# Approach 3: Multi-armed Bandits

In a Multi-Armed Bandit (MAB) based recommendation system, successive alternation of exploration and exploitation of the different item options is performed to gain insight into which one of them yields the highest reward when recommended to many users.

## 3.1 MAB approach specific pre-processing

Using the already pre-processed data as described in section X, we perform further operations to tailor the data to the MAB approach.

As stated in the project description, we limit ourselves to four focal groups of users. Their specific attributes are listed in table X.

|  |  |  |
| --- | --- | --- |
|  | Gender: Male | Gender: Female |
| Age below 30 years (<) | Male, Young : `MY` | Female, Young: `FY` |
| Age above 30 years (>=) | Male, Old: `MO` | Female, Old: `FO` |

Based on this, we separate the training data set into four separate ones, that are individually used for the replay training. For the testing, we then assign the corresponding trained recommendation engine based on the user characteristics.

We have transformed the `rating` column into a `liked` column, by applying the function mapping values between one and three (inclusive) to `false`, while the ones from `four and `five` were mapped to `true`.

Regarding the attributes of the data frames, we have dropped all the rows for the items, where the sum of recommendations with this item is less than 200. This bound was set in the project description and is based on the idea, that the MAB exploration only is effective if there are enough training entries to learn from.

The following Columns where then dropped, because the MAB replay engine only uses the `user\_id`, `item\_id` and the `liked` value.

A screenshot of a computer screen

Description automatically generatedThis preprocessing resulted in the four focal group training data frames like the one shown in figure X.

## 3.2 MAB Model Training

To implement the replay training for the Multi-Armed Bandit recommendation engine, we based our trainer class on the tutorial of week 6.

Modifications include

* Combining the general MAB engine with the MAB specific one to have one
* Tracking the amount of tuples generate until one is found, that is also present in the recommendation training data.
* Tracking the amount of times, that a specific item is chosen.

Furthermore, we have implemented the possibility for a dynamic epsilon value. This allows us to modify the ratio of exploration while replaying. Reasoning behind this is, that in the beginning of the training, an already small epsilon value of e.g. 0.05 means, that in 95 percent of the tries, we exploit. This leads to very small and inefficient sampling of the item data sets and thus prolongs the time until the engine has adequate estimates of the reward fraction per item. Using a higher epsilon value in the beginning and then reducing it can separate the learning into an initial strong exploration, after which the exploitation begins.

We thought of implementing a three-stage epsilon dynamic, that starts with a constant high one, then reduces the value linearly until reaching a preset low value at which it stays constant. In order to have clearer analysis and potentially stronger comparison of values, we then went with a two stage approach, which changes the epsilon value from one constant one to another. Setting epsilon to the value one in the beginning would defeat the purpose of the MAB engine to potentially already earn more than average reward after a few rounds. Various tests were performed to find appropriate epsilon values.

## 3.3 MAB Recommendation Evaluation

We based our evaluation approach on this sentence in the task description “ Evaluation: Percentage of overlap between top 5 recommended movies with five movies that have the highest percentage of likes in the focal group”.

Looking at the data that we get in the test set, we observed that there are certain items, that are only appearing once in the entire test set. So using either a “liked/total” or “liked/disliked” fraction is unreasonable. That would not capture, whether it actually is one of the top 5 movies for that focal group. So we restricted ourselves to the top XXX % of movies based on ratings and then calculated the fraction “liked/total” for those. Then we picked the top 5 movies, which were:

X, X, X, X

## 3.4 Results and Performance Metrics

Using the above mentioned evaluation approach, we gained the following results:

XXX