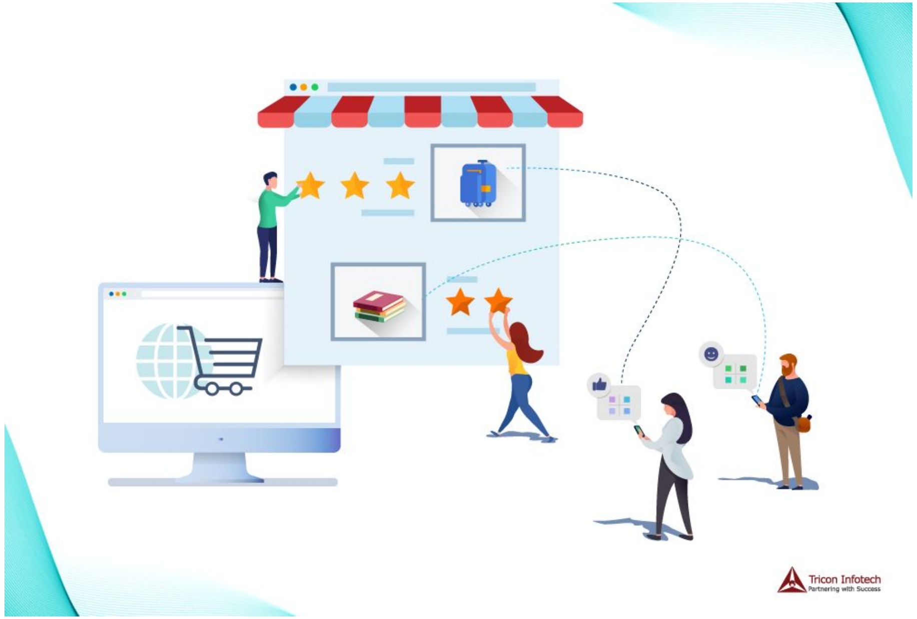
מערכות המלצה 2



יניב הדר
yanivha6@ac.sce.ac.il

measuring accuracy

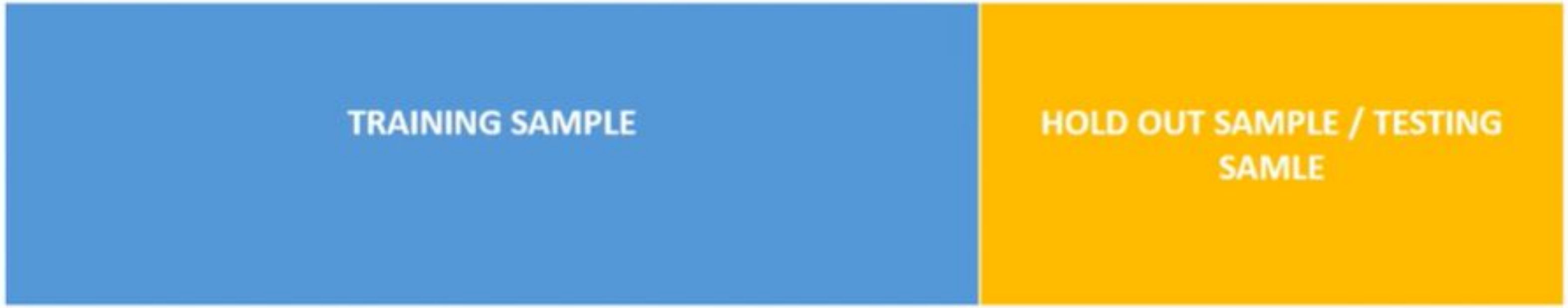
קיים קושי בהערכת המלצה - קיים אספקט סלקטיבי

קשה לתת הערכה לטיב המלצה על מוצר שהמשתמש מעולם לא ראה

Measuring

- Split the available data (so you need to collect data first!), i.e., the user-item ratings into two sets: training and test
- Build a model on the training data
  - For instance, in a nearest neighbor (memory-based) CF simply put the ratings in the training in a separate set
- Compare the predicted rating on each test item (user-item combination) with the actual rating stored in the test set
- You need a metric to compare the predicted and true rating

Hold Out Method



50 - 70 %

50 - 30 %

K-Fold Cross Validation

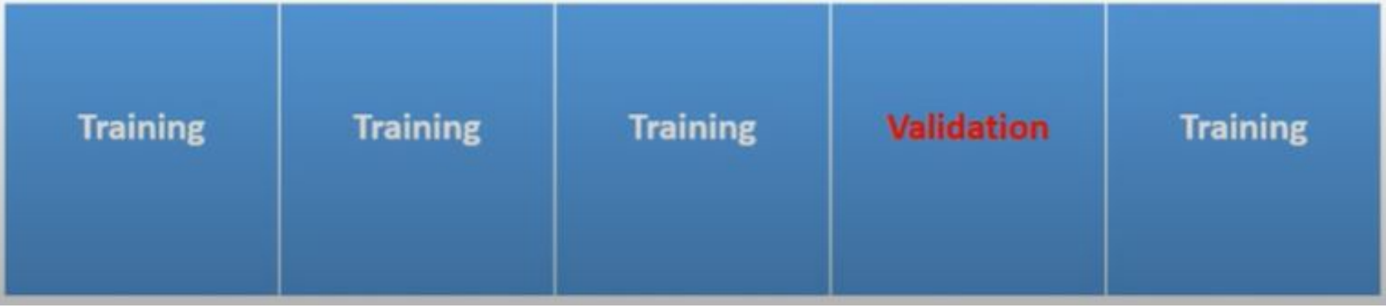
- Split the sample in to K equal size sub samples

(K = 5 - 10)

- Prediction Error = Average(Error)

Advantage

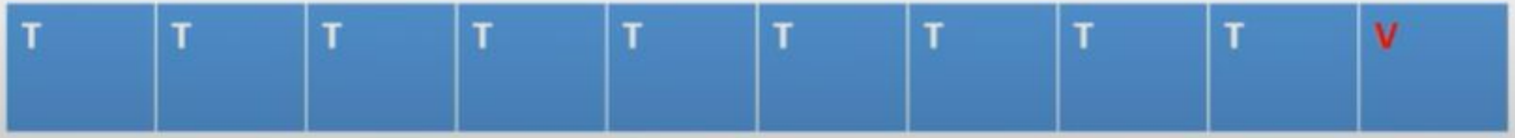
- It matters less how the data gets divided
- Selection bias will not be present



Leave One Out CV

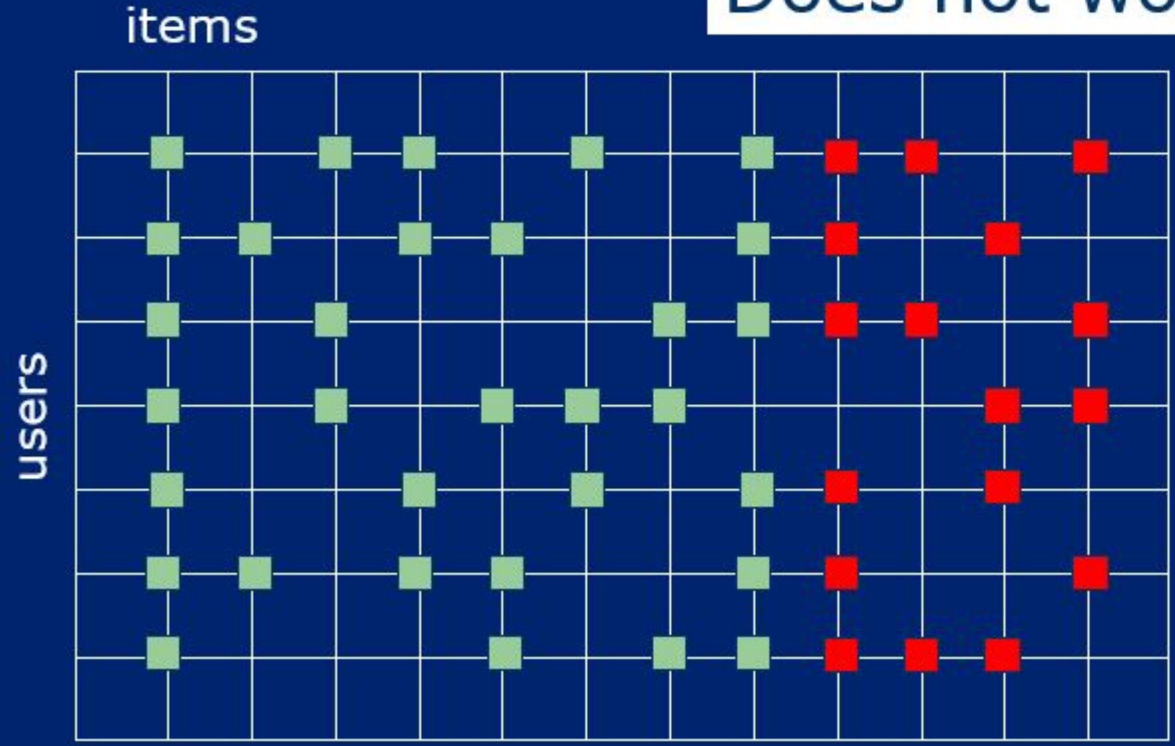
- Specific case of K-fold validation
- K = N

- Good way to validate
- High computation time



Splitting the data

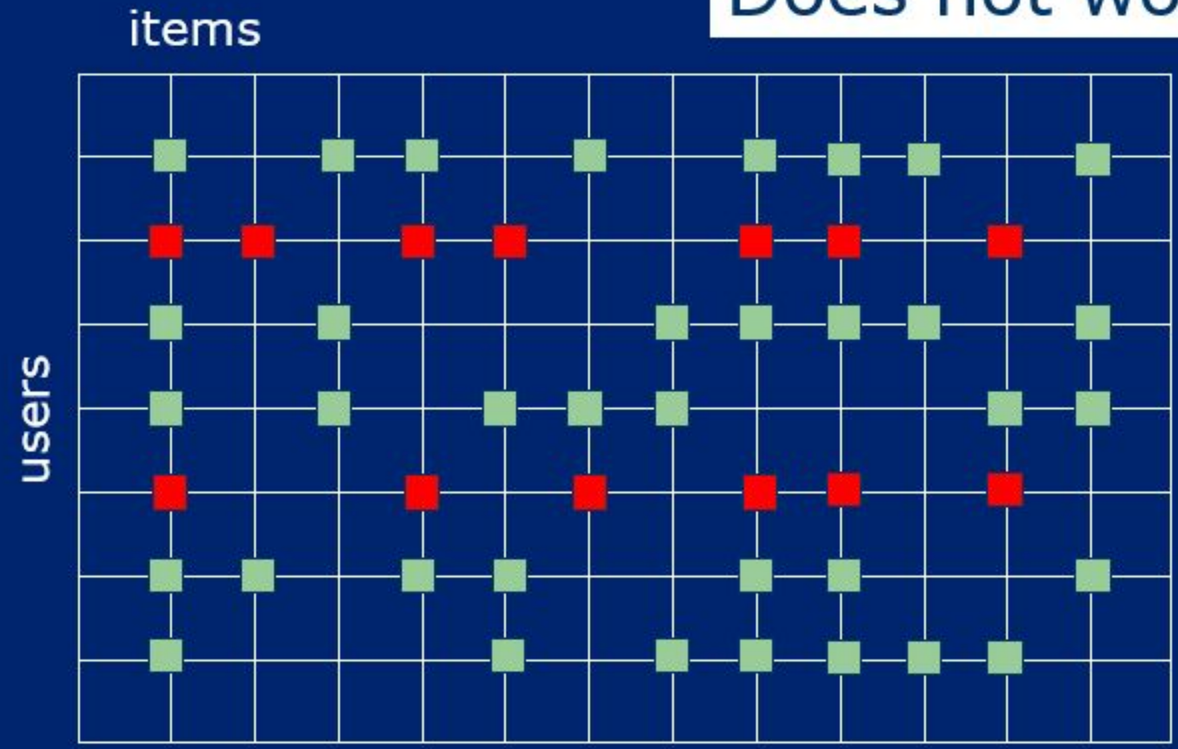
Does not work



🟩 train
🟥 test

Splitting the data

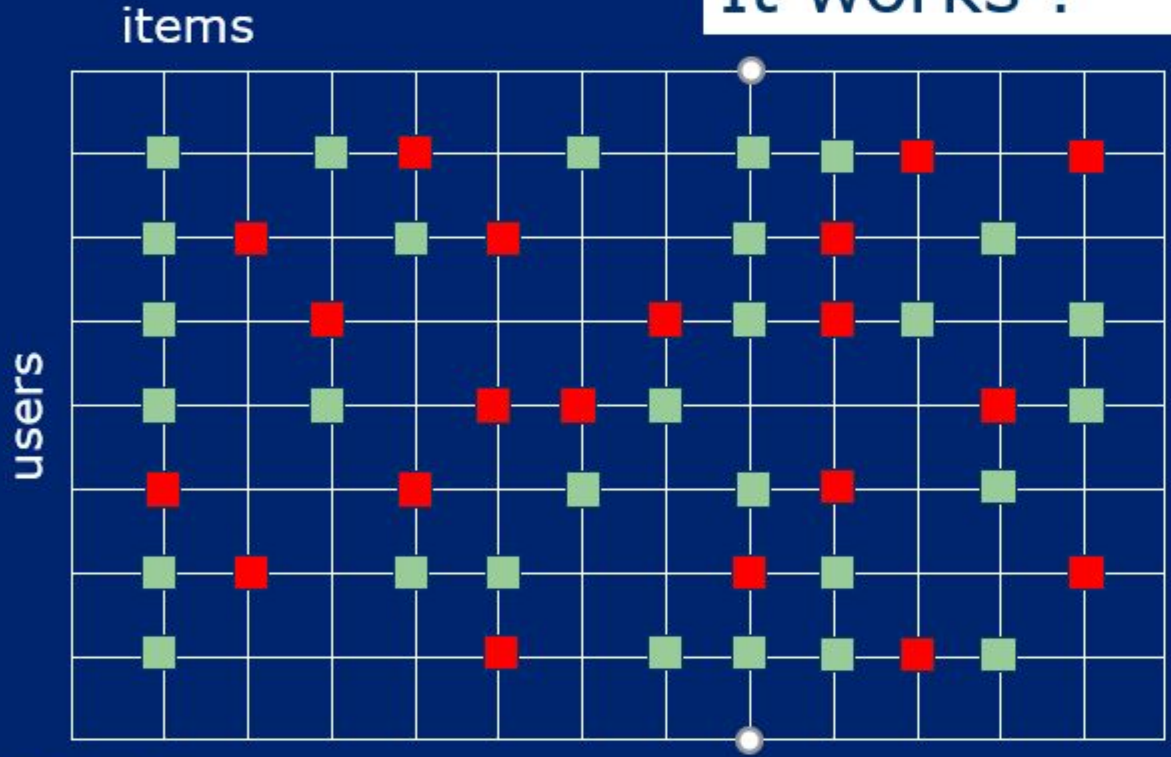
Does not work



🟩 train
🟥 test

Splitting the data

It works !



🟩 train
🟥 test

LOO CV in top-n Recommender

Compute the top-n recommendations for each user in our training data, intentionally remove one of those items from that user training data.

Test our recommenders system's ability to recommend that item that was left out in the top-n results it creates, for that user in the testing phase.

Evaluating Recommender Systems

- The majority focused on system’s accuracy in supporting the “find good items” user’s task
- Assumption: if a user could examine all the available items, he could place them in a ordering of preference
  - Measure how good is the system in predicting the exact rating value (value comparison)
  - Measure how well the system can predict whether the item is relevant or not (relevant vs. not relevant)
  - Measure how close the predicted ranking of items is to the user’s true ranking (ordering comparison)

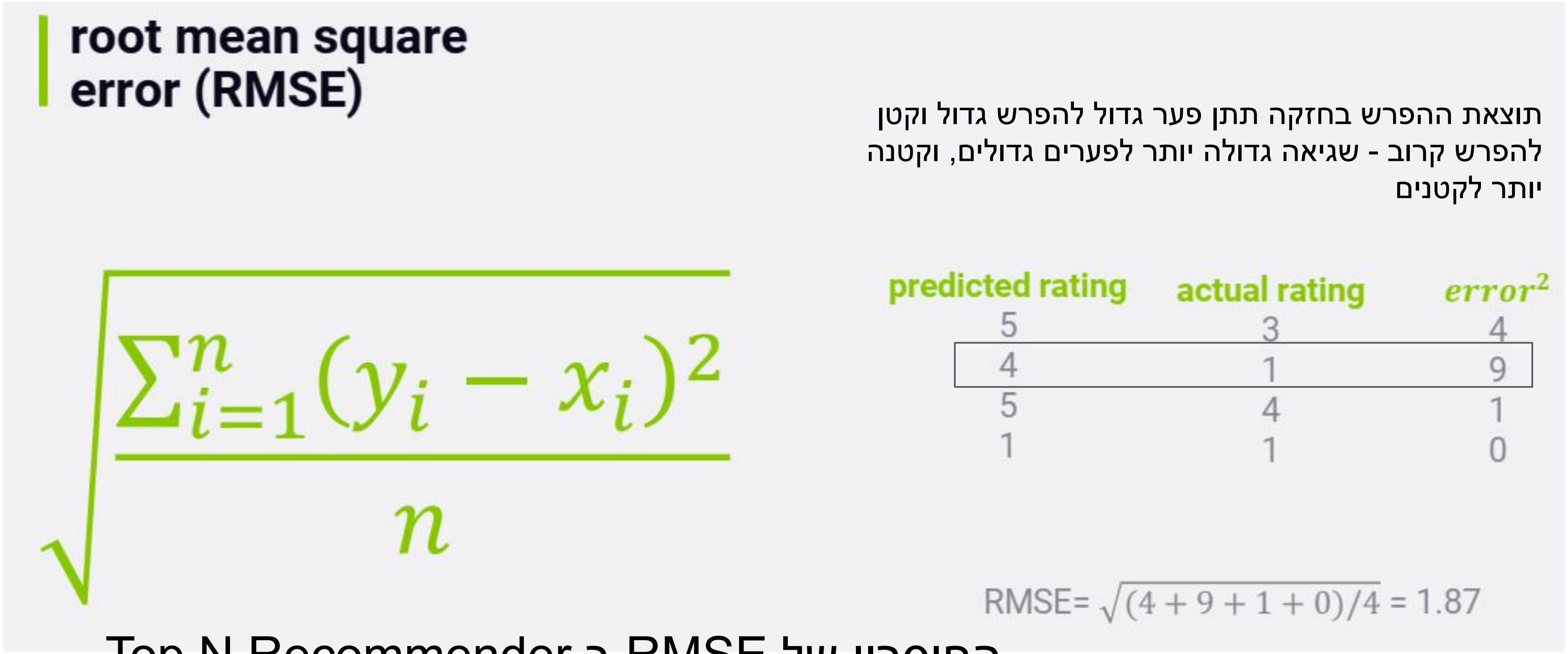
mean absolute error (MAE)

ממוצע הפרש הערכים המוחלטים בין החזוי למצוי

$$\frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

predicted rating	actual rating	error
5	3	2
4	1	3
5	4	1
1	1	0

MAE = (2+3+1+0)/4 = 1.5



root mean square error (RMSE)

תוצאת ההפרש בחזקה תתן פער גדול להפרש גדול וקטן להפרש קרוב - שגיאה גדולה יותר לפערים גדולים, וקטנה יותר לקטנים

$$\sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n}}$$

predicted rating	actual rating	$error^2$
5	3	4
4	1	9
5	4	1
1	1	0

RMSE= $\sqrt{(4+9+1+0)/4}$ = 1.87

החיסרון של RMSE ב Top N Recommender

נטפליקס הציעה פרס להורדת ערכי RMSE וכך מיקדה את התעשיה והמחקר במדד מסוים

הפער בין דירוג רע - לחיזוי דירוג רע ישפיע על RMSE - האם זהו פער שרלוונטי להמלצות ?

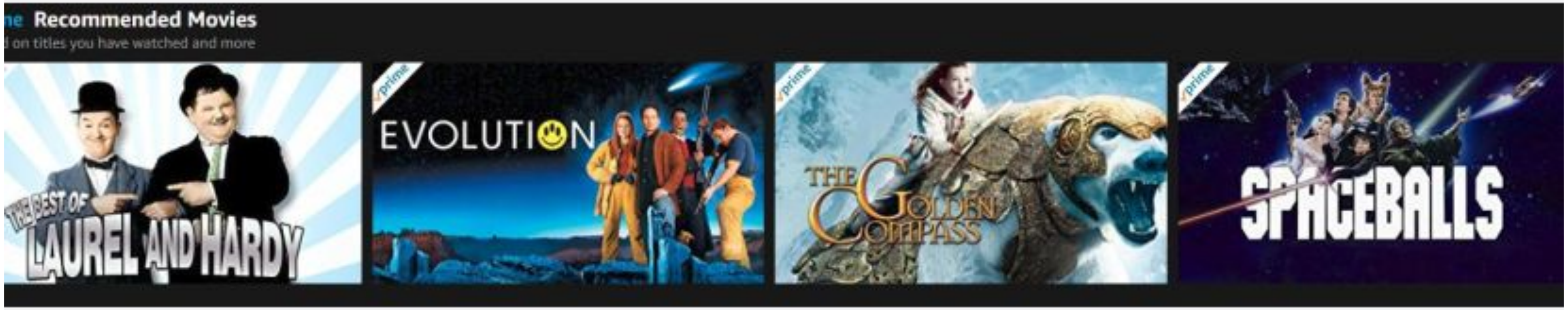
Accuracy: Comparing Values

- Measure how close the recommender system's predicted ratings are to the true user ratings (for all the ratings in the test set).
- Predictive accuracy (rating): Mean Absolute Error (MAE), pi is the predicted rating and ri is the true one:

$$MAE = \frac{\sum_{i=1}^{N} | p_i - r_i |}{N}$$

- It may be less appropriate for tasks such as Find Good Items – because people look only to top rated items
- Variation 1: Mean Squared Error (take the square of the differences), Root Mean Squared Error (and then take the square root). These emphasize large errors.
- Variation 2: Normalized MAE – MAE divided by the range of possible ratings – allowing comparing results on different data sets, having different rating scales.

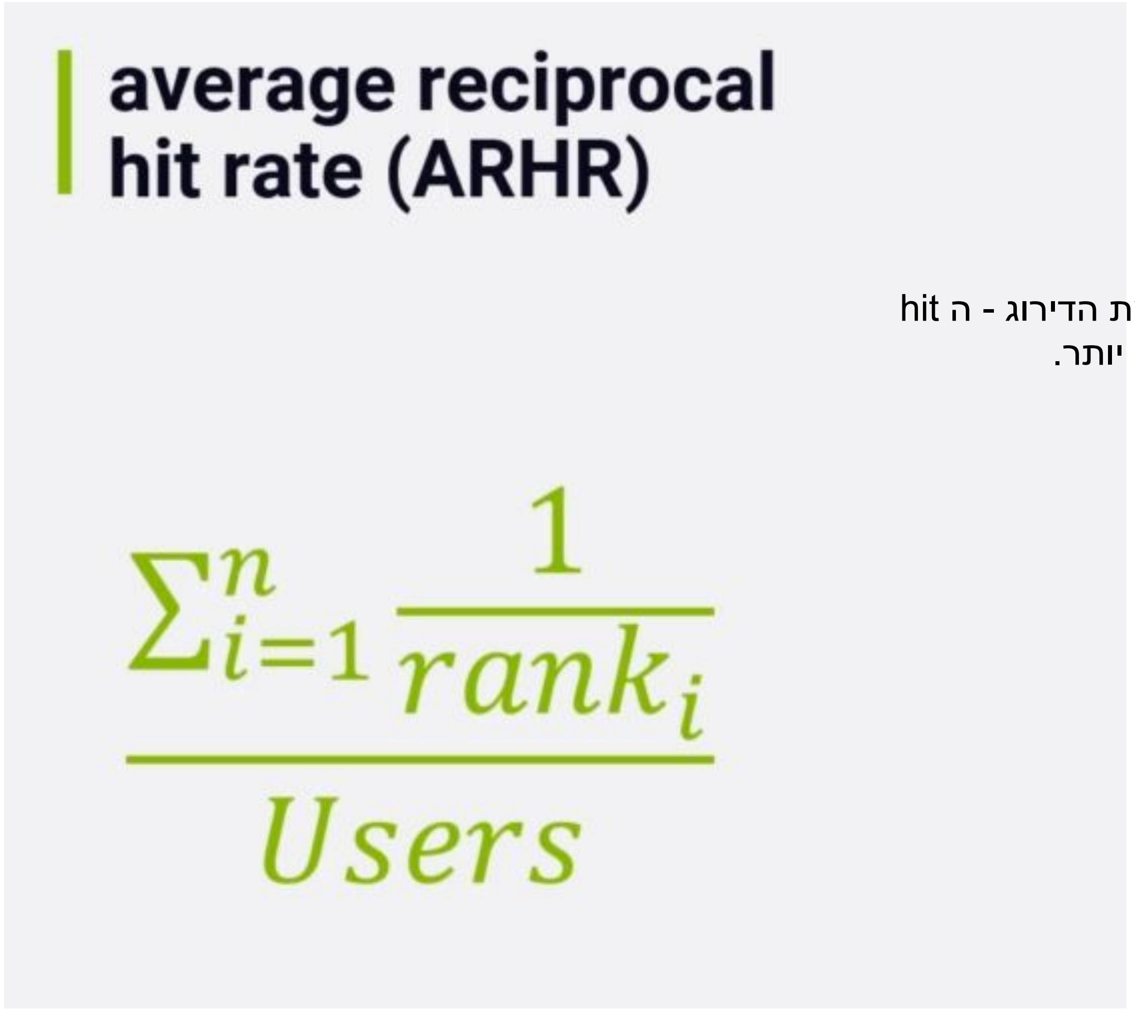
evaluating top-n recommenders


Recommended Movies
on titles you have watched and more
THE BEST OF LAUREL AND HARDY
EVOLUTION
THE GOLDEN COMPASS
SPACEBALLS

hit rate

$$\frac{hits}{users}$$

נגדיר את קבוצת ההמלצות של המשתמש (top-n)
ובמידה ואחת ההמלצות אכן דורגה נקבל hit

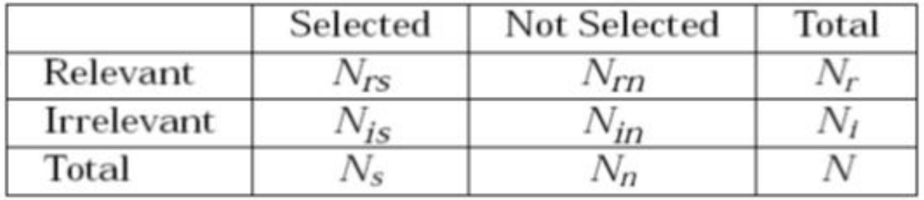


average reciprocal hit rate (ARHR)

$$\frac{\sum_{i=1}^{n}\frac{1}{rank_i}}{Users}$$

במדד זה ניתן לראות התייחסות גם לרמת הדירוג - ה hit חזק יותר אם ההמלצה היא מדירוג גבוה יותר.

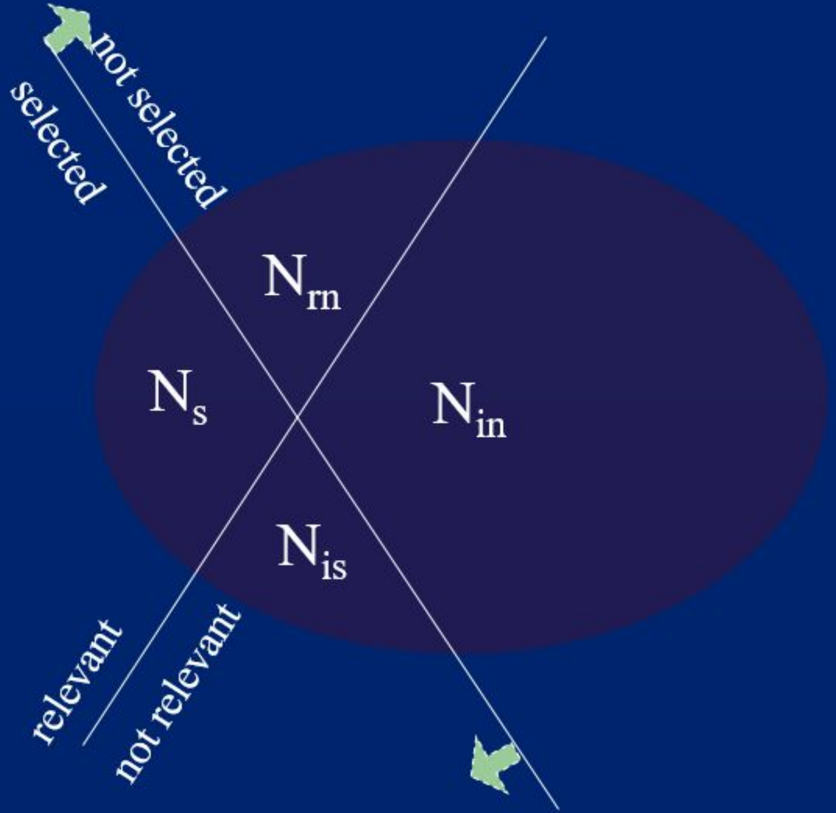
Relevant Recommendations: Precision and Recall



	Selected	Not Selected	Total
Relevant	N_{rs}	N_{rn}	N_r
Irrelevant	N_{is}	N_{in}	N_I
Total	N_s	N_n	N

- The rating scale must be binary – or one must transform it into a binary scale (e.g. items rated above 4 vs. those rated below)
- Precision is the ratio of relevant items selected by the recommender to the number of items selected (Nrs/Ns)
- Recall is the ratio of relevant items selected to the number of relevant (Nrs/Nr)
- Precision and recall are the most popular metrics for evaluating information retrieval systems.

Precision and Recall



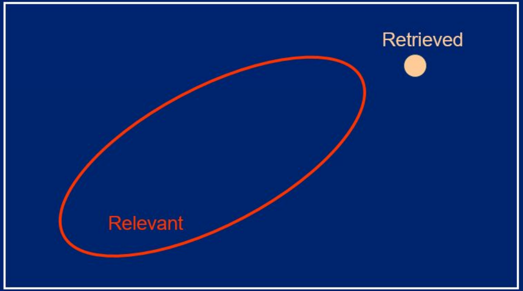
Precision = N_s / (N_s +N_{is})

Recall = N_s / (N_s + N_{rn})

To improve both P and R you need to bring the lines closer together - i.e. better determination of relevance.

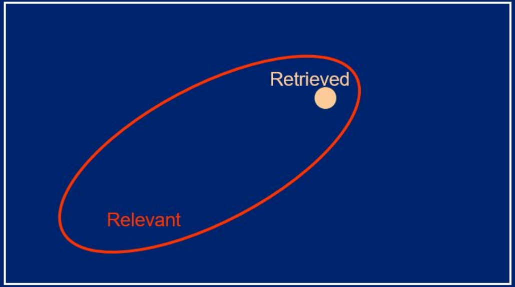
Retrieved vs. Relevant Documents

Very low precision, very low recall (0 in fact)



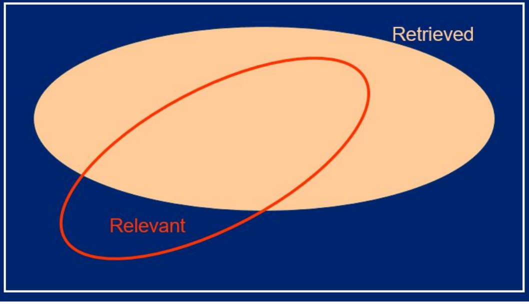
Retrieved vs. Relevant Documents

Very high precision, very low recall



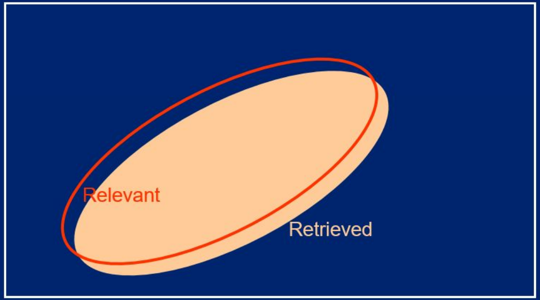
Retrieved vs. Relevant Documents

High recall, low precision



Retrieved vs. Relevant Documents

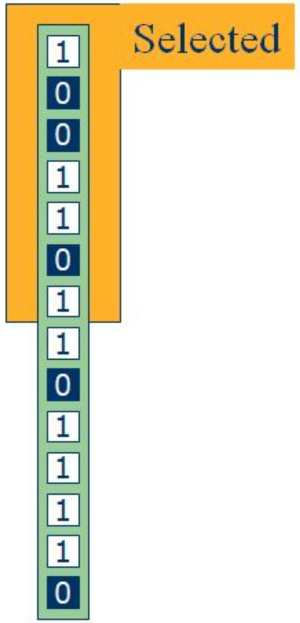
High precision, high recall (at last!)



Example – Complete Knowledge

We assume to know the relevance of all the items in the catalogue for a given user

The orange portion is that recommended by the system



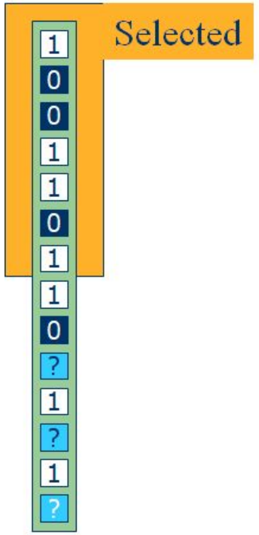
Precision=4/7=0.57
1's in the orange portion
Recall=4/9=0.44
The 1's recommended from the whole

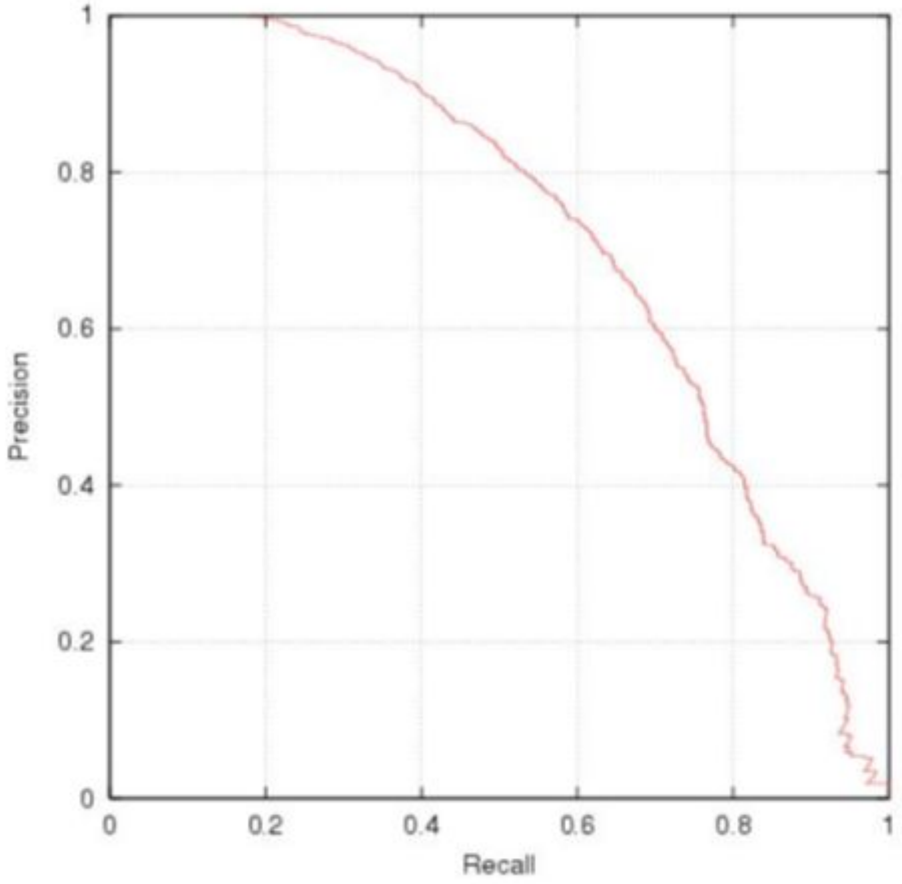
Example – Incomplete Knowledge

We do not know the relevance of all the items in the catalogue for a given user

The orange portion is that recommended by the system

Precision=4/7=0.57 – As before
Recall=4/?
4/10 ≤ Recall ≤ 4/7
4/10 if all unknown are relevant
4/7 if all unknown are irrelevant





Precision vs. Recall

A typical precision and recall curve

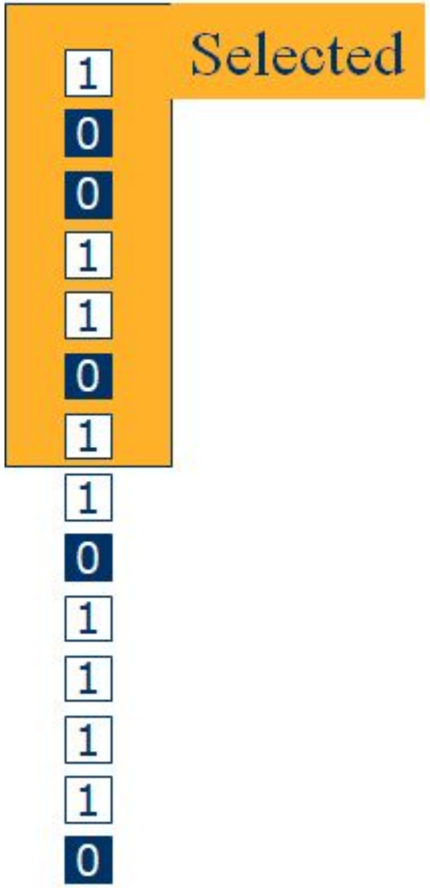
F1

Combinations of Recall and Precision such as F1

Typically systems with high recall have low precision and vice versa

$$F_1 = \frac{2PR}{P+R}$$

P=4/7=0.57
R=4/9=0.44
F1 = 0.5



Problems with Precision and Recall

- To compute them we **must know** what items are **relevant** and what are **not relevant**
- Difficult to know what is relevant for a user in a recommender system that manages **thousands**/**millions** of products
- May be easier for some tasks where, given the user or the context, the number of recommendable products is small – only a small portion could fit
- **Recall** is more **difficult to estimate** (knowledge of all the relevant products)
- **Precision** is a bit **easier** – you must know what part of the selected products are relevant (you can ask to the user after the recommendation – but has not been done in this way – not many evaluations did involve real users).

Rank Accuracy Metrics

Rank is the position in the sorted list (and rating?)

Spearman's rank correlation (ui and vi are the ranks – position in the sorted list - of item i in the user and system order) – it is the same as Pearson but computed on the ranks

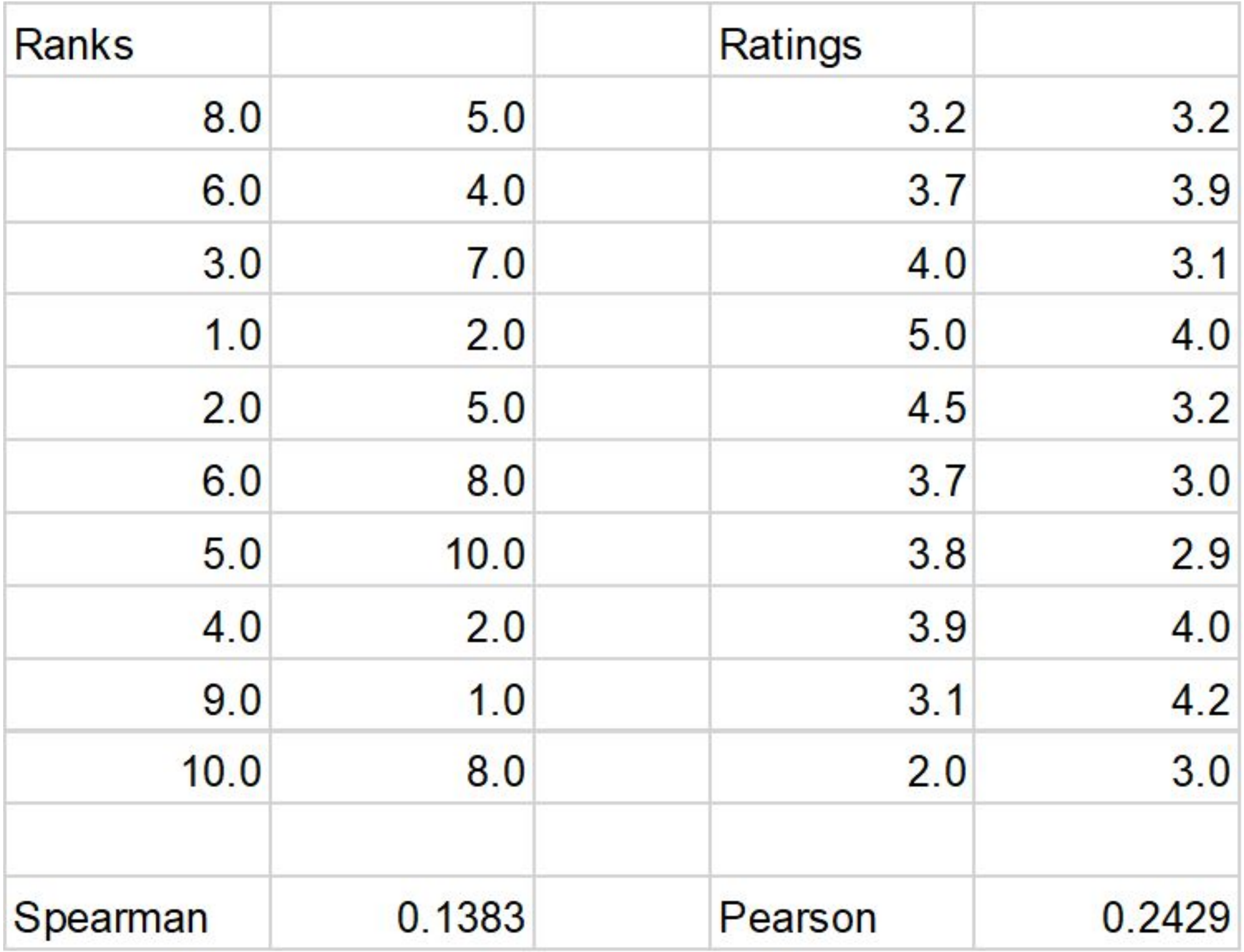
$$\rho = \frac{\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})}{n * stdev(u) stdev(v)}$$

stdev (x): $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2},$

Kendall's Tau (C is the number of concordant pairs – pairs in the same order in both ranked lists – D is the number of discordant pairs, TR tied pairs (same ranking) in the user order, TP tied pair in the predicted order

$$Tau = \frac{C - D}{\sqrt{(C + D + TR)(C + D + TP)}}$$

Example



Ranks			Ratings	
8.0	5.0		3.2	3.2
6.0	4.0		3.7	3.9
3.0	7.0		4.0	3.1
1.0	2.0		5.0	4.0
2.0	5.0		4.5	3.2
6.0	8.0		3.7	3.0
5.0	10.0		3.8	2.9
4.0	2.0		3.9	4.0
9.0	1.0		3.1	4.2
10.0	8.0		2.0	3.0
Spearman	0.1383		Pearson	0.2429

Example

- If the two rankings are the same (perfect agreement) - the coefficient has value 1.
- If one ranking is the reverse of the other (perfect disagreement) - the coefficient has value −1.
- For all other arrangements the value lies between −1 and 1, and increasing values imply increasing agreement between the rankings.
- If the rankings are completely independent, the coefficient has value 0 on average.

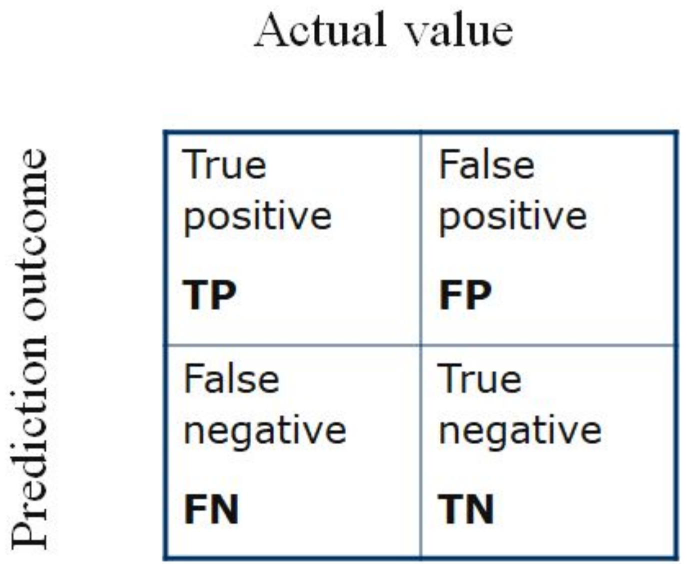
Ranks	
8,0	5,0
6,0	4,0
3,0	7,0
1,0	2,0

$$Tau = \frac{C - D}{\sqrt{(C + D + TR)(C + D + TP)}}$$

$$Tau = \frac{4 - 2}{\sqrt{(4 + 2 + 0) * (4 + 2 + 0)}}$$

ROC Curves

- “Relative Operating Characteristic” or “Receiver Operating Characteristic”
- Definition: a graphical depiction of the relationship between the true positive rate (TPR) and false positive rate (FPR = 1 - TPR) as a function of the cutoff at all points in the recommendation list
- Characteristics
  - Evaluates a binary classification
  - Not a single number metric
  - Covers performance of system at all points in the recommendation list



	Actual value	
Prediction outcome	True positive **TP**	False positive **FP**
	False negative **FN**	True negative **TN**

What is AUC - ROC Curve?

- AUC - ROC curve is a performance measurement for classification problem at various thresholds settings.
- https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

The ROC model assumes that the information system will assign a predicted level of relevance to every potential item.

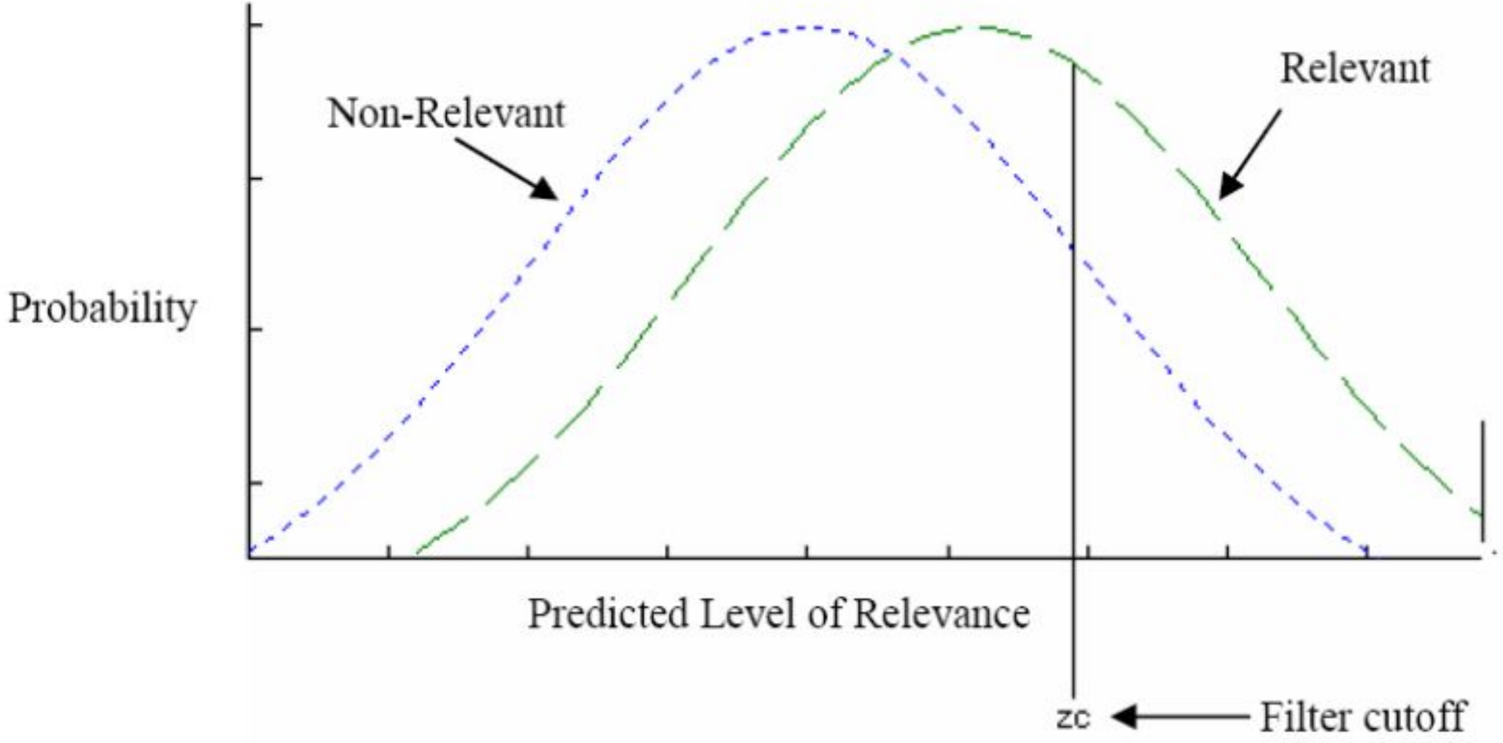
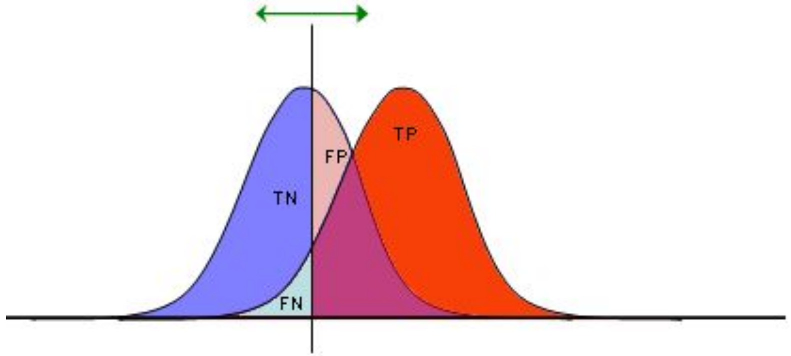
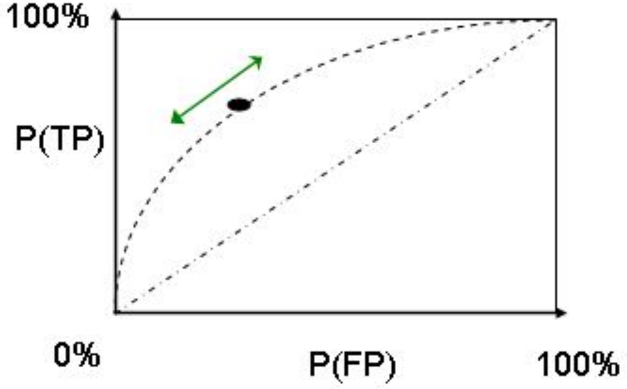


Figure 1. A possible representation of the density functions for relevant and irrelevant items.



TP	FP
FN	TN
1	1



Algorithm for building ROC

A common algorithm for creating an ROC curve goes as follows:

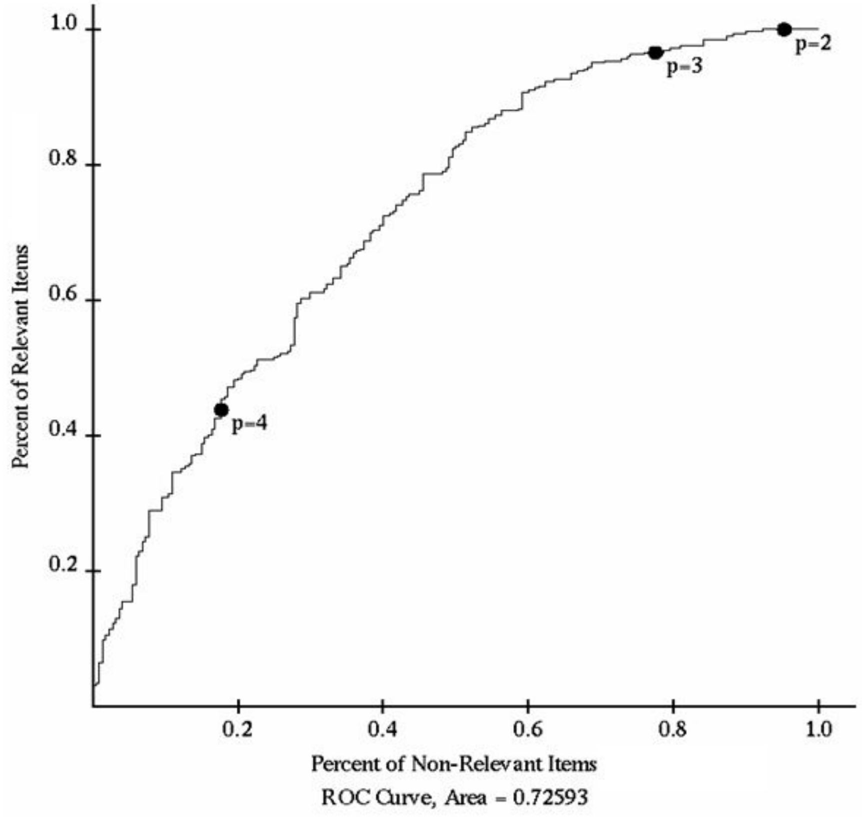
1. Determine how you will identify if an item is relevant or non-relevant.
2. Generate a predicted ranking of items.
3. For each predicted item, in decreasing order of predicted relevance (starting the graph at the origin):
    a. If the predicted item is indeed relevant draw the curve one step vertically.
    b. If the predicted item is not relevant, draw the curve one step horizontally to the right.
    c. If the predicted item has not been rated (i.e. relevance is not known), then the item is simply discarded and does not affect the curve negatively or positively.

An example of an ROC curve

The p-values shown on the curve represent different prediction cutoffs.

For example if we chose to select all items with predictions of 4 or higher, then we experience approximately 45% of all relevant

items and 20% of all non-relevant items.

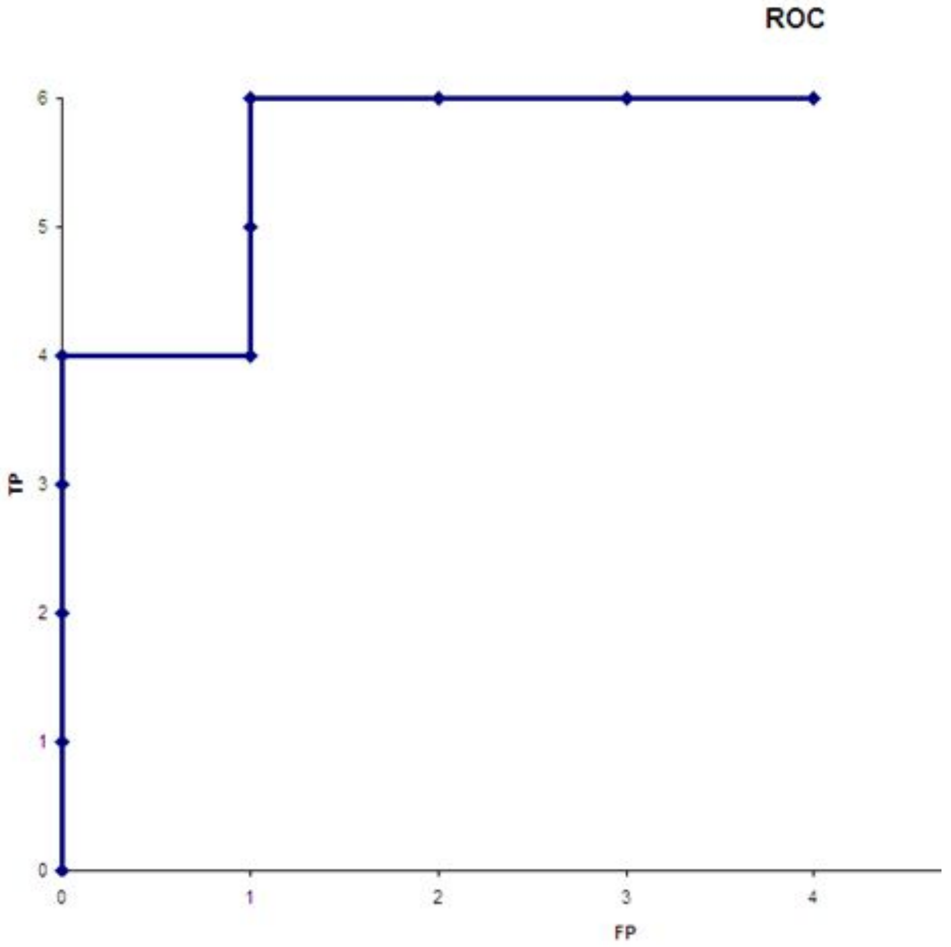


Example

Ranks	
8,0	R
6,0	R
9,0	R
10,0	R
2,0	N
6,0	R
5,0	R
4,0	N
3,0	N
1,0	N

User's ranks

List is sorted by system's (predicted) ranking



Half-life Utility Metric

ra,j represents the rating of user a on item j of the ranked list,

d is the default rating (is generally a neutral or slightly negative rating),

α is the half-life - the rank of the item on the list such that there is a 50% chance that the user will view that item.

Breese et al. used a half-life of 5 for his experiments

$$R_a = \sum_j \frac{\max\left(r_{a,j} - d, 0\right)}{2^{(j-1)/(\alpha-1)}}$$

Half-life Utility Metric

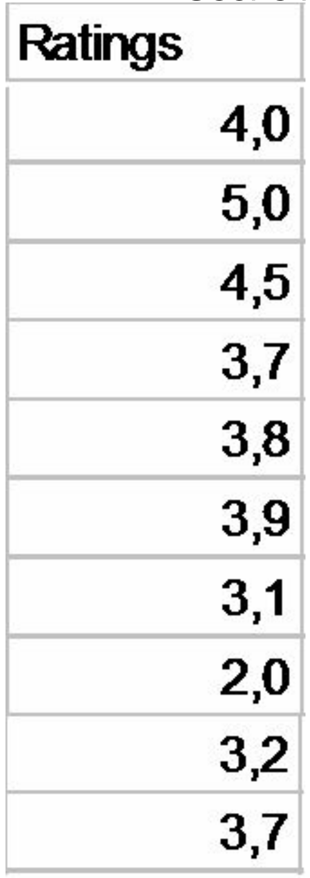
$$R_a = \sum_j \frac{\max(r_{a,j} - d, 0)}{2^{(j-1)/(\alpha-1)}}$$

Characteristics

- Evaluates the utility of a ranked list to the user
  - Utility is defined as the difference between the user's rating for an item and the "default rating" for an item.
- Explicitly incorporates idea of decreasing user utility
  - To obtain high values of the metric, the first predicted rankings must consist of items rated highly by the user
- Tuning parameters reduce comparability (d, α)
- Weak orderings can result in different utilities for the same system ranking
- All items rated less than the max contribute equally
- Only metric to really consider non-uniform utility

Example

User’s ratings



Ratings
4,0
5,0
4,5
3,7
3,8
3,9
3,1
2,0
3,2
3,7

List is sorted by system’s (predicted) ranking

Set d = 2.5, α = 5

Ra = 1.5/1 + 2.5/1.19 + ... = 7.608

NDPM-normalized distance-based performance measure

- Used to compare two different ordered ratings.
  - Measures the difference between the ranking imposed by a user's ratings and the ranking predicted by the system.
- Can be used to compare two different weakly-ordered rankings.
  - Similar in form to the Spearman and Kendall’s Tau rank correlations, but provides a more accurate interpretation of the effect of tied user ranks.
- Suffers from the same interchange weakness as the rank correlation metrics (interchanges at the top of the ranking have the same weight as interchanges at the bottom of the ranking).

$$NDPM = \frac{2C^{-} + C^{u}}{2C^{i}}$$

NDPM-normalized distance-based performance measure

- C- is the number of contradictory preference relations between the system ranking and the user rating.
  - when the system says that item 1 will be preferred to item 2, and the user ranking says the opposite.
- Cu - is the number of half-contradictory preference relations.
  - where the user rates item 1 higher than item 2, but the system ranking has item 1 and item 2 at equal preference levels.
- Ci - is the total number of “preferred” relationships in the user’s ranking.
  - i.e. pairs of items rated by the user for which one is rated higher than the other
-

$$NDPM = \frac{2C^{-} + C^{u}}{2C^{i}}$$

Example

Ranks	
8,0	5,0
6,0	4,0
3,0	7,0
1,0	2,0

$$NDPM = \frac{2C^{-} + C^{u}}{2C^{i}}$$

$$NDPM = \frac{2 * 2 + 0}{2 * 6}$$