AI 705 Recommendation Systems: Midterm Project

Movie Recommendation System

IMT2021009 Madhav Sood

IMT2021065 Vihan Vashishth

IMT2021066 Yukta Rajapur

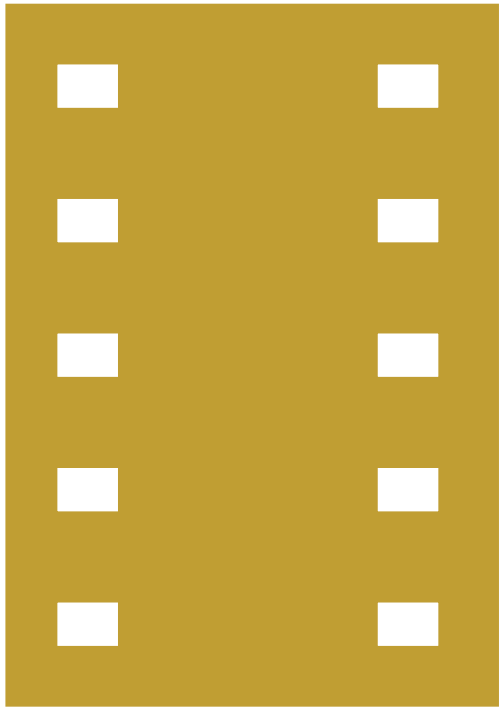
IMT2021067 Brij Desai

Dataset and Problem Statement

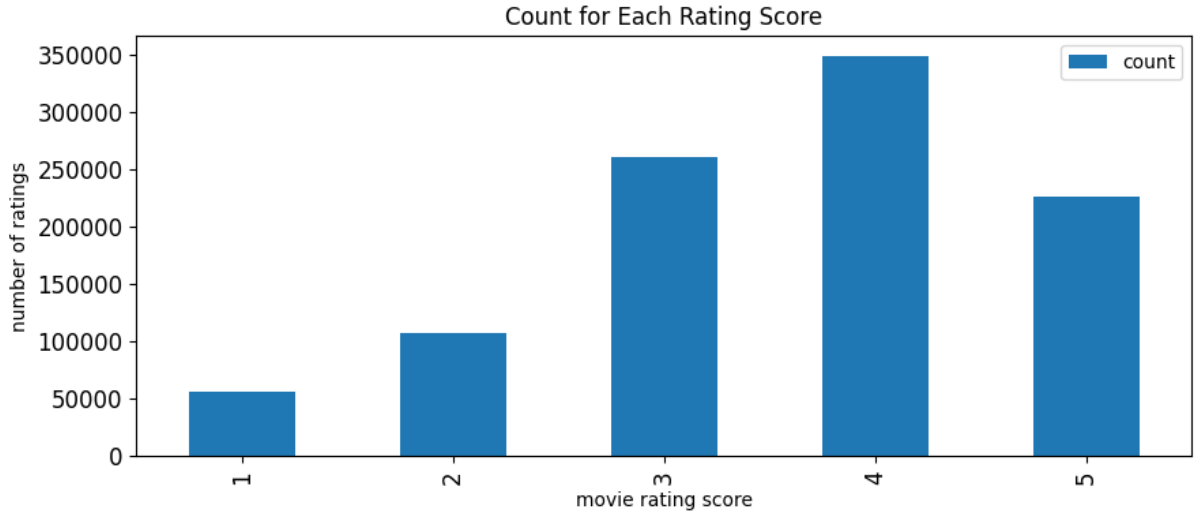
- Dataset used is the MovieLens 1M Dataset with ratings of 6040 users across nearly 4000 movies.

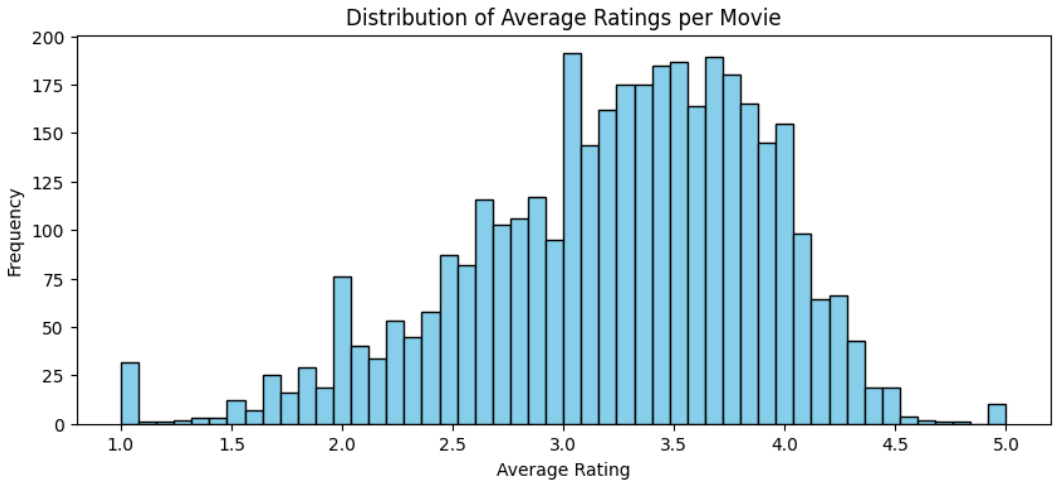
Problem Statement:

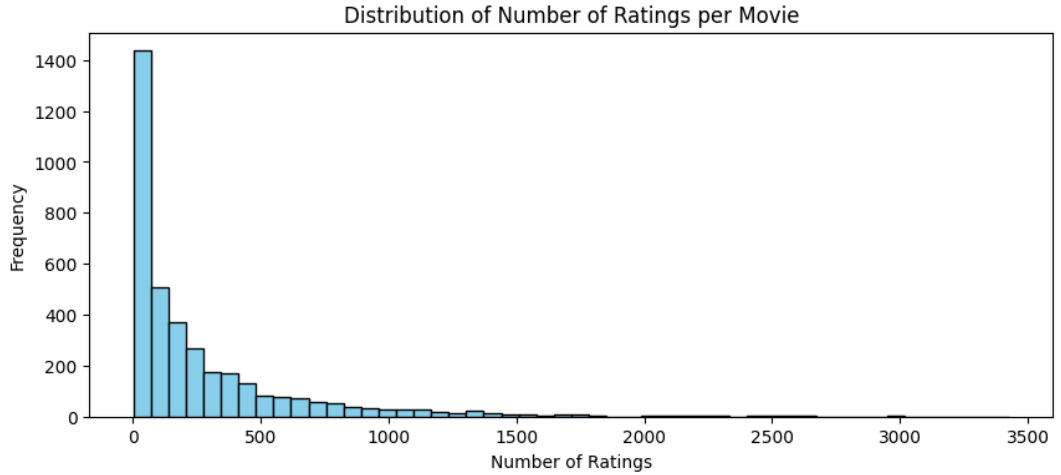
- Build a movie recommendation system, where based on the existing ratings of a user, you should suggest new movies that the user will like.
- Implement necessary methods while adding our own novel ideas.

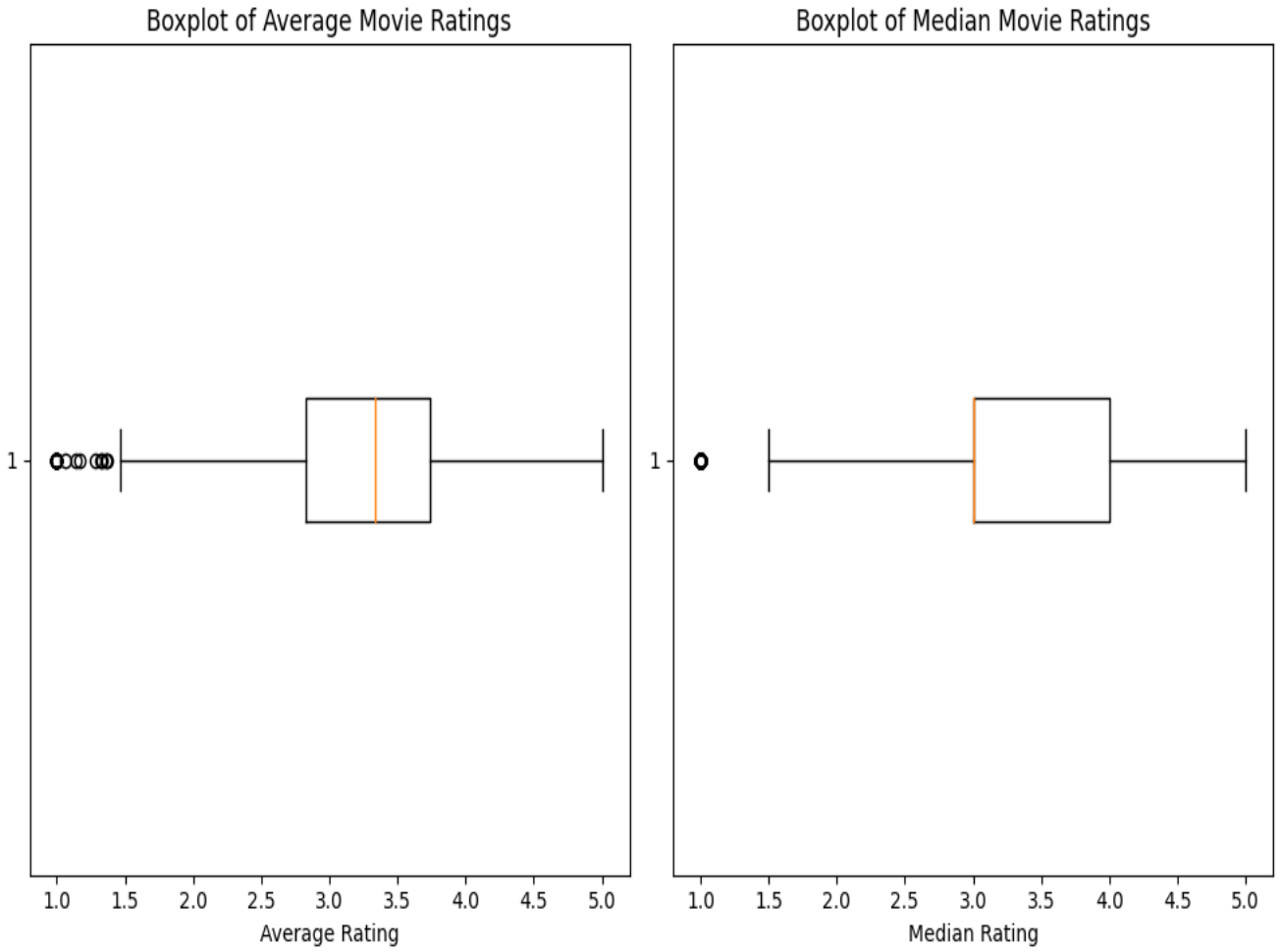


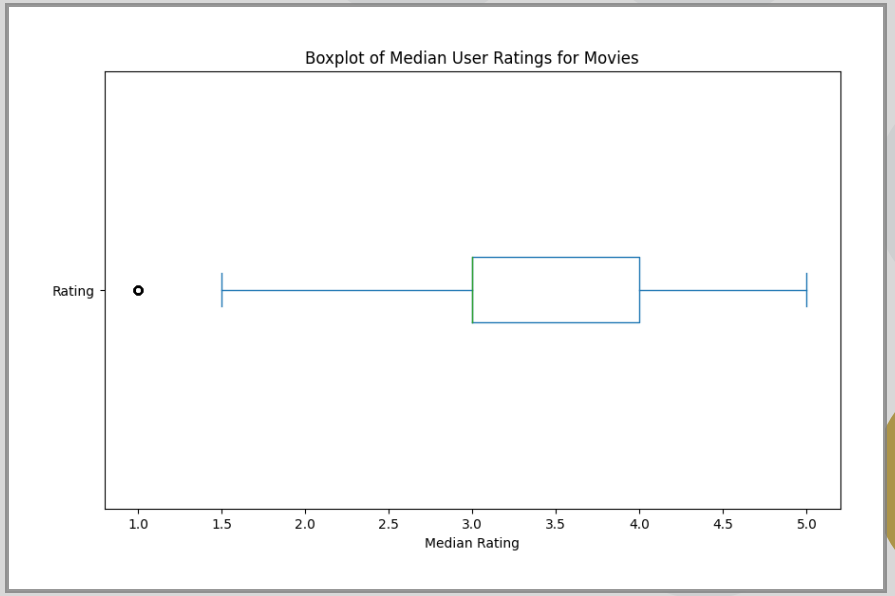
EDA

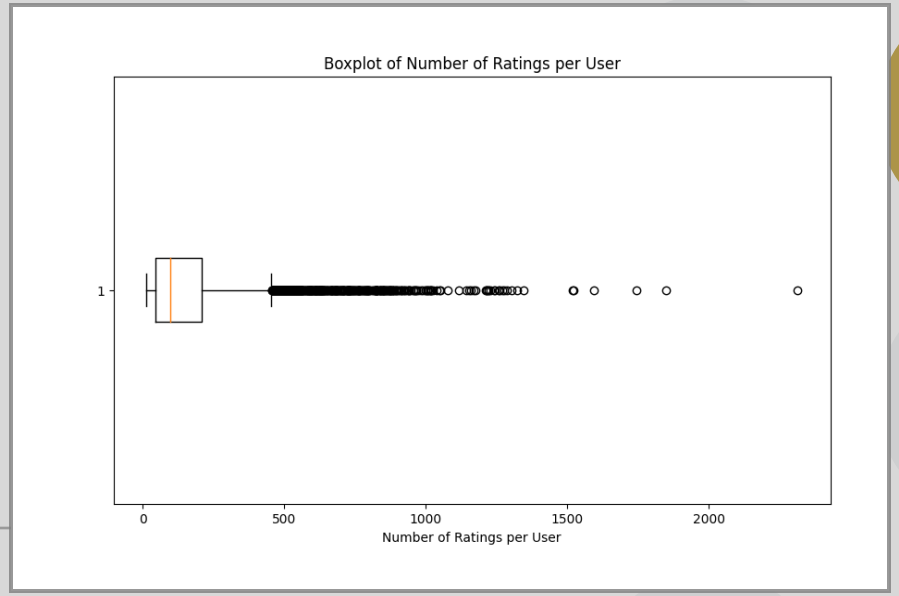

Count for Each Rating Score
count
number of ratings
movie rating score
0
50000
100000
150000
200000
250000
300000
350000
1
2
3
4
5

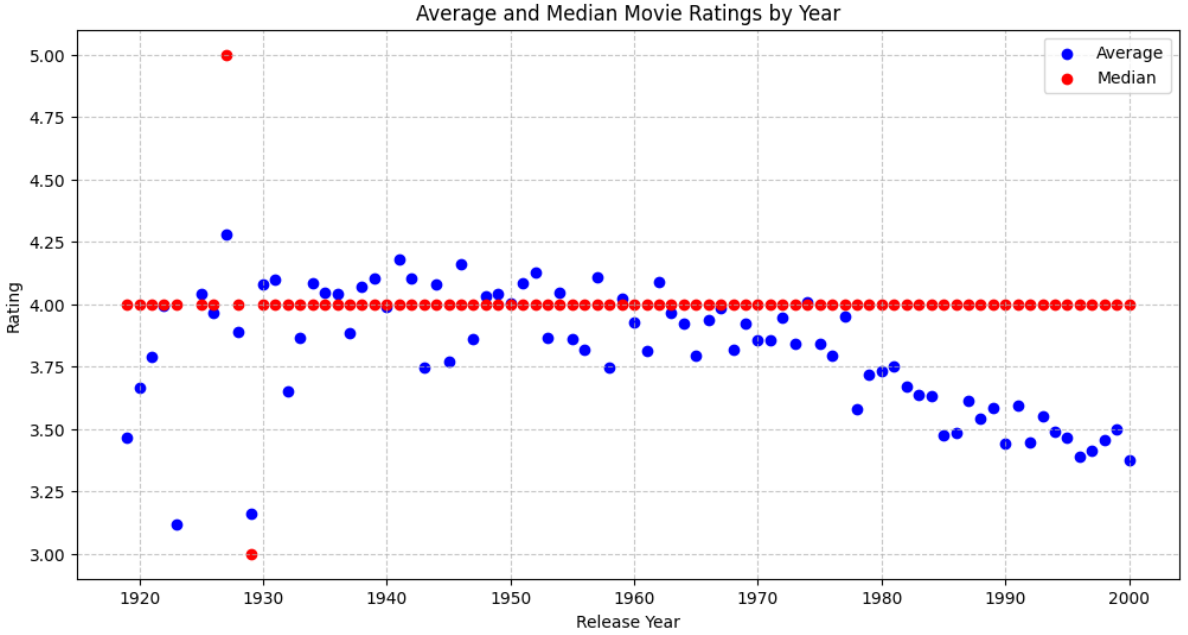

Distribution of Average Ratings per Movie
Frequency
Average Rating
0
25
50
75
100
125
150
175
200
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

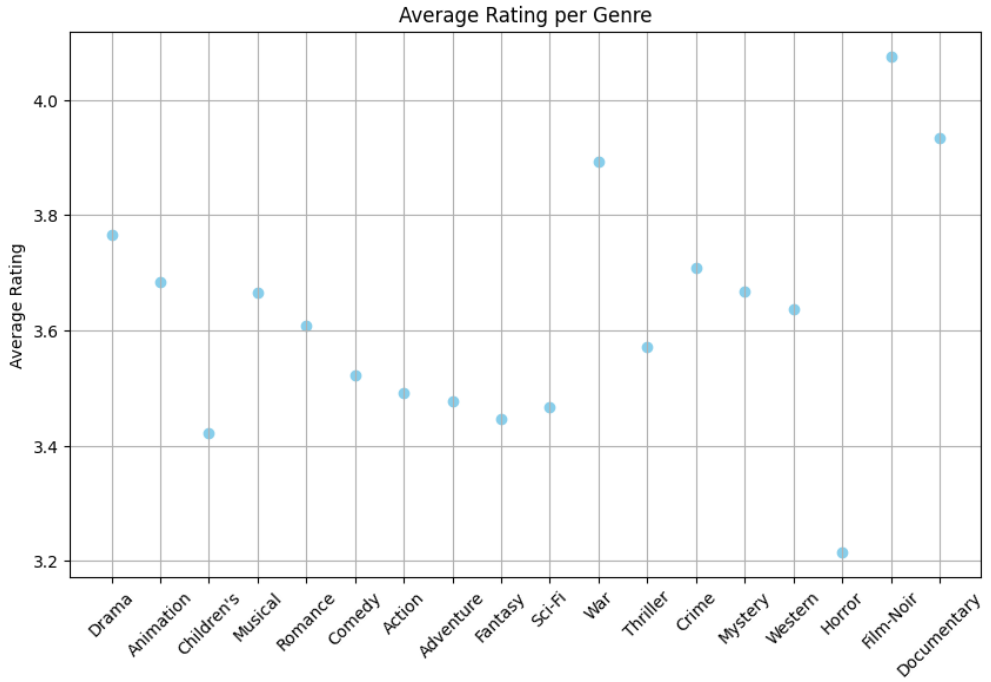

Distribution of Number of Ratings per Movie
Frequency
Number of Ratings
0
200
400
600
800
1000
1200
1400
0
500
1000
1500
2000
2500
3000
3500

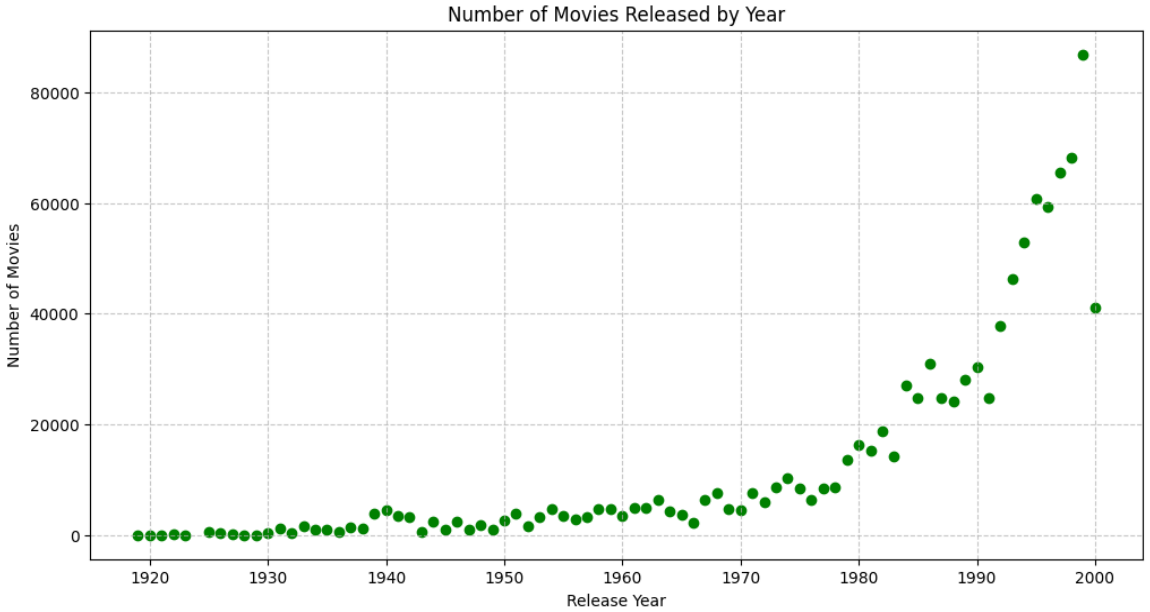

Boxplot of Average Movie Ratings
Boxplot of Median Movie Ratings
1
1
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Average Rating
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Median Rating

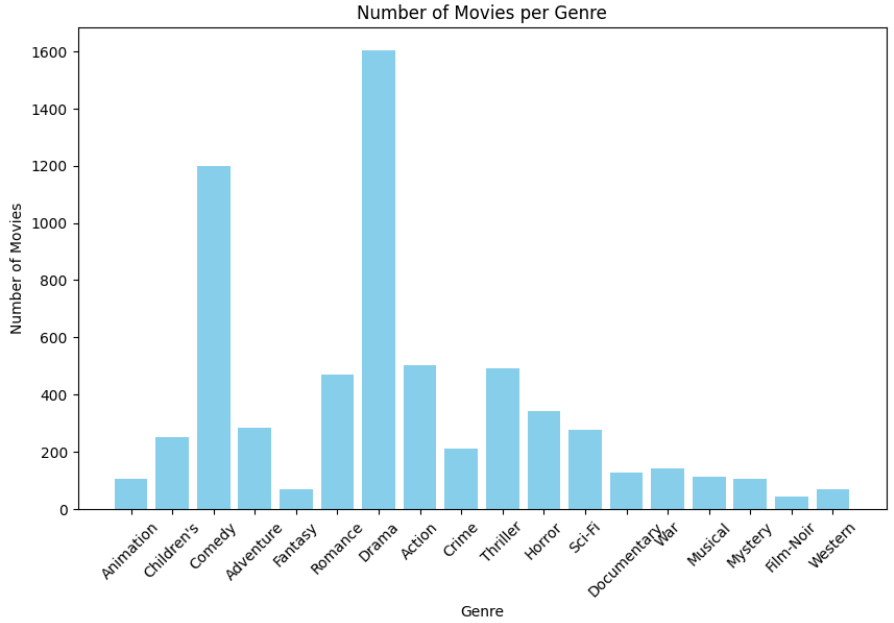

Boxplot of Median User Ratings for Movies
Rating
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
Median Rating

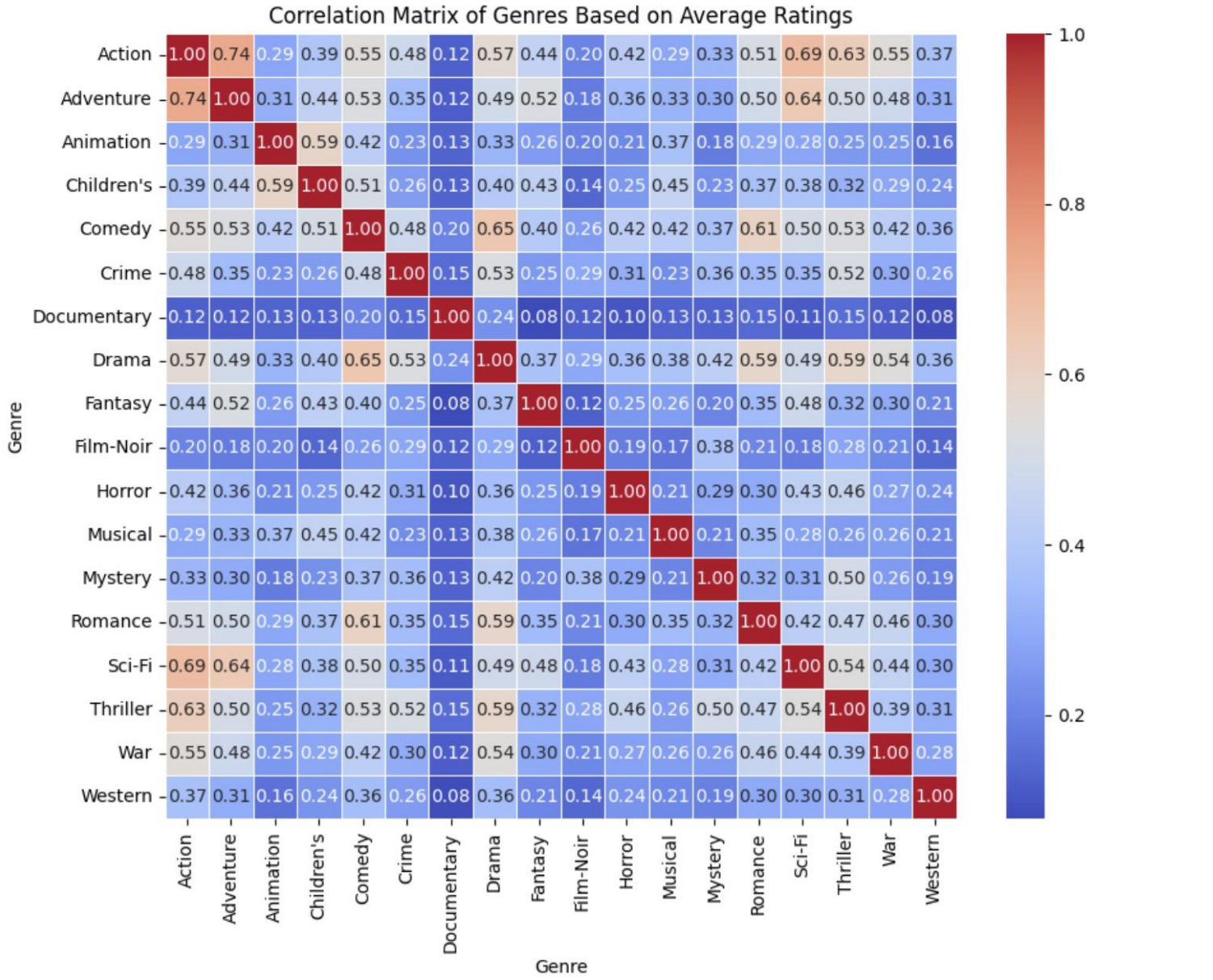

Boxplot of Number of Ratings per User
1
0 500 1000 1500 2000
Number of Ratings per User


Average and Median Movie Ratings by Year
Average
Median
Rating
Release Year
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
1920
1930
1940
1950
1960
1970
1980
1990
2000


Average Rating per Genre
Average Rating
4.0
3.8
3.6
3.4
3.2
Drama
Animation
Children's
Musical
Romance
Comedy
Action
Adventure
Fantasy
Sci-Fi
War
Thriller
Crime
Mystery
Western
Horror
Film-Noir
Documentary


Number of Movies Released by Year
Number of Movies
Release Year
80000
60000
40000
20000
0
1920
1930
1940
1950
1960
1970
1980
1990
2000


Number of Movies per Genre
Number of Movies
Genre
1600
1400
1200
1000
800
600
400
200
0
Animation
Children's
Comedy
Adventure
Fantasy
Romance
Drama
Action
Crime
Thriller
Horror
Sci-Fi
Documentary
War
Musical
Mystery
Film-Noir
Western


Correlation Matrix of Genres Based on Average Ratings
Genre
Action 1.00 0.74 0.29 0.39 0.55 0.48 0.12 0.57 0.44 0.20 0.42 0.29 0.33 0.51 0.69 0.63 0.55 0.37
Adventure 0.74 1.00 0.31 0.44 0.53 0.35 0.12 0.49 0.52 0.18 0.36 0.33 0.30 0.50 0.64 0.50 0.48 0.31
Animation 0.29 0.31 1.00 0.59 0.42 0.23 0.13 0.33 0.26 0.20 0.21 0.37 0.18 0.29 0.28 0.25 0.25 0.16
Children's 0.39 0.44 0.59 1.00 0.51 0.26 0.13 0.40 0.43 0.14 0.25 0.45 0.23 0.37 0.38 0.32 0.29 0.24
Comedy 0.55 0.53 0.42 0.51 1.00 0.48 0.20 0.65 0.40 0.26 0.42 0.42 0.37 0.61 0.50 0.53 0.42 0.36
Crime 0.48 0.35 0.23 0.26 0.48 1.00 0.15 0.53 0.25 0.29 0.31 0.23 0.36 0.35 0.35 0.52 0.30 0.26
Documentary 0.12 0.12 0.13 0.13 0.20 0.15 1.00 0.24 0.08 0.12 0.10 0.13 0.13 0.15 0.11 0.15 0.12 0.08
Drama 0.57 0.49 0.33 0.40 0.65 0.53 0.24 1.00 0.37 0.29 0.36 0.38 0.42 0.59 0.49 0.59 0.54 0.36
Fantasy 0.44 0.52 0.26 0.43 0.40 0.25 0.08 0.37 1.00 0.12 0.25 0.26 0.20 0.35 0.48 0.32 0.30 0.21
Film-Noir 0.20 0.18 0.20 0.14 0.26 0.29 0.12 0.29 0.12 1.00 0.19 0.17 0.38 0.21 0.18 0.28 0.21 0.14
Horror 0.42 0.36 0.21 0.25 0.42 0.31 0.10 0.36 0.25 0.19 1.00 0.21 0.29 0.30 0.43 0.46 0.27 0.24
Musical 0.29 0.33 0.37 0.45 0.42 0.23 0.13 0.38 0.26 0.17 0.21 1.00 0.21 0.35 0.28 0.26 0.26 0.21
Mystery 0.33 0.30 0.18 0.23 0.37 0.36 0.13 0.42 0.20 0.38 0.29 0.21 1.00 0.32 0.31 0.50 0.26 0.19
Romance 0.51 0.50 0.29 0.37 0.61 0.35 0.15 0.59 0.35 0.21 0.30 0.35 0.32 1.00 0.42 0.47 0.46 0.30
Sci-Fi 0.69 0.64 0.28 0.38 0.50 0.35 0.11 0.49 0.48 0.18 0.43 0.28 0.31 0.42 1.00 0.54 0.44 0.30
Thriller 0.63 0.50 0.25 0.32 0.53 0.52 0.15 0.59 0.32 0.28 0.46 0.26 0.50 0.47 0.54 1.00 0.39 0.31
War 0.55 0.48 0.25 0.29 0.42 0.30 0.12 0.54 0.30 0.21 0.27 0.26 0.26 0.46 0.44 0.39 1.00 0.28
Western 0.37 0.31 0.16 0.24 0.36 0.26 0.08 0.36 0.21 0.14 0.24 0.21 0.19 0.30 0.30 0.31 0.28 1.00
Action Adventure Animation Children's Comedy Crime Documentary Drama Fantasy Film-Noir Horror Musical Mystery Romance Sci-Fi Thriller War Western
Genre
1.0
0.8
0.6
0.4
0.2

Preprocessing

1. User Bias:
   - Calculated user bias as the mean of all their ratings. This is necessary to consider in a user-user setting.
2. Defined function to create the movie-user interaction matrix

Our Approach

1. **Collaborative Filtering**
   - User-User based using Pearson's correlation
     - Unweighted correlation: Vanilla approach with only similarity index and user bias.
     - Weighted correlation: Added weights to movies based on their frequency.
   - Item-Item based, using K-Nearest Neighbours (KNN)

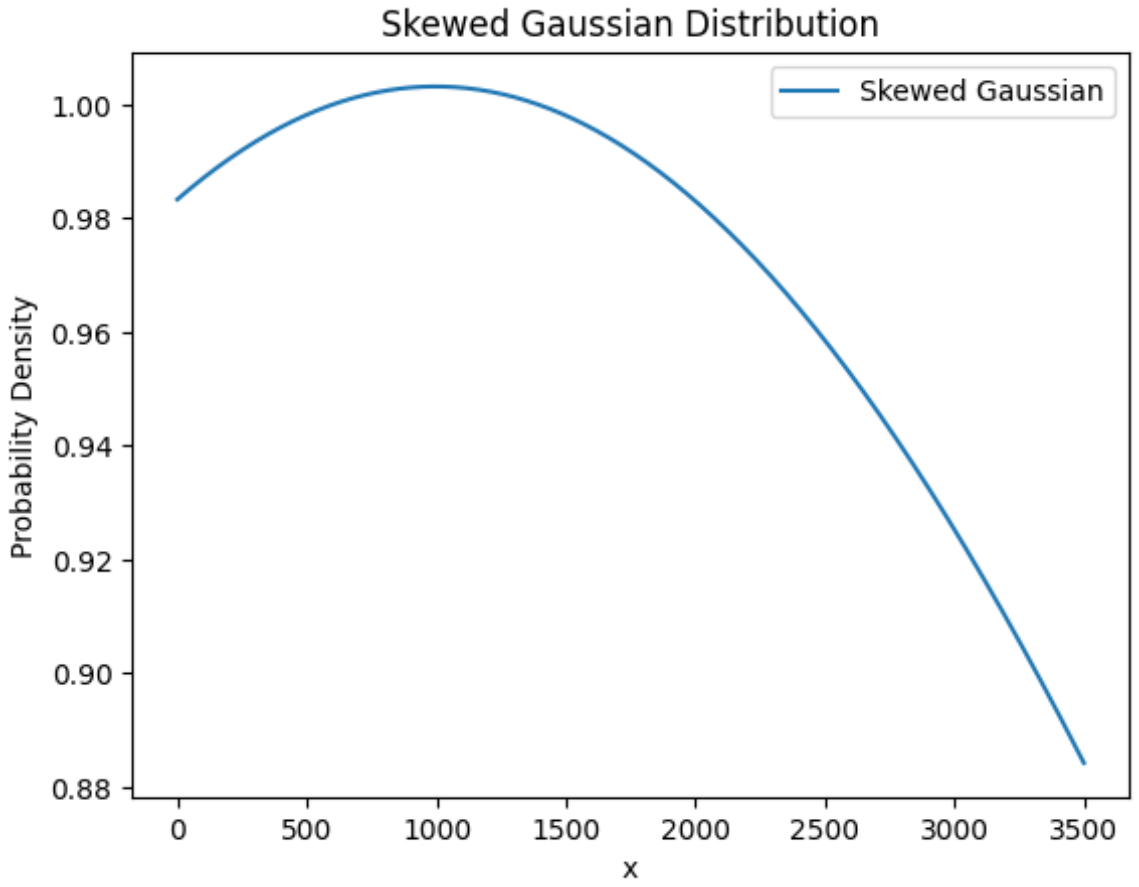
2. **SVD followed by KMeans++**

Naive Approach: User-User Collaborative Filtering using Pearson's correlation

1. **Find valid user pairs:**
   - Group users by those that have watched ***at least one*** movie the target user has watched.
   - Give priority to users that have more movies in common (by sorting based on size of group)
2. **Compute Similarity**
   - Calculate the Pearson correlation coefficient between each pair of users based on their ratings.
   - Return the similarity index for each valid user-user pair
3. **Find Closest users**
   - Select top 50 users that are closest to target user.
4. **Predict Ratings**
   - Weight each neighbor's rating by their correlation coefficient with the target user as per formula.
   - Normalize the ratings by subtracting the user's mean rating to handle biases.
5. **Final Recommendations**
   - Return top k movies based on reverse sort of predicted ratings

Adding Novelty: User-User Collaborative Filtering using Weighted Pearson's correlation

- In class we were introduced to the concept of giving higher weights to movies that were rated by fewer people similar to the Inverse Document Frequency (IDF) approach used in NLP.
- Instead of a IDF graph that is strictly decreasing we make use of a skewed gaussian with maximum value near 1000.
- Our hypothesis is that extremely low frequency movies don't say much about a user's taste (for eg. The users that have rated them might have done so because their friends were involved in the production)
- The skewed gaussian preserves the key ideals of the IDF approach while also taking into account, the possible vagueness of extremely low frequency movies.



Comparison (Unweighted vs Weighted)

Input:
New target user with 11 movies with main genres as ***Drama, Mystery and Romance****. Average rating of user was 4.3. We used both methods to compare the top 10 recommended movies.*

Naïve Pearson's Correlation

- Along with the same 2-3 genres we got few new genres as well like Comedy, Horror.
- This seemed out of place compared to the input genres

Weighted Pearson's Correlation

- Got related 2-3 genres of movies as per target user data along with 1-2 Comedy movies.
- Broadly Matched with overall target user profile.

Given the results, we decided to move ahead with the weighted approach as final recommendations for user-user method.

Item-Item Collaborative Filtering using KNN

1. **Calculate similarity between movies using KNN.**
   - In KNN, we use cosine similarity to find the closest set of movies to each movie by calculating the distances of all other movies and taking the nearest K.
2. **Predict rating for unwatched movies**
   - Find the closest movies based on similarity from previous step and calculate the weighted average of the ratings.
   - The weight used will be inverse of the distance of how close one movie is to the other.
3. **Recommend movies** which have highest predicted rating.

Comparision (User-User vs Item-Item)

Input: New target user with 11 movies with main genres as Drama, Mystery and Romance. Avg rating was 4.3. We used both methods to compare the top 10 recommended movies.

User-User Weighted

- Along with the same 2-3 genres we got few new genres as well like Comedy, Sci-Fi.
- Explainable and overall decent results, but lacked some aspect of Accuracy.

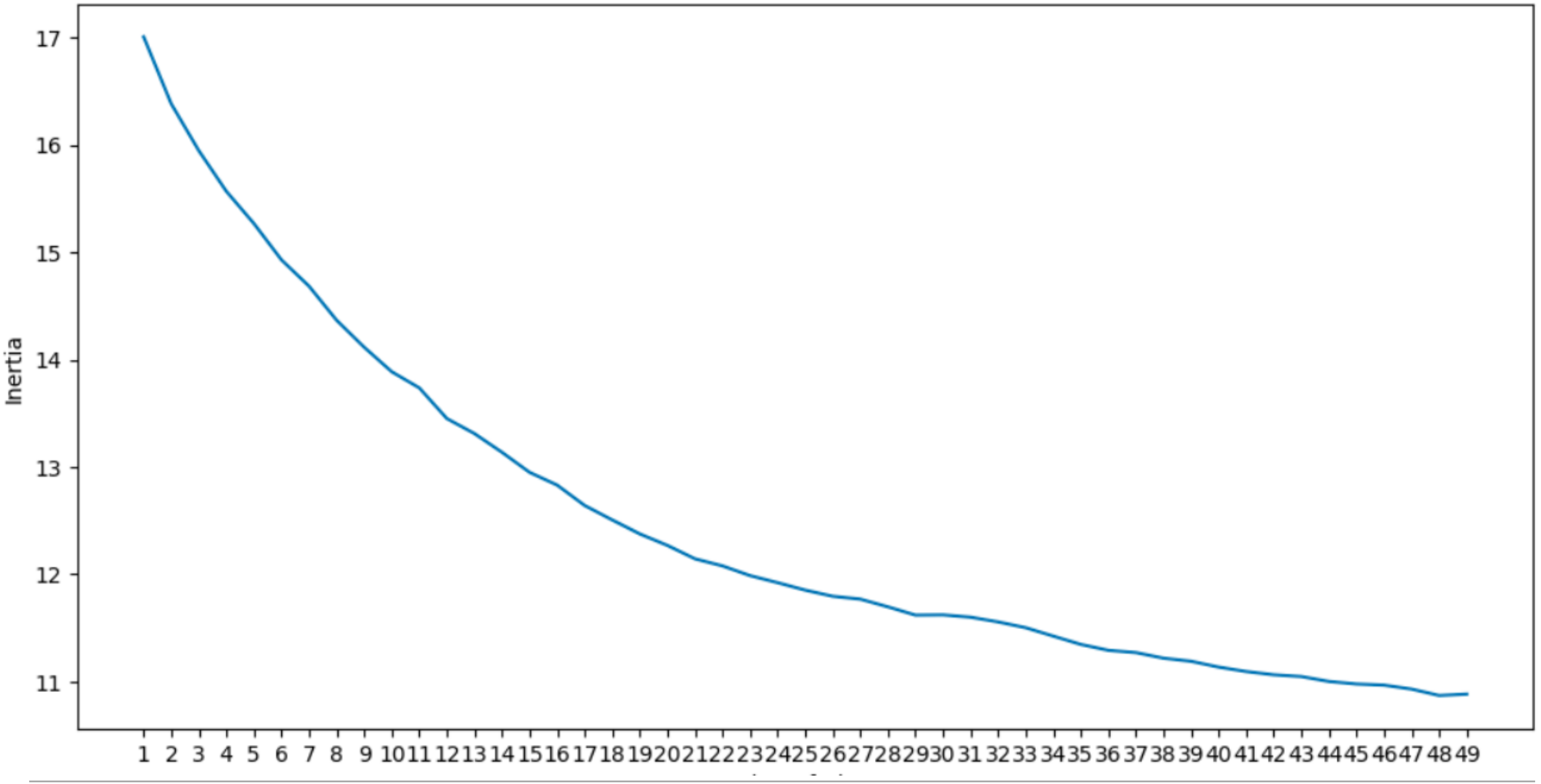
Item-Item KNN

- Got related 2-3 genres of movies as per target user data along with 1-2 Comedy movies.
- More accurate and desired results. For eg. On input of Toy Story, Godfather, recommendations contained Toy Story 2 and Godfather III.

Item-Item gave overall better predictions while user-user captured more novelty

SVD + K-Means++

1. **Formed a User-Genre Rating Matrix:**
   - The ratings for each user in a specific genre is computed by averaging the ratings of the movies that the user has watched within that genre
   - This process is performed for all users across all genres, and the results are stored in a pandas dataframe.
   - Null Values are filled by the average of the column (which is average of the genre)
2. **Computing SVD of the User-Genre Matrix:**
   - SVD of the user-genre matrix is calculated using the power iteration method. This involves 2 steps:
   - First, the covariance matrix ($X^T.X = V.\Sigma^2.V^T$) is used to compute the V matrix and Σ matrix (where $X = U\Sigma V^T$)
   - Second, the equation $XV=U\Sigma$ is used to find the U matrix.
   - After performing SVD on the user-genre matrix, we get the U, Σ and the V matrix.
3. **Performing K-Means++ on the 'U' matrix:**
   - Now we perform K-Means++ on the U matrix, which consists of user representations.
   - Inertia vs K Plot: (Next Slide)


Inertia
17
16
15
14
13
12
11
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49

SVD + K-Means++

3. **Performing K-Means++ on the 'U' Matrix:**
   - The Number of clusters (k) is chosen to be 16 while clustering the users.

4. **Predict Ratings:**
   - After a new user has supplied ratings for several movies, a suitable user representation is constructed based on the U matrix. This user representation is subsequently inputted into the k-means model to predict the cluster to which the new user is assigned.

   - Now, the ratings of this new user are predicted by passing the user's ratings through the weighted collaborative filtering, with the added constraint that similarity is only calculated with the users that are in the same cluster as the new user.

   - To compute the ratings, only the top-30 most similar users were considered.

   - Recommend movies based on the highest value of predicted ratings.

Comparision (User-User vs SVD + K-Means++)

Input: New target user with 11 movies with main genres as Drama, Mystery and Romance. Avg rating was 4.3. We used both methods to compare the top 10 recommended movies.

User-User Weighted

- Along with the same 2-3 genres we got few new genres as well like Comedy, Sci-Fi.
- Explainable and overall decent results, but lacked some aspect of Accuracy.

SVD+K-Means++

- Got movies with related genres as the input movies as well as children's genre and comedy genre
- Got comparable accuracy in terms of average ratings of movies that were suggested

SVD+K-Means++ was comparable in terms of accuracy and recommendations with the user-user collaborative filtering approach

Thank You