Rank-Aware Evaluation Metrics

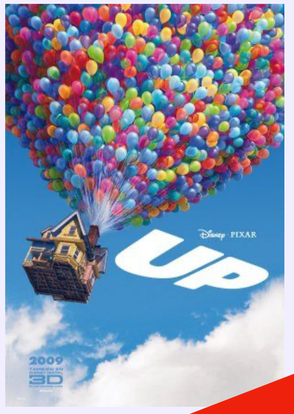
: MRR, MAP, NDCG

- Team [cookie&cream] -

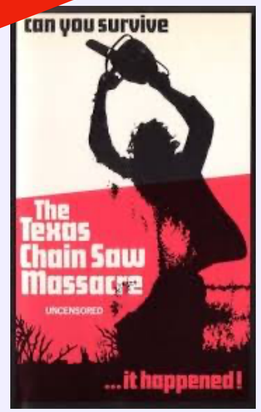
Recommendation Systems

- have a very particular & primary concern

  → preferred items should be ranked top



NEVER!!



Rank-Aware Evaluation Metrics

- Two primary goals
  1) Where does the recommender place the items it suggests?
  2) How good is the recommender at modeling relative preference?

- Three basic metrics
  1) MRR : Mean Reciprocal Rank
  2) MAP : Mean Average Precision
  3) NDCG : Normalized Discounted Cumulative Gain

** families of binary relevance based metrics and utility based metrics

MRR

: Mean Reciprocal Rank

- Rank

  : where the first relative item is located on the recommendation

→ Rank : 3

Relevant item

Non-relevant item

MRR

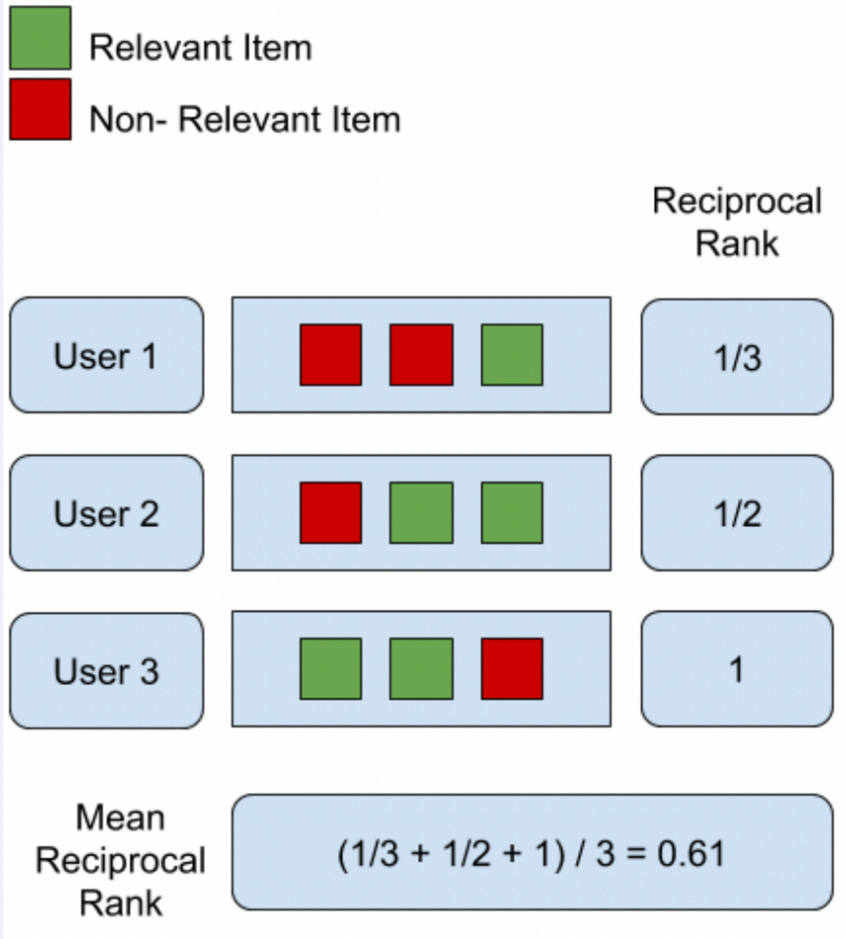
: Calculation Algorithm

For each user u:

- Generate list of recommendations
- Find rank k_u of its first relevant recommendation (the first rec has rank 1)
- Compute reciprocal rank $\frac{1}{k_u}$

Overall algorithm performance is mean recip. rank:

$$\mathrm{MRR}(O, U) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{k_u}$$



MRR
: Pros and Cons

- Pros
  1) simple to compute and easy to interpret
  2) put a high focus on the first relevant element of the list
     → best suited for targeted searches such as ‘best item for me’
  3) good for known-item search such as navigational queries or looking for a fact

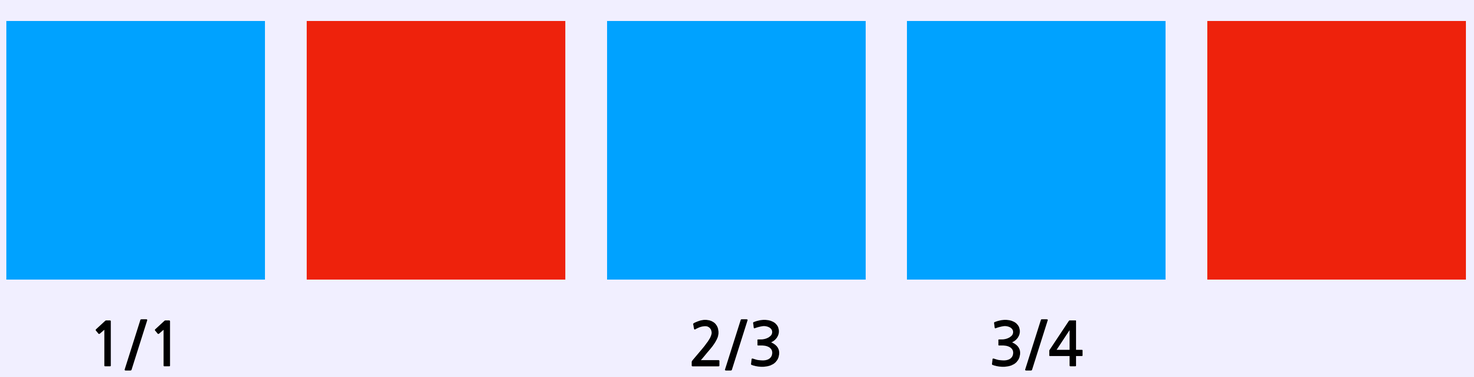
- Cons
  1) does not evaluate the rest of the list of recommended items (only for a single item)
  2) not good for users who want a list of related items to browse

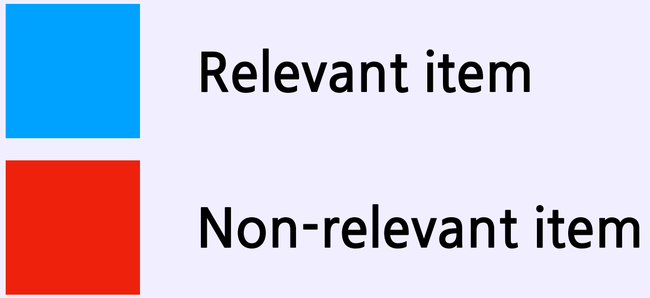
MAP

: Mean Average Precision

- Average precision

  : where the first relative item is located on the recommendation





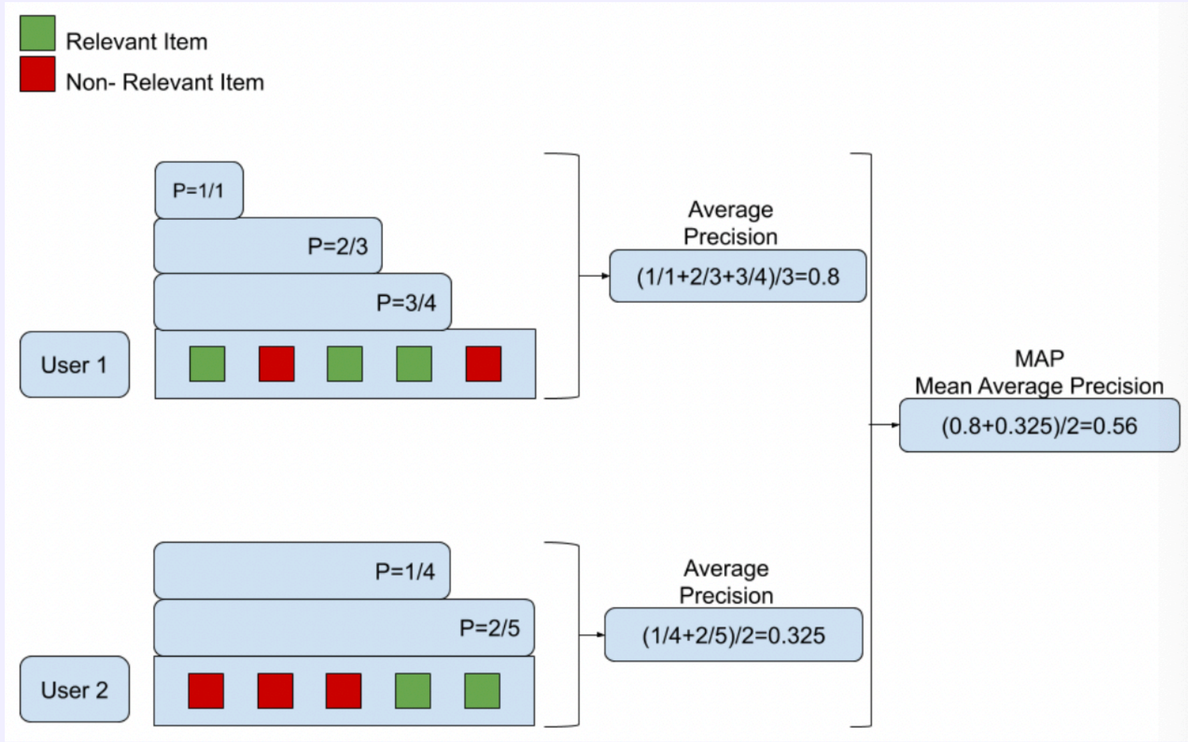
→ Average precision : (1/1 + 2/3 + 3/4) / 3 = 0.8

MAP

: Calculation Algorithm

For each user

- For each relevant item
  - Compute precision of list through that item
- Average sub-list precisions



MAP
: Pros and Cons

- Pros
  1) provide the average precision per list
  2) handle the ranking of lists
     → in contrast to metrics that considering the retrieved items as sets
  3) able to give more weight to errors

- Cons
  1) only for binary ratings
     → not fit for fine-grained ratings (ex-scale from 1 to 5 stars)

NDCG

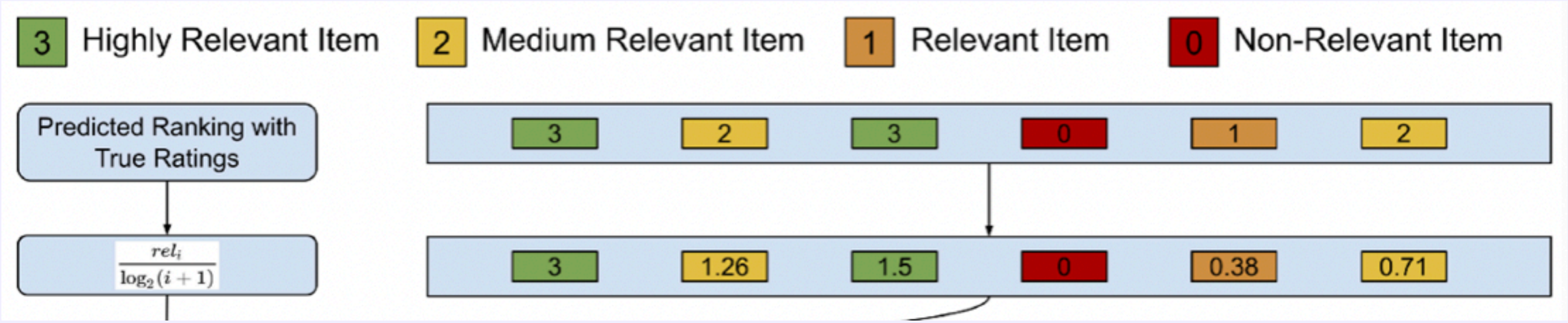
: Normalized Discounted Cumulative Gain

- Cumulative gain

  : sum of relevance score (ex-scale from 1 to 5 stars)

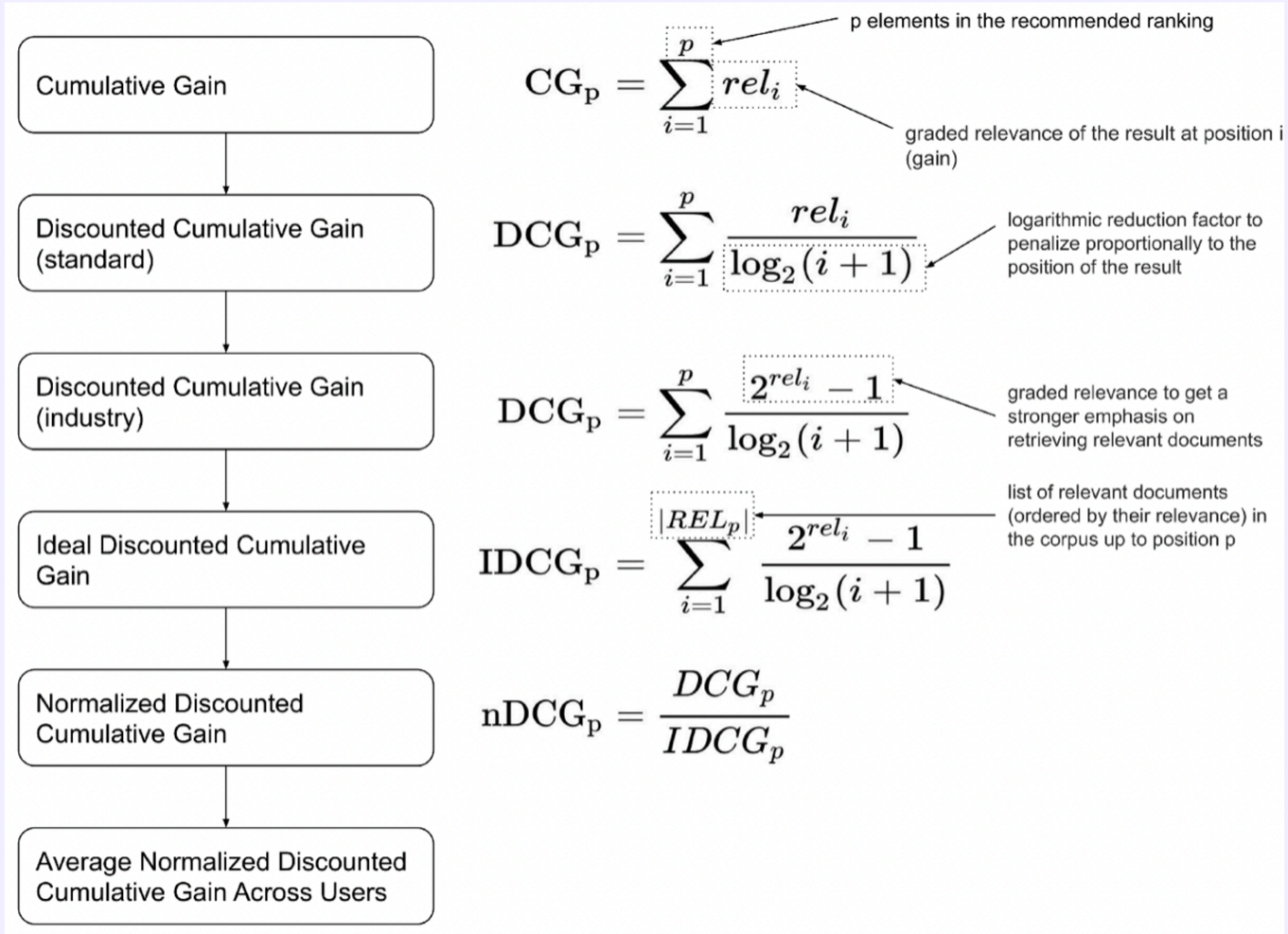
- Discount

  : adjust the relevance score according to the rank



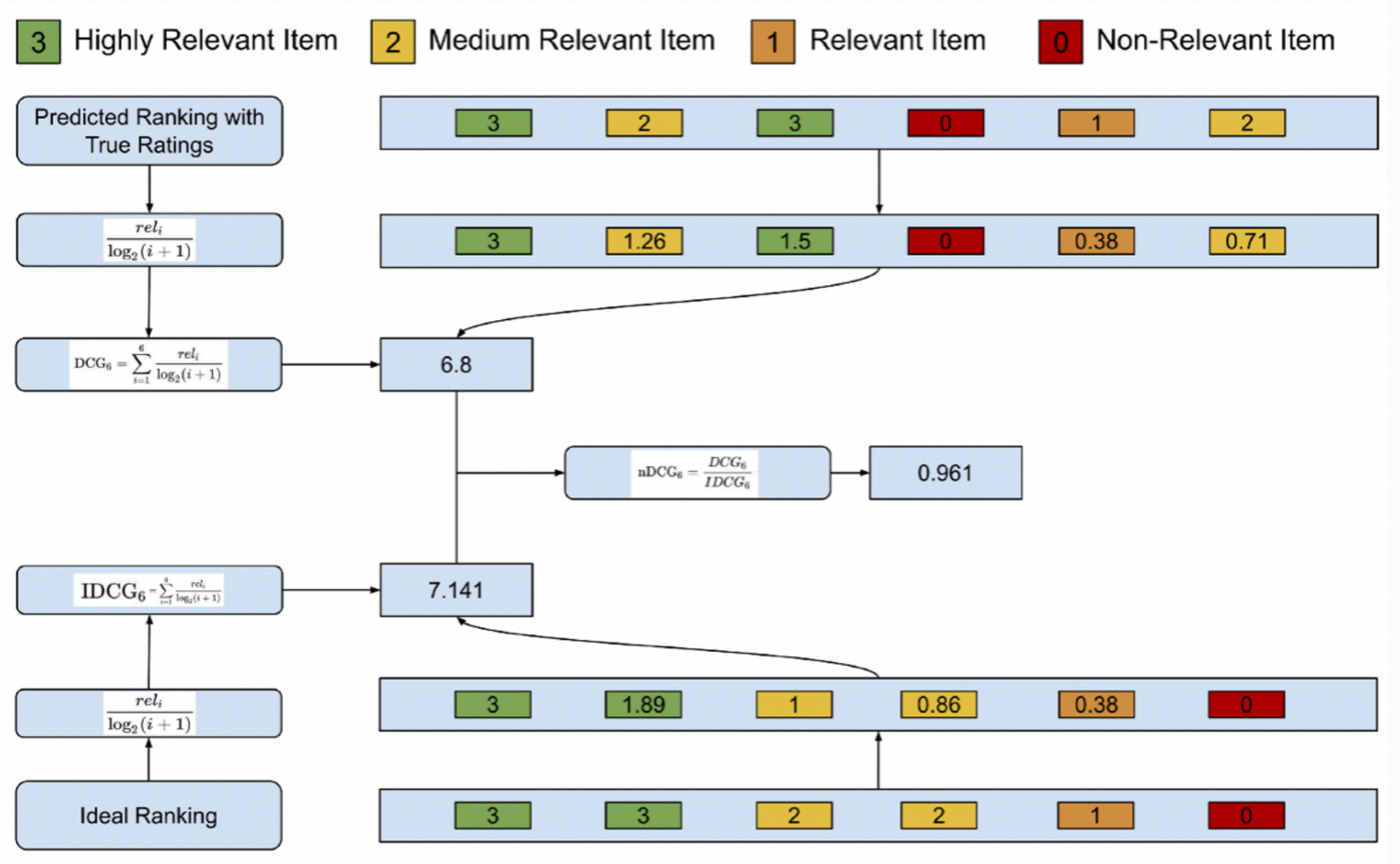
NDCG

: Calculation Algorithm



NDCG

: Example



NDCG

: Pros and Cons

- Pros
  1) take into account the graded relevance values
  2) compared to MAP metric, better at evaluating the position of ranked items

- Cons
  1) the system owner should consider incomplete ratings (missing ratings)
  2) if IDCG equals to 0, how to handle?

Thank you!