**Metrics of Recommender Systems: An Overview**

**Metrics for Recommender Systems differ from traditional metrics. Here we look into 9 such metrics which are used widely in RecSys domain.**



Photo by [Parrish Freeman](https://unsplash.com/@parrish?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Metrics for Recommender Systems differ from traditional metrics like accuracy in the sense that these mostly work cumulatively on an ranked list of predictions instead of scores of individual predictions. It is because, for the scenario of recommender systems, the business needs that we need to optimize almost always get tied up with the set of predictions as a whole and not individually. That being said, let’s look at 9 such metrics which are very important for this domain.

**1. Precision@K**

This is quite similar to normal precision, just that this calculates the precision over the top k items when ordered in the way you want. This way you have the flexibility to vary the K and see how the precision score varies.

https://miro.medium.com/max/498/1*N37hsTEQgIojRh7u2Gg9Ag.png

It has a lot of uses all over the retrieval domain. A very common use would be to measure the performance of a search engine based on its top-10 results to a query.

**2. Recall@K**

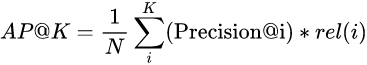
Similar to the usual recall metric and very similar to the above Precision@K metric. It has just one small tweak in the formula compared to the one above.

https://miro.medium.com/max/463/1*V4_mBevljRJgK0iA11SYQw.png

It is useful in cases where there are only a few relevant items and we of course want those at the very front. A good example would be measuring the performance through user clicks on a list of items. Most of the user’s clicks on the items up front would mean the recommendation is working and would have a high recall@K.

**3. MAP@K**

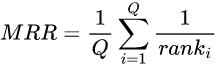
MAP@K or Mean Average Precision @ K is an advanced version of Precision@K. It is good at having a holistic measure of precision instead of basing the metric on just 1 value of K. Let's just look at average precision@K first.



Here, *N* denotes the total relevant items within top-K items. *rel(i)* is the relevance of an item at the *i’th* position. For a simple case, it is either 1 (relevant) or 0 (not relevant). MAP@K is just the mean of all the AP@K over all the queries that you may have. It is a better alternative to both the above metrics.

**4. MRR**

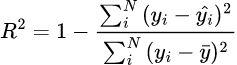
MRR or Mean Reciprocal Rank is the average of the inverse of the rank of the first relevant item for each query. Written into a formula, it goes



Here, *rank\_i* denotes the rank of the first relevant item for the *i’th* query. It is a very simple metric. Perhaps too simple that it can not be adopted widely. It is perfect however for recommendation systems with 1 perfect/correct answer. It may not be a good idea for a scenario with multiple relevant items like e-commerce.

**5. R² Score**

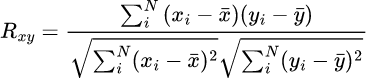
Now we move on to a little different domain where we don’t directly evaluate on the ranked list. Evaluating directly on the ranked list sometimes makes us miss information like how close (in terms of confidence) are the 1st and 2nd items in the list. The coefficient of determination, otherwise known as R², is a very handy and useful metric for regression-type problems.



Here, the numerator denotes the sum of residuals (true -predicted) and the denominator denotes the sum of squares (N \* variance). R² can provide information about how closely a model’s recommendations match the ground truth or some other model’s recommendations.

**6. Pearson Coefficient**

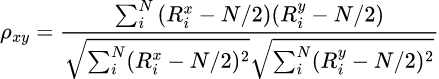
Following the same nature as R² albeit having a different underlying method, Pearson correlation coefficient helps us measure the similarity between two sets of data. At its core, it is the normalized covariance between two variables- the ratio between their covariance and the product of their standard deviations.



Here, x and y are our two sets of data. N is the size of the data. Pearson score has been widely used in the past in old-school collaborative filtering to calculate the similarity between users or items.

**7. Spearman Coefficient**

Imagine you are calculating the similarity between two models’ set of recommendations using Pearson Coefficient with a small twist. Instead of calculating the correlation over the raw item scores, we calculate it over the rank of the items ordered in the set. That’s essentially what the Spearman coefficient is. It is a correlation of the ranks of items between two different sets.



Here, the ***R\_x*** and ***R\_y*** are the ranks in the sets x and y for each *i’*th item. The mean rank is obviously N/2. It can be used to compare between any two ranked lists of items.

**8. NDCG**

NDCG or Normalized Discounted Cumulative Gain is a widely used metric in information retrieval. It is used to calculate a cumulative score of an ordered set of items. The ‘discounted’ part of the metric penalizes items of higher relevance appearing at lower ranks. The normalized DCG takes the DCG till some rank ‘p’ and divides it by the higher possible DCG obtainable by sorting the list (sorting the list by relevance scores will put the most relevant items at the top - which will give the most optimal DCG)

https://miro.medium.com/max/189/1*ae3Ash4XiWAS6UbuEQ3-Ew.png

https://miro.medium.com/max/179/1*x5JZlbpsh55uVczJL6MjSg.png

Here ***rel\_i*** is the relevance of the item at the *i’*th position. ***IDCG\_p*** is the ideal or the highest possible DCG till the rank p

**9. Kendall Tau**

Kendall Tau is another rank correlation metric similar to the spearman coefficient in the sense of what it measures. But its formula is vastly different than anything we have seen so far. The Kendall tau coefficient is defined as

https://miro.medium.com/max/598/1*GjIviYARZ98rzqCY-mOWwQ.png

To explain concordance, let's take a pair (item i, item j) in two lists X and Y. Let X\_i be the rank of the *i’*th item in list X. Then (i,j) is concordant if either of the below is true, else discordant:

1. ***X\_i*** < ***X\_j*** and ***Y\_i*** < ***Y\_j***
2. ***X\_i*** >***X\_j*** and ***Y\_i*** >***Y\_j***

Basically, it expects all pairs to follow the same kind of order in both sets.

We did not cover everything obviously. There are many other metrics, especially in other domains which can be valuable metrics for different use cases in recsys applications. Finalizing the perfect evaluation metric for a specific problem is an important step before even attempting to solve the problem.