Metrics

Let's look at the online and offline metrics used to judge the performance of the recommendation system.

We'll cover the following

In this lesson, you will look at different metrics that you can use to gauge the performance of the movie/show recommendation system.

Types of metrics#

Like any other optimization problem, there are two types of metrics to measure the success of a movie/show recommendation system:

1. **Online metrics**

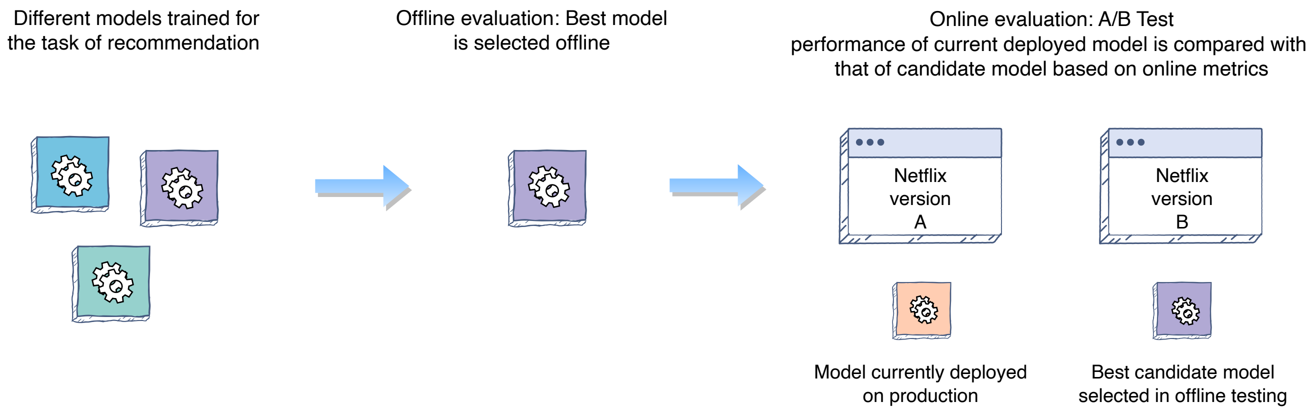
   Online metrics are used to see the system's performance through online evaluations on live data during an A/B test.

evaluations on live data during an A/B test.

2. **Offline metrics**

   Offline metrics are used in offline evaluations, which simulate the model’s performance in the production environment.

We might train multiple models and tune and test them *offline* with the *held-out* test data (historical interaction of users with recommended media). If its performance gain is worth the engineering effort to bring it into a production environment, the best performing model will then be selected for an *online* A/B test on live data.


Different models trained for the task of recommendation
Offline evaluation: Best model is selected offline
Online evaluation: A/B Test performance of current deployed model is compared with that of candidate model based on online metrics
Netflix version A
Netflix version B
Model currently deployed on production
Best candidate model selected in offline testing

> 📝 If a model performs well in an offline test but not in the online test, we need to think about where we went wrong. For instance, we need to consider whether our data was biased or whether we split the data appropriately for train and test.
>
> Have a look at the lesson about online experimentation.

Driving online metrics in the right direction is the ultimate goal of the recommendation system.

Online metrics#

The following are some options for online metrics that we have for the system. Let's go over each of them and discuss which one makes the most sense to be used as the key online success indicator.

Engagement rate#

The success of the recommendation system is directly proportional to the number of recommendations that the user engages with. So, the engagement rate ($\frac{sessions\ with\ clicks}{total\ number\ of\ sessions}$) can help us measure it. However, the user might click on a recommended movie but does not find it interesting enough to complete watching it. Therefore, only measuring the engagement rate with the recommendations provides an incomplete picture.

Videos watched#

To take into account the unsuccessful clicks on the movie/show recommendations, we can also consider the average number of videos that the user has watched. We should only count videos that the user has spent at least a significant time watching (e.g., more than two minutes).

However, this metric can be problematic when it comes to the user starting to watch movie/series recommendations but not finding them interesting enough to finish them.

Series generally have several seasons and episodes, so watching one episode and then not continuing is also an indication of the user not finding the content interesting. So, just measuring the average number of videos watched might miss out on overall user satisfaction with the recommended content.

Session watch time#

Session watch time measures the overall time a user spends watching content based on recommendations in a session. The key measurement aspect here is that the user is able to find a meaningful recommendation in a session such that they spend significant time watching it.

To illustrate intuitively on why session watch time is a better metric than engagement rate and videos watched, let’s consider an example of two users, A and B. User A engages with five recommendations, spends ten minutes watching three of them and then ends the session. One the other end, user B engages with two recommendations, spends five minutes on first and then ninety minutes on the second recommendation. Although user A engaged with more content, user B’s session is clearly more successful as they found something interesting to watch.

Therefore, measuring session watch time, which is indicative of the session success, is a good metric to track online for the movie recommendation system.

Offline metrics#

The purpose of building an offline measurement set is to be able to evaluate our new models quickly. Offline metrics should be able to tell us whether new models will improve the quality of the recommendations or not.

Can we build an ideal set of documents that will allow us to measure recommendation set quality? One way of doing this could be to look at the movies/series that the user has completely watched and see if your recommendation system gets it right using historical data.

Once we have the set of movies/series that we can confidently say should b on the user’s recommendation list, we can use the following offline metrics to measure the quality of your recommendation system.

mAP @ N#

One such metric is the Mean Average Precision(mAP @ N).

> 📝 N = length of the recommendation list

Let’s go over how this metric is computed so you can build intuition on why it’s good to measure the offline quality.

Precision measures the ratio between the relevant recommendations and total recommendations in the movie recommendation list. It will be calculated as follows:

$$P = \frac{number\ of\ relevant\ recommendations}{total\ number\ of\ recommendations}$$

We can observe that precision alone does not reward the early placement of relevant items on the list. However, if we calculate the precision of the subset of recommendations up until each position, **k** (k = 1 to N), on the list and take their weighted average, we will achieve our goal. Let’s see how.

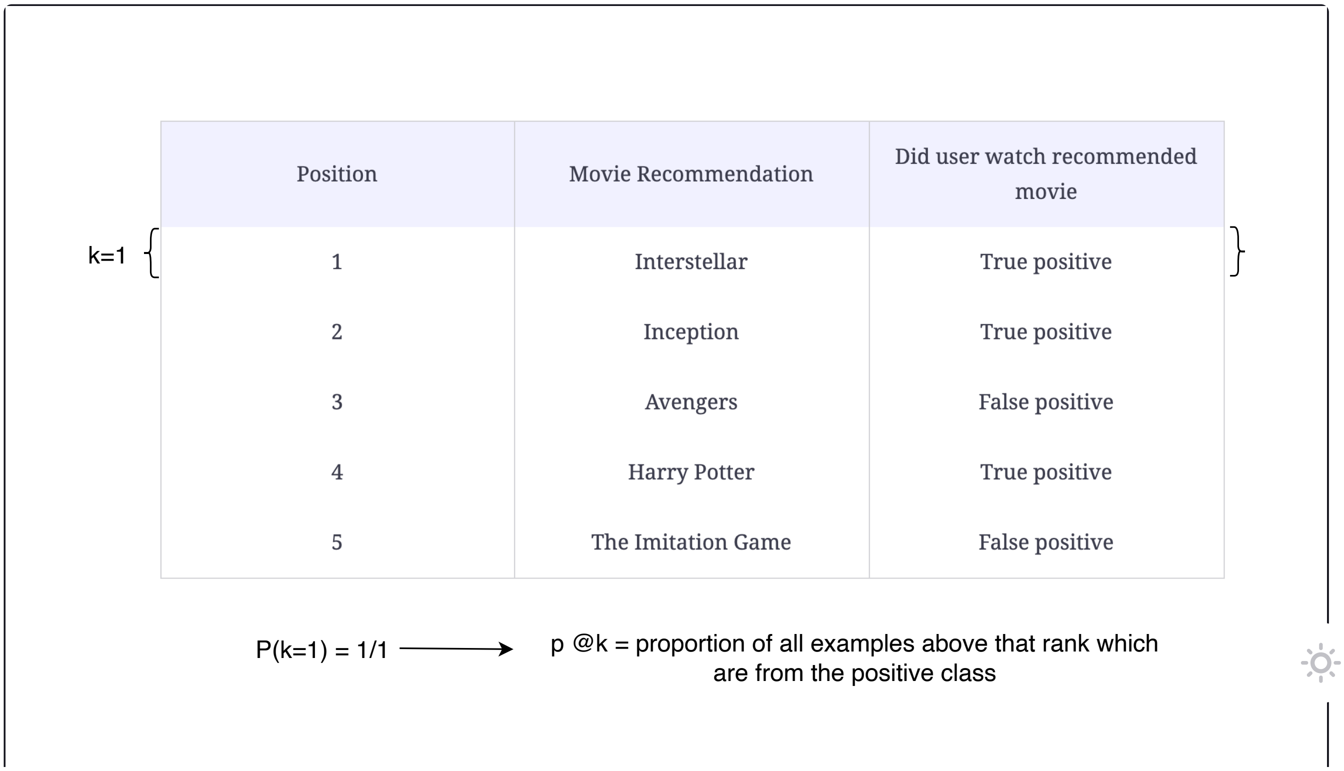
Assume the following:

1. The system recommended **N** = 5 movies.
2. The user watched three movies from this recommendation list and ignored the other two.
3. Among all the possible movies that the system could have recommended (available on the Netflix platform), only **m** = 10 are actually relevant to the user (historical data).

Position	Movie Recommendation	Did user watch recommended movie

Position	Movie Recommendation	Did us at - ommended movie
1	Interstellar	True positive
2	Inception	True positive
3	Avengers	False positive
4	Harry Potter	True positive
5	The Imitation Game	False positive

In the following diagram, we calculate the precision of recommendation subsets up to each position, k, from 1 to 5.



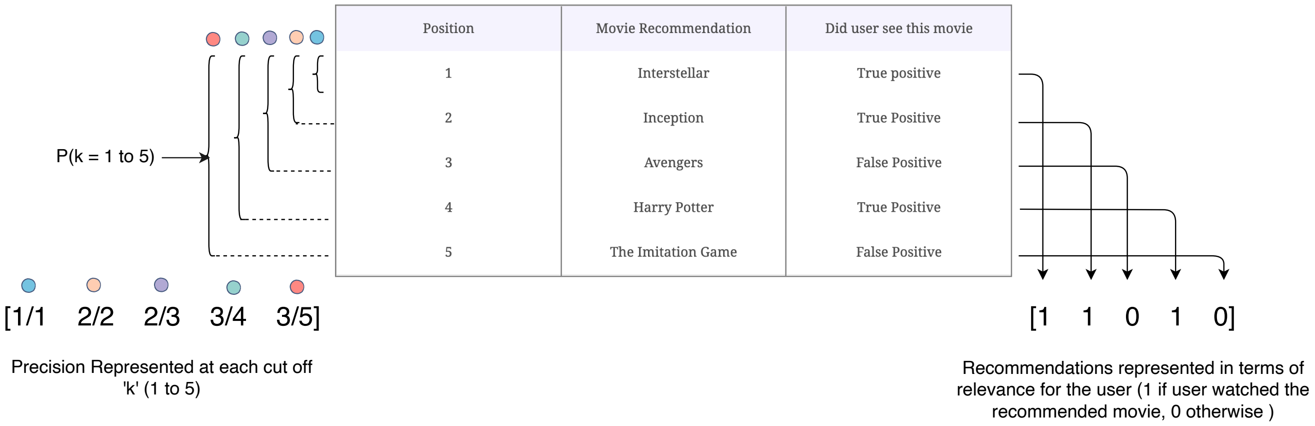
Position	Movie Recommendation	Did user watch recommended movie
1	Interstellar	True positive
2	Inception	True positive
3	Avengers	False positive
4	Harry Potter	True positive
5	The Imitation Game	False positive

k=1
P(k=1) = 1/1
p @k = proportion of all examples above that rank which are from the positive class

Calculating precision up to cutoff k = 1

1 of 5

The movie recommendation list and precisions at each cutoff “k” (1 to 5) can be represented as follows.


P(k = 1 to 5)
[1/1 2/2 2/3 3/4 3/5]
Precision Represented at each cut off 'k' (1 to 5)
[1 1 0 1 0]
Recommendations represented in terms of relevance for the user (1 if user watched the recommended movie, 0 otherwise)

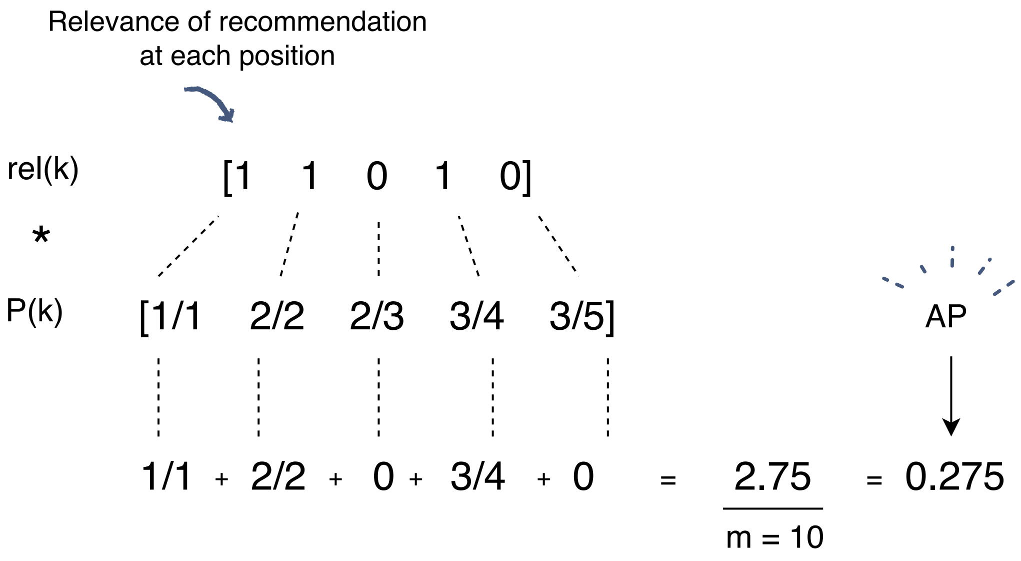
Position	Movie Recommendation	Did user see this movie
1	Interstellar	True positive
2	Inception	True Positive
3	Avengers	False Positive
4	Harry Potter	True Positive
5	The Imitation Game	False Positive

Now to calculate the average precision (AP), we have the following formula:

$$\text{AP@N} = \frac{1}{m}\sum_{k=1}^{N}(P(k) \text{ if } k^{th} \text{ item was relevant}) = \frac{1}{m}\sum_{k=1}^{N} P(k) \cdot rel(k),$$

In the above formula, rel(k) tells whether that k^{th} item is relevant or not.

Applying the formula, we have:


Relevance of recommendation at each position
rel(k)
[1 1 0 1 0]
*
P(k)
[1/1 2/2 2/3 3/4 3/5]
AP
1/1 + 2/2 + 0 + 3/4 + 0 = 2.75 / m = 10 = 0.275

Average precision of the recommendation list given above

Here, we see that P(k) only contributes to AP if the recommendation at position k is relevant. Also, observe the “placement legalization” by AP by the following scores of three different recommendation lists:

User interaction with recommendation	Precision @ k	AP @ 3
[1 0 0]	[1/1 1/2 1/3]	(1/10)*(1/1) = 0.1
[0 1 0]	[0/1 1/2 1/3]	(1/10)*(1/2) = 0.05
[0 0 1]	[0/1 0/2 1/3]	(1/10)*(1/3) = 0.03

Note that a true positive *(1)*, down the recommendation list, leads to low a mAP compared to the one that is high up in the list. This is important because we want the best recommendations to be at the start of the recommendation set.

Lastly, the “mean” in mAP means that we will calculate the AP with respect to each user’s ratings and take their mean. So, mAP computes the metric for a large set of users to see how the system performs overall on a large set.

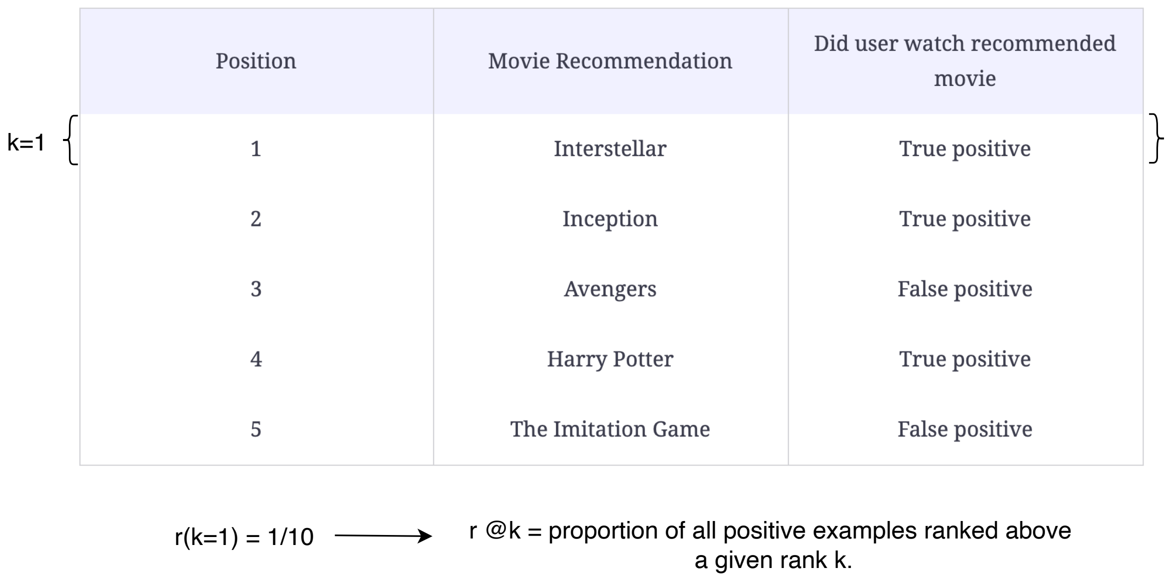
mAR @ N#

Another metric that rewards the previously mentioned points is called Mean Average Recall (mAR @ N). It works similar to mAP @ N. The difference lies in the use of recall instead of precision.

Recall for your recommendation list is the ratio between the number of relevant recommendations in the list and the number of all possible relevant items(shows/movies). It is calculated as:

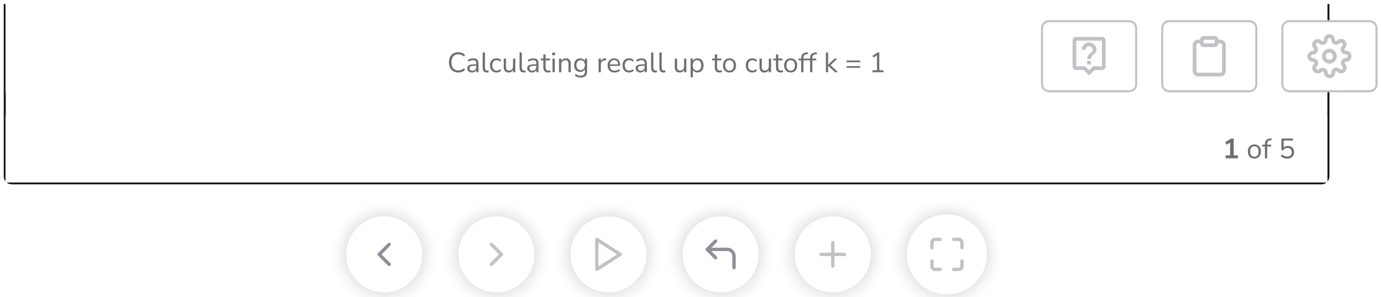
$$r = \frac{number\ of\ relevant\ recommendations}{number\ of\ all\ possible\ relevant\ items}$$

We will use the same recommendation list as used in the mAP @ K example, where N = 5 and m = 10. Let’s calculate the recall of recommendation subsets up to each position, k.



Position	Movie Recommendation	Did user watch recommended movie
1	Interstellar	True positive
2	Inception	True positive
3	Avengers	False positive
4	Harry Potter	True positive
5	The Imitation Game	False positive

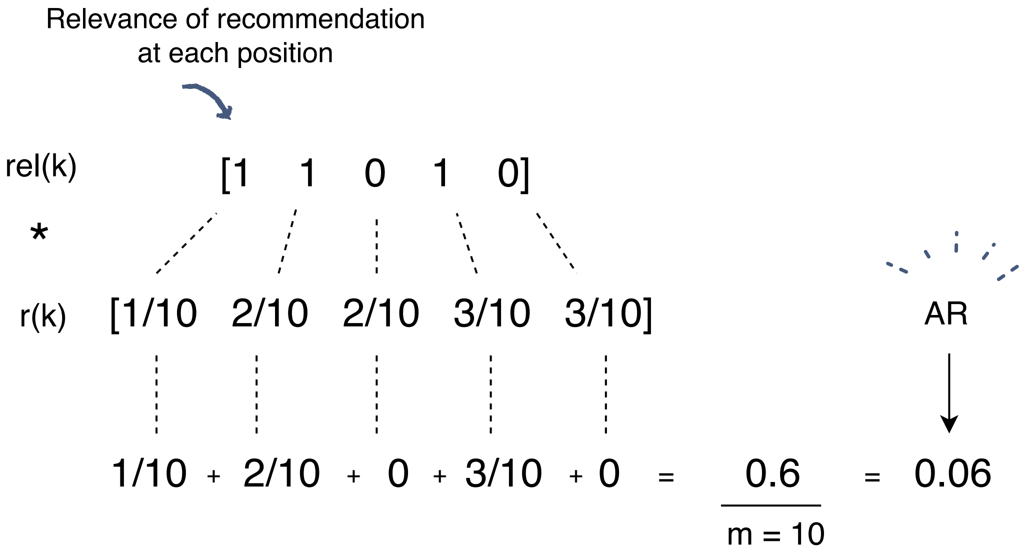
r(k=1) = 1/10 ⟶ r @k = proportion of all positive examples ranked above a given rank k.



Calculating recall up to cutoff k = 1

1 of 5

The average recall (AR) will then be calculated as follows:


Relevance of recommendation
at each position
rel(k)
[1 1 0 1 0]
*
r(k)
[1/10 2/10 2/10 3/10 3/10]
AR
1/10 + 2/10 + 0 + 3/10 + 0 = 0.6 / m = 10 = 0.06

Mean average recall of the recommendation list given above

Lastly, the "mean" in mAR means that we will calculate AR with respect to each user's ratings and then take their mean.

So, mAR at a high-level, measures how many of the top recommendations (based on historical data) we are able to get in the recommendation set.

F1 score#

> 📝 Consider that we have two models, one is giving a better mAP @ N score and the other one was giving a better mAR @ N score. How should you decide which model has better overall performance? If you want to give equal importance to precision and recall, you need to look for a score that conveys the balance between precision and recall.

mAP @ N focuses on how relevant the top recommendations are, whereas mAR @ N shows how well the recommender recalls all the items with positive feedback, especially in its top recommendations. You want to consider both of these metrics for the recommender. Hence, you arrive at the final metric "*F1 score*".

F1 score = 2 * $\frac{mAR*mAP}{mAP+mAR}$

So, the F1 score based on mAP and mAR will be a fairly good offline way to measure the quality of your models. Remember that we selected our recommendation set size to be five, but it can be differ based on the recommendation viewport or the number of recommendations that users on the platform generally engage with.

Offline metric for optimizing ratings#

We established above that we optimize the system for implicit feedback data. However, what if the interviewer says that you have to optimize the recommendation system for getting the ratings (explicit feedback) right. Here, it makes sense to use root mean squared error (RMSE) to minimize the error in rating prediction.

RMSE = $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$

$\hat{y}_i$ is the recommendation system's predicted rating for the movie, and y_i is the ground truth rating actually given by the user. The difference between these two values is the error. The average of this error is taken across N movies.

← Back

Problem Statement

Next →

Architectural Components

☑ Mark as Completed

Report an Issue