



# Architectural Components

Let's take a look at the architectural components of the recommendation system.

We'll cover the following



- Candidate generation
- Ranker
- Training data generation

It makes sense to consider generating the best recommendation from a large corpus of movies, as a multi-stage ranking problem. Let's see why.

We have a huge number of movies to choose from. Also, we require complex models to make great, personalized recommendations. However, if we try to run a complex model on the whole corpus, it would be inefficient in terms of execution time and computing resources usage.

Therefore, we split the recommendation task into two stages.

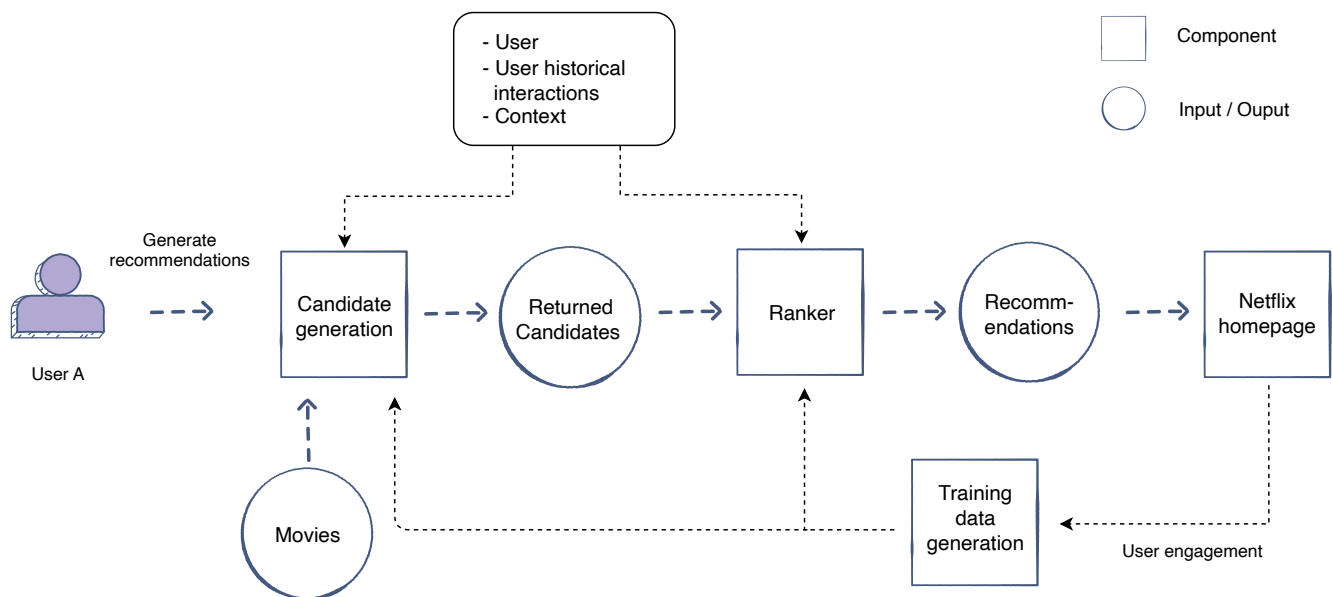
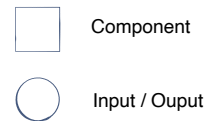
- Stage 1: Candidate generation
- Stage 2: Ranking of generated candidates

Stage 1 uses a simpler mechanism to sift through the entire corpus for possible recommendations. Stage 2 uses complex strategies only on the candidates given by stage 1 to come up with personalized recommendations.





Key



Architectural diagram of the recommendation system

## Candidate generation#

Candidate generation is the first step in coming up with recommendations for the user. This component uses several techniques to find out the best candidate movies/shows for a user, given the user's historical interactions with the media and context.

This component focuses on **higher recall**, meaning it focuses on gathering movies that might interest the user from all perspectives, e.g., media that is relevant based on historical user interests, trending locally, etc.

## Ranker#

The ranker component will score the candidate movies/shows generated by the candidate data generation component according to how interesting they

might be for the user.




 This component focuses on **higher precision**, i.e., it will focus on the ranking of the top k recommendations.

It will ensemble different scores given to a media by multiple candidate generation sources whose scores are not directly comparable. Moreover, it will also use a lot of other dense and sparse features to ensure highly relevant and personalized results.

## Training data generation#

The user's engagement with the recommendations on their Netflix homepage will help to generate training data for both, the ranker component and the candidate generation component.

 Find out more about training data generation for the ranker component and the candidate generation component in the upcoming lessons.

[< Back](#)

Metrics

[Next >](#)

Feature Engineering

 Completed

 Report an Issue



