# Recommender – Result Evaluation

## Base strategies

Generally, the base strategies have some good recommendations, but also include bad outliers.

E.g.: Recommending a Horror movie for Toy Story.

Algorithm based analysis:

* Genre: Works well, but the movies should also be sorted in some way -> or movies with many genres always dominate the predication.
* Actors/ Directors: Ok, idea for some actors/ directors with a narrow role cast (e.g.: Jason Statham) -> for diverse actors/ directors this may be worse (except if the user really likes them).
* User Ratings: On its own has a very high discovery value, but the recommendations can appear random, since the users may have only liked this movie well, but normally watches a different type of movie.
* TF-IDF for the summary: Has some good attempts, but without a good training process some metrics get valued way too high. For example, for Toy Story the main words used are the names since they appear multiple times. While this leads to suggesting the other movies of the series/ universe it also suggests completely unrelated movies with similar named characters (here: recommends a horror movie about a guy called Andy).
* Keyword similarity: This is theoretically the best metric, but the keywords are not always the best way to describe a movie and sometimes not that accurate. Also, it is not clear who gave the keywords: different users will probably assign different keywords to the same movie, so the keywords may not be 100% consistent.
* Popularity: Interesting metric to apply in addition to other metrics -> leads to “well received” movies being suggested more often

## Combinations

As we discussed in the base strategies the idea here is to just use multiple algorithms and then combine their relevance score. To accomplish this all our recommenders, return normalized relevance scores between 0 and 1.  
Simple Metadata: The idea here is to use only metadata that every movie “must have”: Genre, Popularity, Actors, and directors.

Here we did not apply any factors but just added or multiplied the scores to generate the final score. This already gives very good results, since the unwanted outliers get smoothed out by the other algorithms.

ALL algorithms: Here we used all the algorithms and assigned them a factor to weight them differently. We discovered that the results are solid, but it showed that even if we normalize the scores, it is hard to figure out how we should weigh the scores and if perhaps some scores are skewed after normalizing them. Example: if a movie has only 2 genre the score is either 0 0.5 or 1, which may or may not be a good thing if the score has such a big difference between the different options (may lead to the genre overlap being too impactful).

Another thing is some metrics may be similar and lead to overfitting of a specific variable.

## Ideas for a better recommender

During our project we found some ideas on how we would try to improve recommender systems for movies:

* Pipeline approach: our combination approach worked well, but perhaps using the genre overlap to filter out unrelated movies would have led to better results.
* Centralized meaningful keywords: Keywords should only be assigned by one party and the keywords should be meaningful and distinctly differ the movies.
* Archetype of the story: As mentioned in our presentation most of the stories follow one of 7 (or more) basic arches. E.g.: Something happens, and the main character must fight the enemy to survive. Link to a detailed description of the basic archetypes: <https://en.wikipedia.org/wiki/The_Seven_Basic_Plots>

Having this in addition of genres would create a new meaningful metric to use when recommending movies.

* Demographic information: Could be interesting because if the majority of the users of that demographic watched and like a movie this user would probably be interested in at least watching the movie, to be able to form their own opinion and participate in discussions.