**CSE523 - Machine Learning**

**Movie Recommendation System using Machine Learning**

**Faculty - Prof. Mehul Raval**

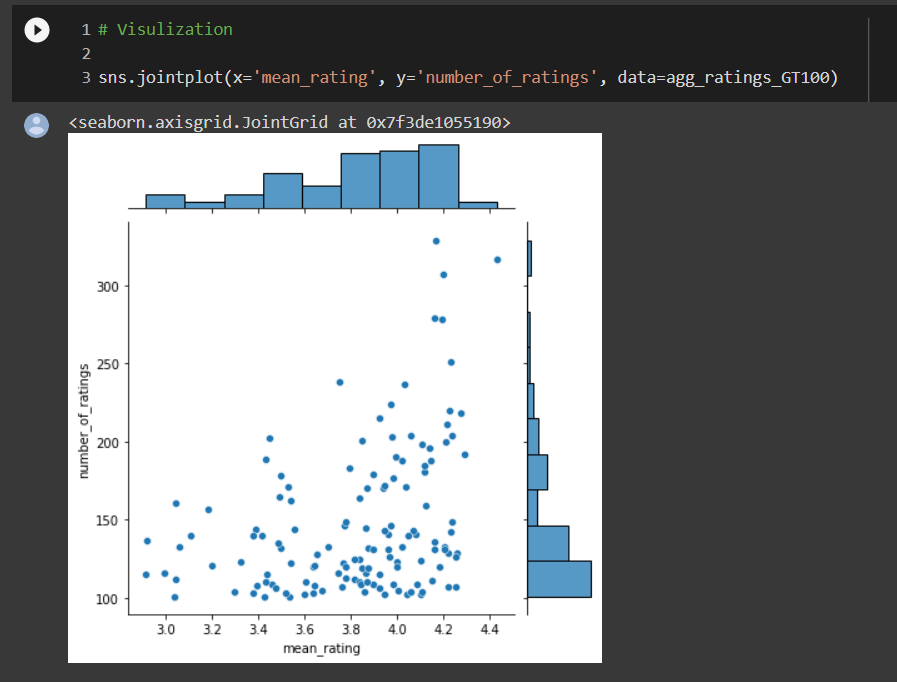
**Weekly Report 5**

**Submission Date: 18-03-23**

**Group: Tech Titans**

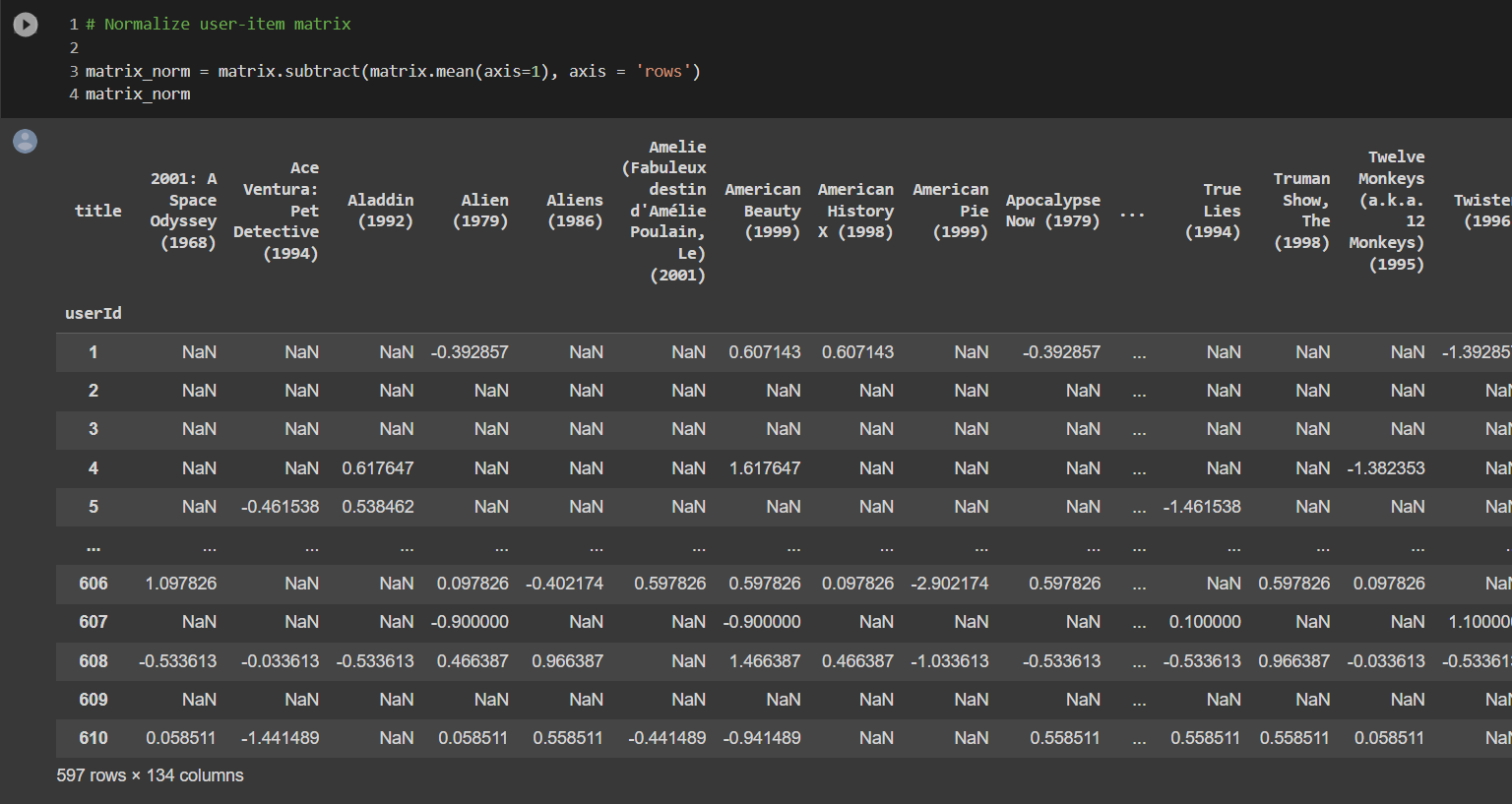
| **Name** | **Roll Number** | **Email** |
| --- | --- | --- |
| Jainam Shah | AU2040186 | jainam.s4@ahduni.edu.in |
| Yash Chotaliya | AU2040193 | yashkumar.c@ahduni.edu.in |
| Akshay Parmar | AU2040199 | akshay.p@ahduni.edu.in |
| Shubham Bhatt | AU2040206 | shubham.b1@ahduni.edu.in |

After aggregating our data by movies we keep only those movies that received more than 100 numbers of ratings to reduce the size of the data.



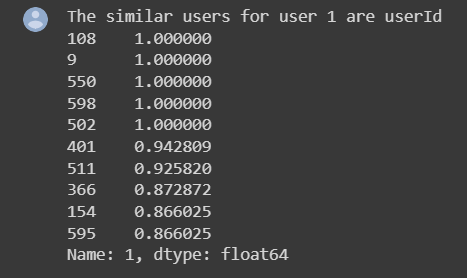
To get a better understanding of the mean rating we plotted the above chart. The x-axis represents the mean rating for each movie, and the y-axis represents the number of ratings each movie received. The plot displays a scatter plot of points, with each point representing a movie. The joint plot also includes a histogram for each axis showing the distribution of the data.

The graph will help to visualize the relationship between the mean rating and the number of ratings for movies in the dataset. It can give us an idea of whether highly rated movies tend to receive more number of ratings or whether movies with more number of ratings tend to have higher ratings. We can also use the plot to identify any outliers or patterns in the data.

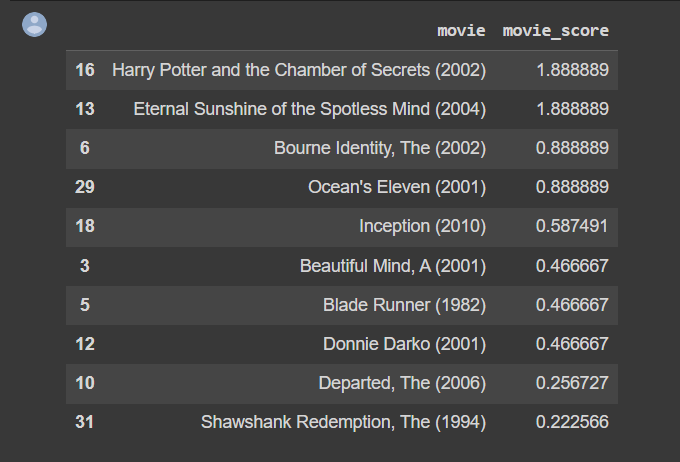


After this, we generated a user-movie matrix. The rows of that matrix show the user id and the columns show the name of a movie. Values of that matrix are a rating of a particular movie given by any user. Some users may have a tendency to rate movies more positively or negatively than others, resulting in their ratings being higher or lower than the average rating. By normalizing the user-movie matrix, we can adjust for these differences in rating scales and ensure that the similarity metrics we use to find similar users or movies are not biased by individual rating tendencies. So to get more accurate results we normalize the matrix by the average ratings of the user. Ratings which has a value less than the average user rating get a negative value and a value higher than the average user rating gets a positive value.

To find similar users we used Pearson correlation and cosine similarity respectively on our generated matrix. The Pearson correlation coefficient measures the linear correlation between two users' ratings for movies. The Pearson correlation is calculated as the covariance between two users' ratings divided by the product of their standard deviations. In movie recommendation systems, Pearson correlation is used to identify users who have similar preferences for movies. If two users have a high Pearson correlation coefficient, it means that they have rated movies in a similar way and are likely to enjoy similar movies. Once similar users are identified, the system recommends movies to a given user based on the movies liked by similar users. Unlike Pearson correlation, which measures the linear correlation between two users' ratings, cosine similarity measures the cosine of the angle between two users' rating vectors.



The values of the Pearson correlation coefficient can be from -1 to 1. 1 means two users are similar movie preferences and -1 means those two users have opposite preferences. Then we tried to pick the top 10 similar users for user 1. For that, we have to decide user similarity threshold. We decided if the Pearson correlation coefficient is more than 0.3 then it can be similar user to user 1.



To recommend the movie to a user we calculated the movie score. We list the movies along with their corresponding scores, which were calculated as the weighted average of user similarity score and the movie ratings given by similar users. The users with higher similarity receive higher weights in the movie ratings, which are weighted by similarity scores. The movies are sorted in descending order of their scores, with the highest-scoring movies appearing at the top of the table. These movies are the ones that are most likely to be enjoyed by the user based on their past movie ratings and the ratings of similar users. The similarity scores weigh the movie ratings, so the users with higher similarity get higher weights. Knowing the rank of the things is sufficient if the objective is to select the suggested items. We must, however, recalculate the movie score after adding the average user movie rating score if our objective is to estimate the user's rating by this we predict the approximate rating. So basically we increase the movie score by 4.39 since user 1 has an average movie rating of 4.39.