# Draft Idea for Evaluation of Neural Network to Recommend Movies

**The 20M MovieLens Dataset**

<https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset>

This data contains 20 million ratings made by 138k users about 27k movies. Each user has at least 20 movies rated. The data also includes genres for each movie and 466k tags applies by users.

**Ratings:** Each rating is on a scale from 0.5 to 5.0 stars in increments of 0.5. The rating histogram looks like this (mean = 3.5, std = 1.1, median = 3.5):

Chart, histogram

Description automatically generated

**Ratings per Movie and per User:**

Of the 27k movies, there are a small number of movies with many ratings. For example, the 50 most-rated movies (e.g. Pulp Fiction, Forrest Gump, The Shawshank Redemption) have 32k or more ratings each. Most movies have much fewer ratings. For example, the median movie has only 18 ratings and a quarter of movies in the dataset have only 1-3 ratings.

Of the 139k users, the data has a similar shape with a small number of users having a very high number of ratings. However, the “tail” of many users has more ratings each than the “tail” of movies. All users in this dataset have at least 20 movies rated and the median number of movies rated is 68.

|  |  |  |
| --- | --- | --- |
|  | **n\_ratings for movie** | **n\_ratings\_per\_user** |
| count | 26,729 | 138,493 |
| mean | 748 | 144 |
| std | 3,087 | 230 |
| min | 1 | 20 |
| 25th Percentile | 3 | 35 |
| Median | 18 | 68 |
| 75th Percentile | 205 | 155 |
| max | 67,310 | 9,254 |

**Neural Network Testing Idea**

To test the neural network, we should divide the user data into 75% training (used to create the model) and 25% testing to evaluate the performance of the model.

For each user in the testing set, we should simulate showing them the 30 most commonly rated movies in the entire data set and asking them to rate them on the 0.5 to 5.0 scale. To simulate this, we should assume that whatever ratings for these 30 movies exist are “known” while the user’s other ratings are “unknown.”

Based on these 30 known results we should use the neural network to recommend a list of 30 additional movies for them to watch next. These should be chosen from the approximately 1000 movies with 5000 ratings or more each. These 30 recommended movies will be some mix of movies with unknown ratings and no ratings from the same user. The accuracy of the model is the average rating of the unknown ratings that were recommended. The more the user likes the movies that were recommended the higher better the model’s performance will be rated.

**Example:**

Let’s say user #100 has 68 movies rated. Of the 30 most commonly rated movies in the entire set, user #100 rated 20 of the 30. This means that of their 68 ratings, 20 are now “known” and 48 are “unknown.” The 20 known ratings are used as the input of the neural network. The output is a list of 30 movies (not including the known movies nor any of the other 30 most common).

To evaluate the model, we take the 30 recommended movies and see how many of them are rated in the “unknown” ratings. For example, we find that 11 of the 30 were rated and 19 were not. For these 11, we find the average rating given by the user and that is the performance of the model. In other words, the better the model does at predicting movies the person actually liked (without knowing that in advance) the better the model is performing.

**Control Model:**

To evaluate the success of the model, we should compare the results of the actual neural network to a “control” model that naively makes recommendations without relying on the training data set. The control model should always recommend the 30 movies out of those with 5k or more ratings that have the highest average rating.