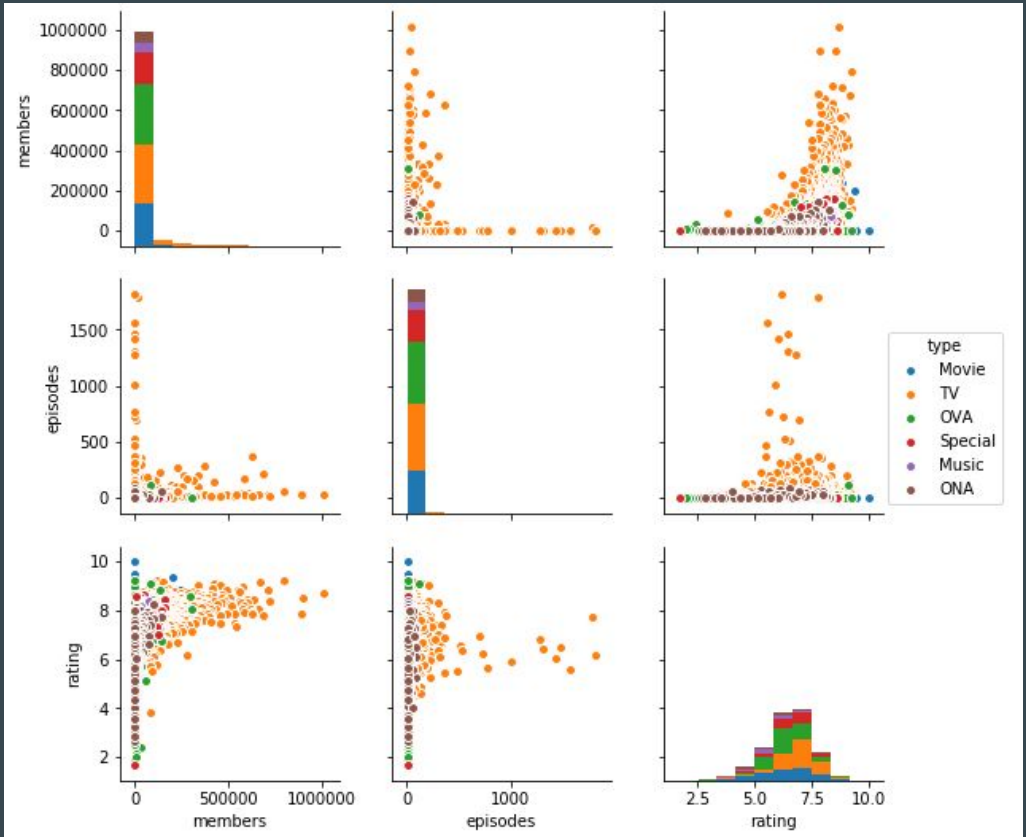
Data Exploration: Episodes vs Ratings



Top 3 Episodes segment

- 100 - 200
- 16 - 36
- 37 - 100

Data Exploration: Members, Episodes & Rating for Anime Types



- We can see that the members are skewed right towards higher rating and skewed left towards lesser episodes
- Episodes vs. Rating graph follows a normal distribution pattern

Modelling

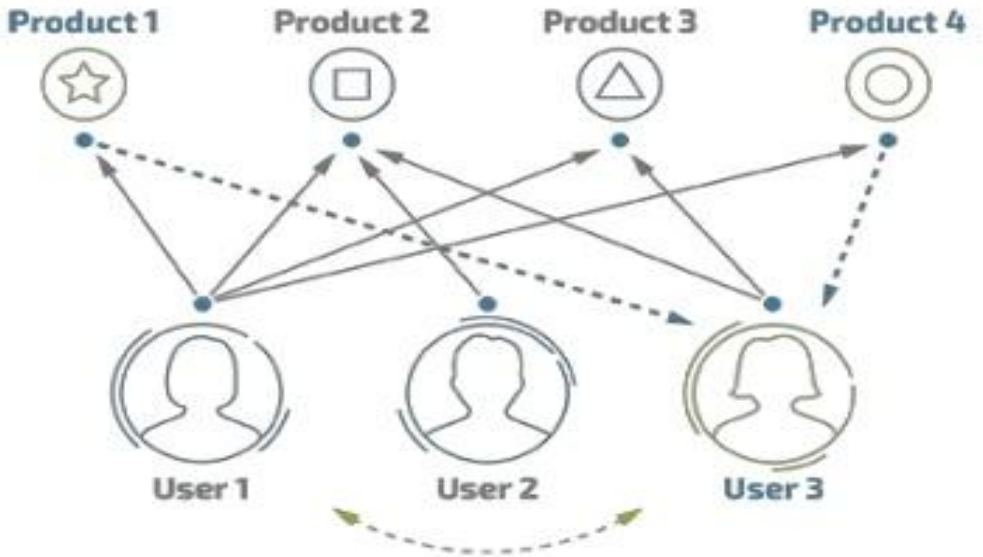
Modelling Approach:

- Two Approaches
  - Collaborative Filtering
    - Pearson Similarity
    - Alternating least squares
  - Content Filtering
    - Custom algorithm

Collaborative Filtering - Feature selection and scaling:

- Formed a crosstab of the merged data, with Anime as the rows and Users as the columns and user ratings as cell values. This crosstab is required to provide the recommendations based on user collaborative filtering.
- Normalised the ratings for each user to adjust for the differential ratings between the users
- Dropped all columns with only 0's, representing the users who didn't rate

Collaborative Filtering - CF

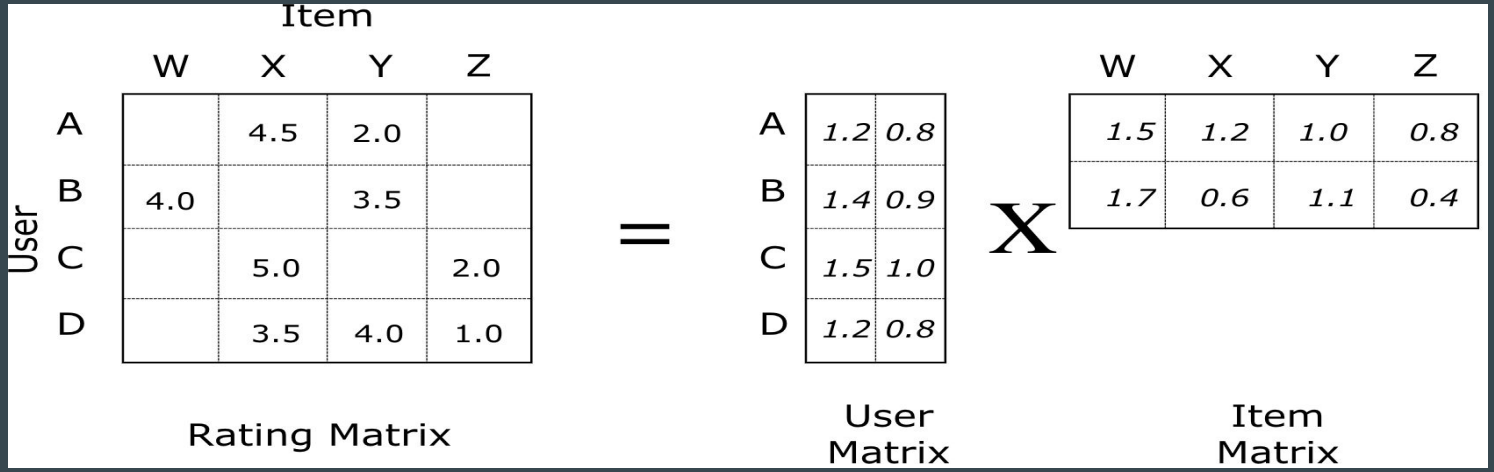


Collaborative Filtering : Pearson Similarity(correlation)

1. We have used the pearson similarity to find the similar users to the current user
2. Created the set of anime that the similar users have watched and removed the anime that the target user has watched
3. Now from the remaining set we have selected the top animes with the highest frequency (no of times similar users have watched the anime)
4. These anime are the recommended anime based on user similarity

$$sim(u,v) = \frac{\sum (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sigma_u \sigma_v}, \quad sim(u,v) > 0$$

Matrix Factorization Algorithms



Matrix Factorization using SGD

SGD Algorithm for MF

Input: training matrix V, the number of features K, regularization parameter λ, learning rate ϵ
Output: row related model matrix W and column related model matrix H

1: Initialize W, H to $UniformReal(0, \frac{1}{\sqrt{K}})$
2: **repeat**
3: **for random** $V_{ij} \in V$ **do**
4: $error = W_{i*}H_{*j} - V_{ij}$
5: $W_{i*} = W_{i*} - \epsilon(error \cdot H_{*j}^{\intercal} + \lambda W_{i*})$
6: $H_{*j} = H_{*j} - \epsilon(error \cdot W_{i*}^{\intercal} + \lambda H_{*j})$
7: **end for**
8: **until** convergence

$$f(U, M) = \sum_{(i,j) \in I} (r_{ij} - \mathbf{u}_i^T \mathbf{m}_j)^2 + \lambda \left(\sum_i n_{u_i} \|\mathbf{u}_i\|^2 + \sum_j n_{m_j} \|\mathbf{m}_j\|^2 \right)$$

Alternating Least Squares

```
function M = updateM(lAcols, U)
    lamI = lambda * eye(Nf);
    lM = zeros(Nf,Nlm); lM = single(lM);
    for m = 1:Nlm
        users = find(lAcols(:,m));
        Um = U(:, users);
        vector = Um * full(lAcols(users, m));
        matrix = Um * Um' + locWtM(m) * lamI;
        X = matrix \ vector;
        lM(:, m) = X;
    end
    M = gather(darray(lM));
end
```

Content filtering : Using ratings and genre tags

This algorithm leverages the genre similarity, user ratings and anime overall ratings

Algorithm:
Input: (userId, animeDf, usersDf, genreComboDict)
Output: recommendations List sorted by anime ratings

The description of the algorithm:
1) Get the user rated animes
2) Compute the user average rating
3) User liked animes: Filter the animes with user rating greater than the average
4) For each user liked anime:
    Get the genre of the anime
    Get the top most similar genre combinations
    Take the animes in the top most similar combinations other than user watched animes
5) Return the top animes sorted by the anime ratings

Conclusions

Merits & Demerits :

- Collaborative filtering :
  - Captures other user similarity. Doesn't leverage genre similarity for recommendations
  - Ex: The recommendations are a bit more skewed towards other users preferences compared to that particular user

- Content filtering :
  - Takes in each users preferences. Doesn't take in other users' preferences
  - Ex: It gets difficult for a user to shift to a new type of anime/genre because the recommendations are based on just the history of that particular user

Output :

User Watched:

Gakkou no Kaidan, Durarara!!, Highschool of the Dead, Ao no Exorcist, Sword Art Online

Collaborative Filtering (Pearson Similarity) :

Chobits
Code Geass: Hangyaku no Lelouch
Death Note
Dragon Ball
High School DxD
Mirai Nikki (TV)
Monster
Neon Genesis Evangelion
Attack on Titan

Collaborative Filtering (ALS) :

Bleach
Death Note
Fairy Tail
Fullmetal Alchemist
Fullmetal Alchemist: Brotherhood
Naruto
Attack on Titan
Soul Eater

Content Filtering (Custom) :

Ao no Exorcist: Kyoto Fujouou-hen
Durarara!!x2 Ketsu
Durarara!!x2 Shou
Durarara!!x2 Ten
Jigoku Shoujo Futakomori
Jigoku Shoujo
Sword Art Online II
Umineko no Naku Koro ni

Conclusions :

- The three methods used gave a set of anime as results, with minimal intersections among the three results, as each method captures different levels of characteristics
- We can extend the algorithm to cluster the users by finding the favourite genres for each user. This can be used to capture the target customers at a higher level
- Combination of the methods will give a more refined set of recommendations

References

Matrix factorization and ALS introduction:

https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1

ALS Paper:

https://endymecy.gitbooks.io/spark-ml-source-analysis/content/%E6%8E%A8%E8%8D%90/papers/Large-scale%20Parallel%20Collaborative%20Filtering%20the%20Netflix%20Prize.pdf

Critical Questions

1) How do you suggest to make the hybrid algorithm to combine content based and collaborative based filtering?
2) How can we extend the content based custom algorithm to make user clusters?
3) What additional data would we need to make the recommendations more accurate?

Arigato / Thank you