Training Data Generation

Let's generate training data for the recommendation task with respect to implicit user feedback.

We'll cover the following

As mentioned previously, you will build your model on implicit feedback from the user. You will look at how a user interacts with media recommendations to generate positive and negative training examples.

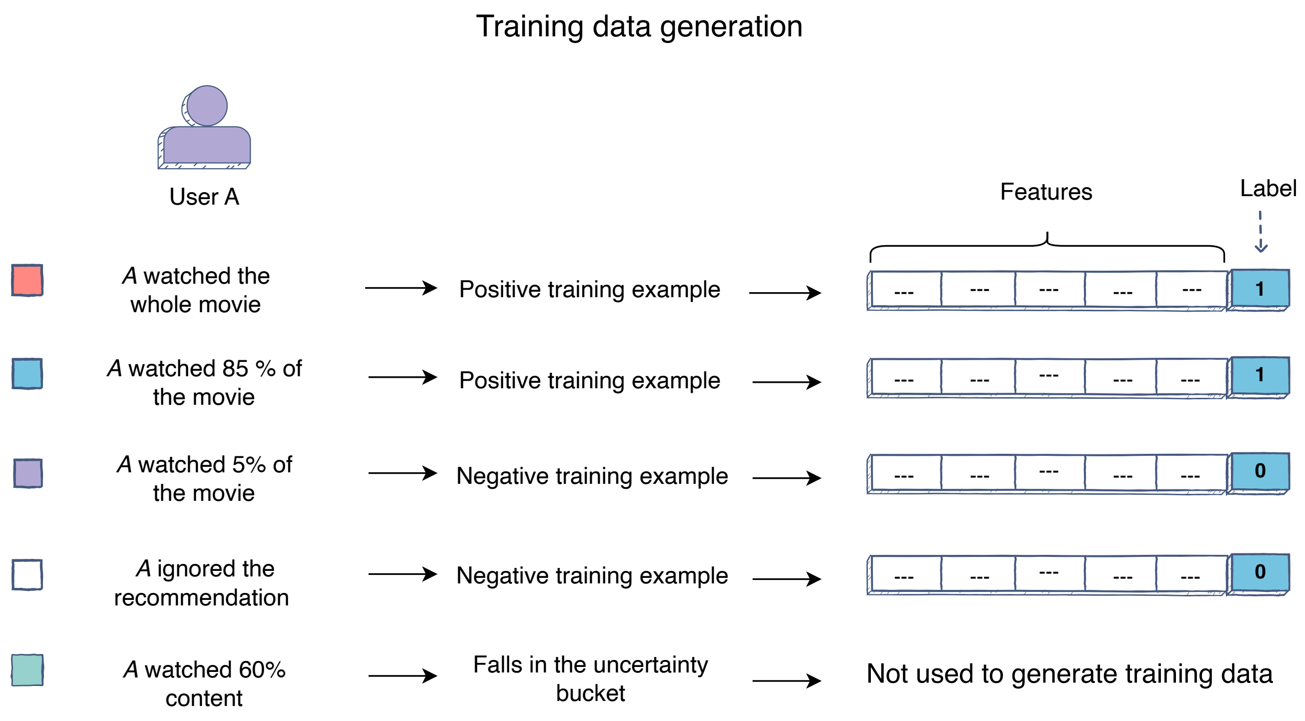
Generating training examples#

One way of interpreting *user actions* as positive and negative training examples is based on the duration for which the user watched a particular show/movie. You take positive examples as ones where the user ended up watching most of a recommended movie/show, i.e., watched 80% or more. You take negative examples, again where we are confident that the user ignored a movie/show, i.e., watched 10% or less.

If the percentage of a movie/show watched by the user falls between 10% and 80%, you will put it in the uncertainty bucket. This percentage is not clearly indicative of a user's like or dislike, so you ignore such examples. For instance, let's say a user watched 55% of a movie. This could be considered a positive example considering that they liked it enough to wa m y.

However, it could be that a lot of people had recommended it to them, so they wanted to see what all the hype was about by at least watching it halfway through. However, they, ultimately, decided that it was not according to their liking.

Hence, to avoid these kinds of misinterpretations, you label examples as positive and negative only when you are certain about it to a higher degree.


Training data generation
User A
Features
Label
A watched the whole movie
Positive training example
1
A watched 85 % of the movie
Positive training example
1
A watched 5% of the movie
Negative training example
0
A ignored the recommendation
Negative training example
0
A watched 60% content
Falls in the uncertainty bucket
Not used to generate training data

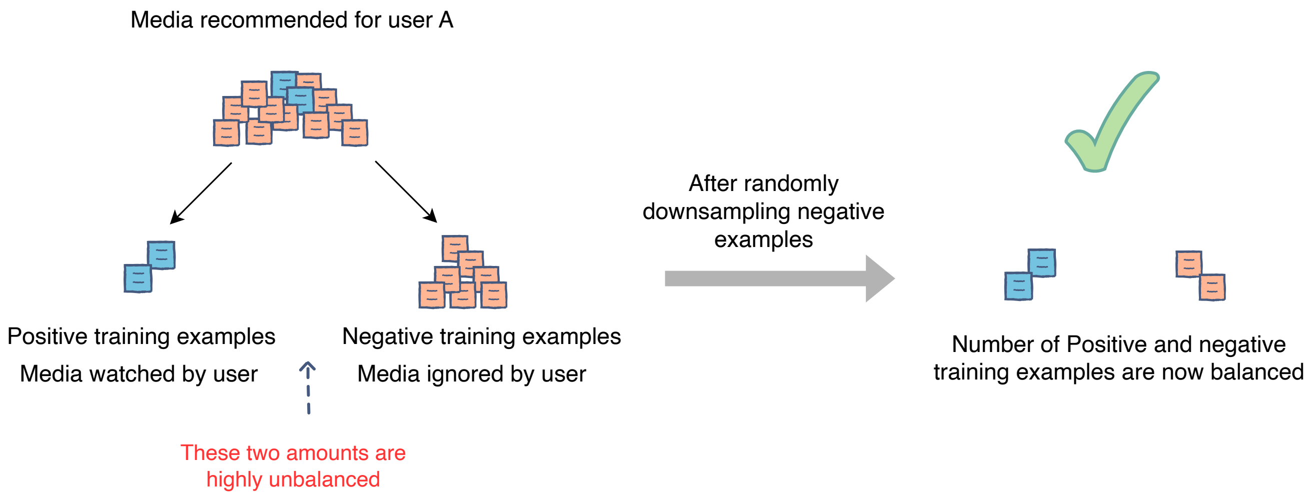
Positive and negative training examples being generated from user actions

Balancing positive and negative training examples#

Each time a user logs in, Netflix provides *a lot of recommendations*. The user cannot watch all of them. Yes, people do binge-watch on Netflix, but still, th does not improve the positive to negative training examples ratio significantly. Therefore, you have a lot more negative trainin m th

positive ones. To balance the ratio of positive and negative training samples, you can **randomly downsample** the negative examples.

> 📝 We balance the negative and positive examples to prevent classifier from favouring the outcome that has more examples.


Media recommended for user A
Positive training examples
Media watched by user
Negative training examples
Media ignored by user
These two amounts are highly unbalanced
After randomly downsampling negative examples
Number of Positive and negative training examples are now balanced

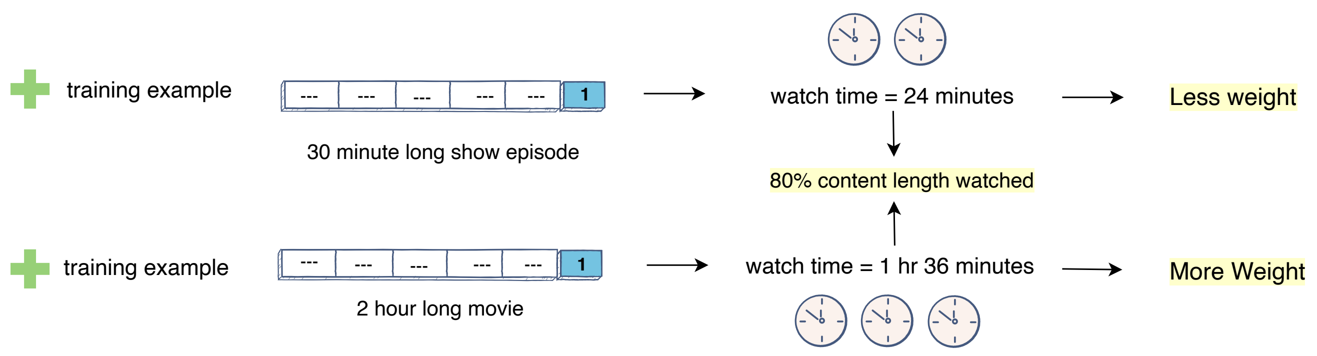
Balancing positive and negative training examples

Weighting training examples#

Based on our training data discussion so far, all of the training examples are weighted equally, i.e., all have a weight of 1. According to Netflix's business objectives, the main goal could be to increase the time a user spends on the platform.

One way to incentivize your model to focus more on examples that have a higher contribution to the session watch time is to weight examples based on their contribution to session time. Here, you are assuming that your prediction model's optimization function utilizes weight per example in its objective.

Reweighing training data based on watch time



In the diagram above, you have two positive training examples. The first is a thirty-minute long show episode while the second is a two-hour long movie. The user watches 80% of both media. For the show, this equals twenty-four minutes, but for the movie, it means one hour and thirty-six minutes. You would assign more weight to the second example than the first one so that your model learns which kinds of media increases the session watch time.

> 📝 One caveat of utilizing these weights is that the model might only recommend content with a longer watch time. So, it's important to choose weights such that we are not solely focused on watch time. We should find the right balance between user satisfaction and watch time, based on our online A/B experiments.

Train test split#

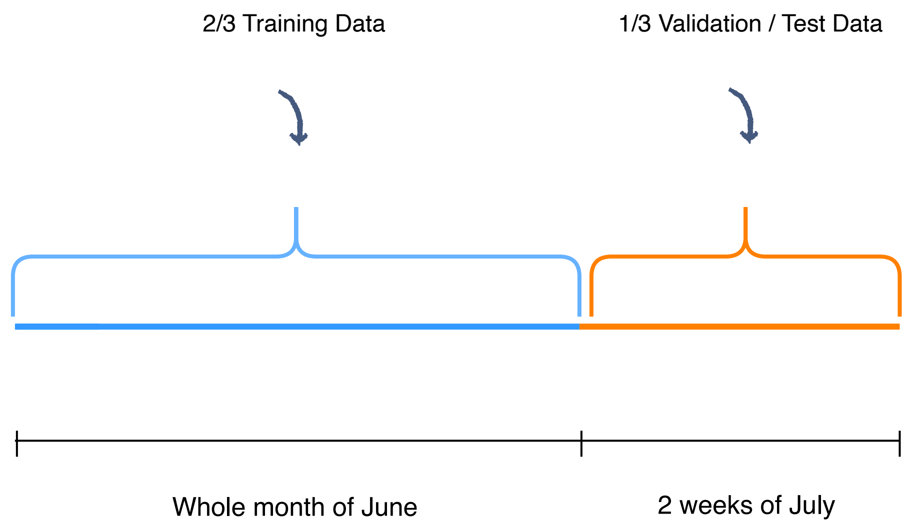
Now, it's time to split your training data for training and then testing your models. While doing so, you need to be mindful of the fact that the user's interaction patterns may differ throughout the week. Hence, you will use the interaction with recommendations throughout a week to capture all of the patterns during model training.

You can randomly select $\frac{2}{3}^{rd}$, or 66.6%, of the training data rows generated and utilize them for training purposes. The rest of the $\frac{1}{3}^{rd}$, or 33.3%, can be used for validation and model testing.

However, this random splitting defeats the purpose of training the model on a whole week's data. Also, the data has a time dimension, i.e., you know the interaction on previous recommendations, and you want to predict the interaction with future recommendations. Hence, you will train the model on data from one time interval and validate it on the data from its succeeding time interval. This will give you a more accurate picture of how your model will perform in a real scenario.

> 📝 You are building models with the intent to forecast the future.

In the following illustration, you are training the model using data generated from the month of June and using data generated in the first and second week of July for validation and testing purposes.



Splitting data for training, validation, and testing

Even though Netflix has millions of subscribers, on average they may watch a maximum of 3 movies/1 show per week. Therefore, to generate a sufficient amount of data, you need to increase the time span over which you gather the data. Hence, we are using an entire month's data to train your model.

Back
Candidate Generation

Next
Ranking

Completed

Report an Issue