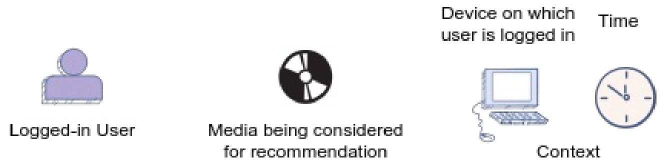
Feature Engineering

Let's engineer features for the candidate generation and ranking model.

We'll cover the following

To start the feature engineering process, we will first identify the main **actors** in the movie/show recommendation process:

1. Logged-in user
2. Movie/show
3. Context (e.g., season, time, etc.)


Logged-in User
Media being considered for recommendation
Device on which user is logged in
Time
Context

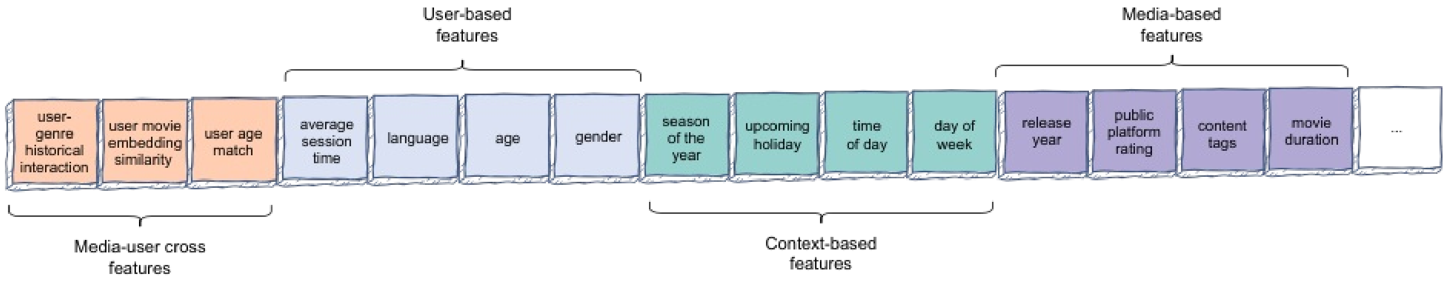
Main actors in media recommendation

Features

Now it’s time to generate features based on these actors. The features would fall into the following categories:

1. User-based features
2. Context-based features
3. Media-based features
4. Media-user cross features

A subset of the features is shown below.



Features in the training data row

User-based features

Let’s look at various aspects of the user that can serve as useful features for the recommendation model.

- **age**

  This feature will allow the model to learn the kind of content that is appropriate for different age groups and recommend media accordingly.

- **gender**

  The model will learn about gender-based preferences and recommend media accordingly.

- **language**

  This feature will record the language of the user. It may be used by the model to see if a movie is in the same language that the user speaks.

- **country**

  This feature will record the country of the user. Users from different geographical regions have different content preferences. This feature can help the model learn geographic preferences and tune recommendations accordingly.

- **average_session_time**

  This feature (user’s average session time) can tell whether the user likes to watch lengthy or short movies/shows.

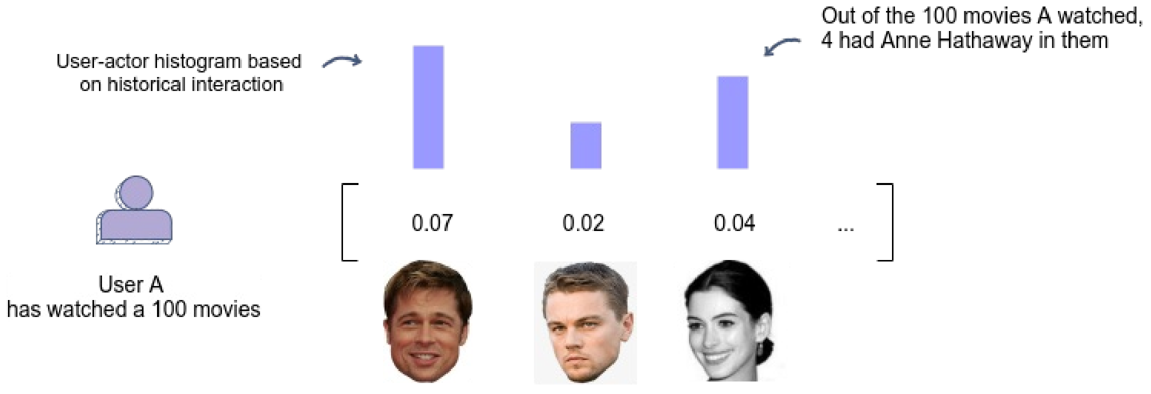
- **last_genre_watched**

  The genre of the last movie that a user has watched may serve as a hint for what they might like to watch next. For example, the model may discover a pattern that a user likes to watch thrillers or romantic movies.

The following are some user-based features (derived from historical interaction patterns) that have a *sparse representation*. The model can use these features to figure out user preferences.

- **user_actor_histogram**

  This feature would be a vector based on the histogram that shows the historical interaction between the active user and all actors in the media on Netflix. It will record the percentage of media that the user watched with each actor cast in it.


User-actor histogram based on historical interaction
Out of the 100 movies A watched, 4 had Anne Hathaway in them
0.07
0.02
0.04
...
User A has watched a 100 movies

User-actor histogram vector as a feature for the model

- **user_genre_histogram**

  This feature would be a vector based on the histogram that shows historical interaction between the active user and all the genres present on Netflix. It will record the percentage of media that the user watched belonging to each genre.

- **user_language_histogram**

  This feature would be a vector based on the histogram that shows historical interaction between the active user and all the languages in the media on Netflix. It will record the percentage of media in each language that the user watched.

Context-based features

Making context-aware recommendations can improve the user's experience. The following are some features that aim to capture the contextual information.

- **season_of_the_year**

  User preferences may be patterned according to the four seasons of the year. This feature will record the season during which a person watched the media. For instance, let's say a person watched a movie tagged "summertime" (by the Netflix tagger) during the *summer season*. Therefore, the model can learn that people prefer "summertime" movies during the summer season.

- **upcoming_holiday**

  This feature will record the upcoming holiday. People tend to watch holiday-themed content as the different holidays approach. For instance, Netflix tweeted that fifty-three people had watched the movie "A Christmas Prince" daily for eighteen days before the Christmas holiday. Holidays will be region-specific as well.

- **days_to_upcoming_holiday**

  It is useful to see how many days before a holiday the users started watching holiday-themed content. The model can infer how many days before a particular holiday users should be recommended holiday-themed media.

- **time_of_day**

  A user might watch different content based on the time of the day as well.

- **day_of_week**

  User watch patterns also tend to vary along the week. For example, it has been observed that users prefer watching shows throughout the week and enjoy movies on the weekend.

- **device**

  It can be beneficial to observe the device on which the person is viewing content. A potential observation could be that users tend to watch content for shorter periods on their mobile when they are busy. They usually chose to watch on their TV when they have more free time. So, they watch media for a longer period consecutively on their TV. Hence, we can recommend shows with short episodes when a user logs in from their mobile device and longer movies when they log in from their TV.

Media-based features

We can create a lot of useful features from the media's metadata.

- **public-platform-rating**

  This feature would tell the public's opinion, such as IMDb/rotten tomatoes rating, on a movie. A movie may launch on Netflix well after its release. Therefore, these ratings can predict how the users will receive the movie after it becomes available on Netflix.

- **revenue**

  We can also add the revenue generated by a movie before it came to Netflix. This feature also helps the model to figure out the movie's popularity.

- **time_passed_since_release_date**

  The feature will tell how much time has elapsed since the movie's release date.

- **time_on_platform**

  It is also beneficial to record how long a media has been present on Netflix.

- **media_watch_history**

  Media's watch history (number of times the media was watched) can indicate its popularity. Some users might like to stay on top of trends and focus on only watching popular movies. They can be recommended popular media. Others might like less discovered indie movies more. They can be recommended less watched movies that had good implicit feedback (the user watched the whole movie and did not leave it midway). The model can learn these patterns with the help of this feature.

We can look at the media's watch history for different time intervals as well. For instance, we can have the following features:

- media_watch_history_last_12_hrs
- media_watch_history_last_24_hrs

> The media-based features listed above can collectively tell the model that a particular media is a blockbuster, and many people would be interested in watching it. For example, if a movie generates a large revenue, has a good IMDb rating, came to the platform 24 hours ago, and a lot of people have watched it, then it is definitely a blockbuster.

- **genre**

  This feature records the primary genre of content, e.g., comedy, action, documentaries, classics, drama, animated, and so on.

- **movie_duration**

This feature tells the movie duration. The model may use it in combination with other features to learn that a user may prefer shorter movies due to their busy lifestyle or vice versa.

- **content_set_time_period**

  This feature describes the time period in which the movie/show was set in. For example, it may show that the user prefers shows that are set in the '90s.

- **content_tags**

  Netflix has hired people to watch movies and shows to create extremely detailed, descriptive, and specific tags for the movies/shows that capture the nuances in the content. For instance, media can be tagged as a "Visually-striking nostalgic movie". These tags greatly help the model understand the taste of different users and find the similarity between the user's taste and the movies.

- **show_season_number**

  If the media is a show with multiple seasons, this feature can tell the model whether a user likes shows with fewer seasons or more.

- **country_of_origin**

  This feature holds the country in which the content was produced.

- **release_country**

  This feature holds the country where the content was released.

- **release_year**

  This feature shows the year of theatrical release, original broadcast date or DVD release date.

- **release_type**

  This feature shows whether the content had a theatrical, broadcast, DVD, or streaming release.

- **maturity_rating**

  This feature contains the maturity rating of the media with respect to the territory (geographical region). The model may use it along with a user's age to recommend appropriate movies.

Media-user cross features

In order to learn the users' preferences, representing their historical interactions with media as features is very important. For instance, if a user watches a lot of Christopher Nolan movies, that would give us a lot of information about what kind of movies the user likes. Some of these interaction-based features are as follows:

User-genre historical interaction features

These features represent the percentage of movies that the user watched with the same genre as the movie under consideration. This percentage is calculated for different time intervals to cater to the dynamic nature of user preferences.

- **user_genre_historical_interaction_3months**

The percentage of movies that the user watched with the same genre as the movie under consideration in the last 3 months. For example, if the user watched 6 comedy movies out of the 12 he/she watched in the last 3 months, then the feature value will be:

$\frac{6}{12}$= 0.5 or 50%

This feature shows a more recent trend in the user's preference for genres as compared to the following feature.

- **user_genre_historical_interaction_1year**

  This is the same feature as above but calculated for the time interval of one year. It shows a more long term trend in the relationship between the user and genre.

- **user_and_movie_embedding_similarity**

  Netflix has hired people to watch movies and shows to create incredibly detailed, descriptive, and specific tags for the movies/shows that capture the nuances in the content. For instance, media can be tagged as "Visually-striking nostalgic movie".

  You can have a user embedding based on the tags of movies that the user has interacted with and a media embedding based on its tags. The dot product similarity between these two embeddings can also serve as a feature.

- **user_actor**

  This feature tells the percentage of media that the user has watched, which has the same cast (actors) as that of the media under consideration for recommendation.

- **user_director**

  This feature tells the percentage of movies that the user has watched with the same director as the movie under consideration.

- **user_language_match**

  This feature matches the user's language and the media's language.

- **user_age_match**

  You will keep a record of the age bracket that has mostly viewed a certain media. This feature will see if the user watching a particular movie/show falls into the same age bracket. For instance, movie A is mostly (80% of the times) watched by people who are 40+. Now, while considering movie A for a recommendation, this feature will see if the user is 40+ or not.

Some ***sparse features*** are described below. Each of them can show popular trends in their respective domains and also the preferences of individual users. We will go over how these sparse features are used in the ranking chapter. You can also go over the embedding chapter (https://www.educative.io/collection/page/10370001/6237869033127936/6130870193750016) about how to generate vector representation of this sparse data to use them in machine learning models.

- **movie_id**

Popular movie IDs are be repeated frequently.

- **title_of_media**

  This feature holds the title of the movie or the TVV series.

- **synopsis**

  This feature holds the synopsis or summary of the content.

- **original_title**

  This feature holds the original title of the movie in its original language. The media may be released for a different country with a different title keeping in view the preference of the nationals. For example, Japanese/Korean movies/shows are released for English speaking countries with English titles as well.

- **distributor**

  A particular distributor may be selecting very good quality content, and hence users might prefer content from that distributor.

- **creator**

  This feature contains the creator/s of the content.

- **original_language**

  This feature holds the original spoken language of the content. If multiple, you can record the choose the majority language.

- **director**

  This feature holds the director/s of the content. This feature can indicate directors who are widely popular, such as Steven Spielberg, and it can also showcase the individual preference of users.

- **first_release_year**

  This feature holds the year in which content had its first release anywhere (this is different from production year).

- **music_composer**

  The music in a show or a film's score can greatly enhance the storytelling aspect. Users may fancy the work of a particular composer and may be more drawn to their work.

- **actors**

  This feature includes the cast of the movie/show.

Minimize

Sparse features like the distributor and synopsis can be made into dense features.

- Dense representation of distributor feature: Does the distributor of this movie fall in the list of top ten distributors?
- Dense representation of synopsis feature: Perform text summarization of synopsis to pull out the keywords.

However, this process requires extra effort on the engineer's part so you will let the neural network figure out the relationships (This approach will be discussed in the ranking lesson).

← Back
Architectural Components

Next →
Candidate Generation

☑ Mark as Completed

Report an Issue

Ask a Question
(https://discuss.educative.io/tag/feature-engineering__recommendation-system__grokking-the-machine-learning-interview)