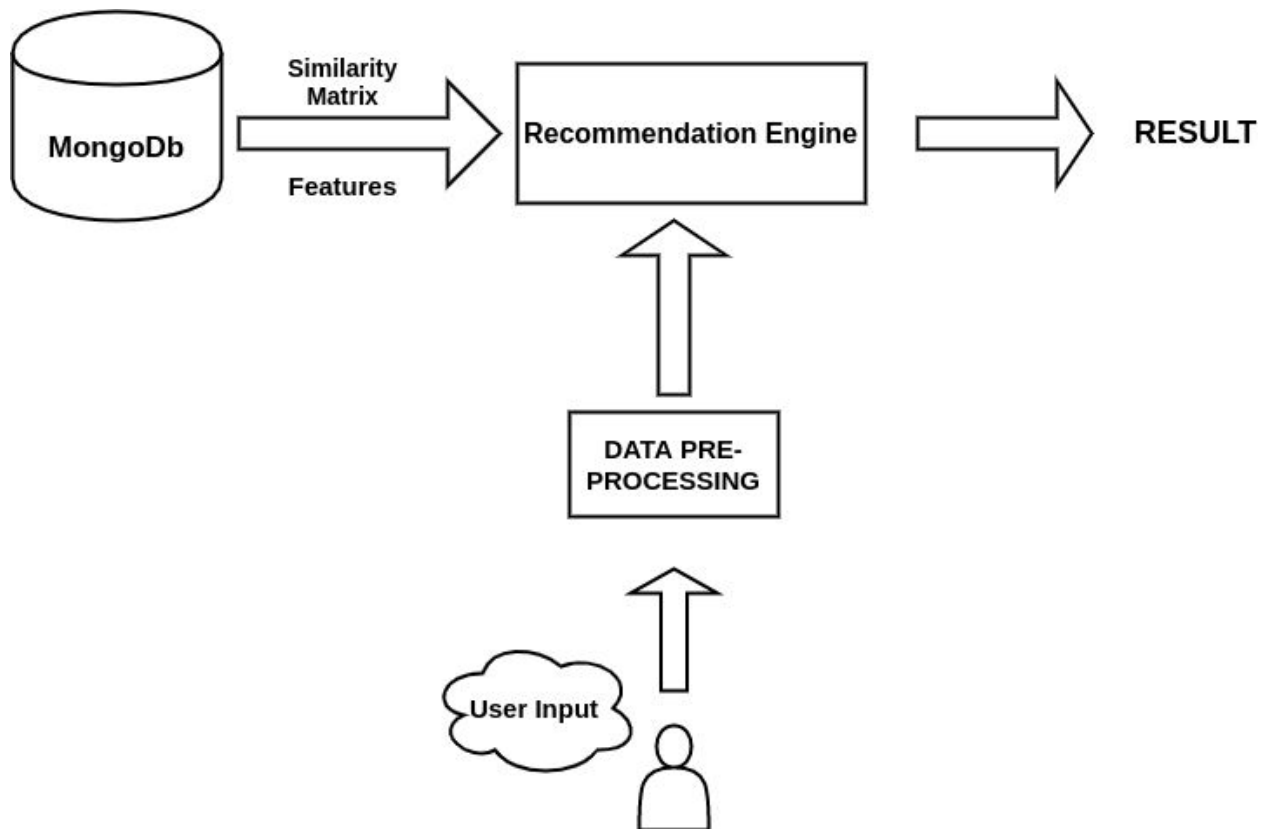


Recommendation System

Problem Statement: Recommend similar movies using the MongoDB database as a data source.

Architecture-



Data Source-

MongoDB database

Technologies Used-

Python, MongoDB

Metrics-

Linear Kernel (sklearn library)
Pearson Correlation

Data Understanding- Data consists of movies name with the corresponding summary of the movie with genres.

Approach

1. Popularity Based Recommendation System:

- we get our hand dirty with the movie dataset. we look carefully at the user ratings and think what's to be done?
- The answer that strikes first is the most popular Item. This is exactly what we will be doing.
- Technically this is super fast but it does come with a major drawback which is lack of personalization. The dataset that has many files but we will be looking at a few of these files mainly the ones which relate to movie ratings.
- In this approach, we recommend items that are liked by most users. This is a blazing fast and dirty approach and thus has a major drawback. The things are, there is no personalization involved with this approach. Such methods are widely used in news portals and work effectively because of the following points
- There is division by section so the user can look at the section of his interest.
- At a time there are only a few hot topics and there is a high chance that a user wants to read the news which is being read by most others

2. Collaborative Based Recommendation System:

User-Based Collaborative Filtering

Here we find lookalike customers (based on similarity) and offer products which the first customer's look-alike has chosen in the past. This algorithm is very effective but takes a lot of time and resources. It requires compute every customer pair information which takes time. Therefore, for big base platforms, this algorithm is hard to implement without a very strong parallelizable system.

1. Build a matrix of things each user bought viewed rated
2. compute similarity scores between users
3. find users similar to you
4. Recommend stuff they bought/viewed/rated that you haven't yet

Problems

1. People taste change
2. They are usually many more people than things
3. People do bad things

Item Based Collaborative Filtering

It is quite similar to the previous algorithm, but instead of finding customers look-alike, we try finding items look alike. Once we have items look-alike matrix, we can easily recommend alike items to customers who have purchased an item from the store. This algorithm is far less resource consuming than user-user collaborative filtering.

1. Find every pair of movies that were watched by the same person
2. Measure the similarity of rating across all the users who watched both
3. Sort movie then by similarity strength

Interesting fact

Item- Item Collaboration is extensively used in amazon or most of the E-commerce companies and they came out with it in great detail.

3. Content-Based Recommendation System:

- In this, we mainly focus on features or content of movie features according to that we recommend similar movies. For eg. Whenever we visit movie platforms like Netflix or amazon prime as soon as you click on movie we get similar movies based on the similarity of the movie.
- This is basic which has been used by almost every E-commerce platform company.