Master on Foundations of Data Science


# Recommender Systems

**Evaluation**

Santi Seguí | **2023-2024**

How can we evaluate if
our recommender is **good**?


How can we evaluate if
our recommender is **good**?

# Hypothesis

- What does **good** mean?
    - Recommendation **accuracy**?
    - Recommendation **quality**?
    - System **usability**?
    - System **satisfaction**?

# Hypothesis

- What does **good** mean?

1. Recommendation **accuracy**?
   - Ability to estimate a ranked list of items accurately. A good recommender system should be capable of predicting user preferences with high precision.

2. Recommendation **quality**?
   - Beyond accuracy, recommendation quality encompasses various factors such as relevance, novelty, diversity, and serendipity. A truly effective system not only suggests items accurately but also introduces users to new and unexpected choices, enriching their experience.

3. System **usability**?
   - The manner in which recommendations are presented significantly impacts user engagement and satisfaction.

4. System **satisfaction**?
   - System satisfaction is a measure of user happiness and fulfillment with the recommendations provided. It involves factors such as trust, transparency, and the system's responsiveness to user feedback

You do need to **understand** who your **customers** are because **different people want different recommendations**

not everyone wants the most accurate system

# When implementing a recommender system, one of the first things you should ask yourself is:

**Why did you implement a recommender?**

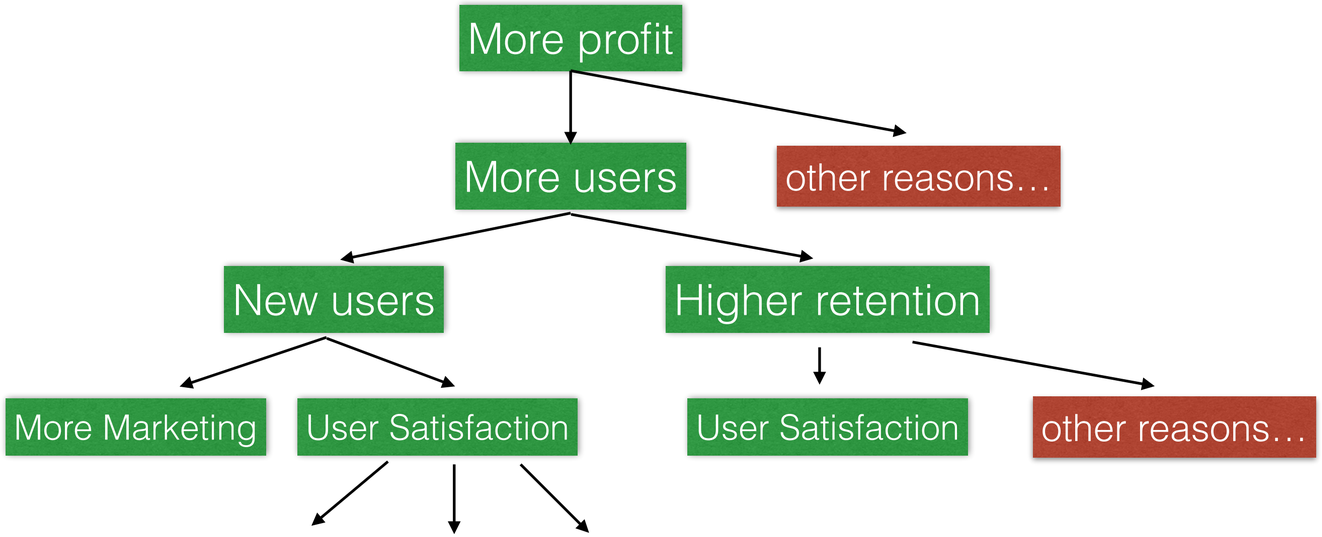
What did you want to gain?
Do you want to earn more?
Have more visitors?
Try out new technology?

No matter what you answer, it might not directly translate into a way to calculate whether or not you're improving.
You often hear about algorithms that are better or slightly improved compared to the current cutting-edge algorithms, but improving what and how?


# Why a recommender system?


# Why a recommender system?

Let's say that what to increase their **profit**



“Netflix has deduced (probably by looking at the data) that user retention is correlated with the user watching more”

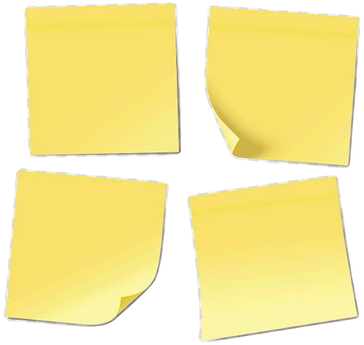
# Evaluating Recommender Systems

- This is one of the **most critical** steps when building a recommender system.
- A single criterion cannot capture many goals of the designer.


**We need to define procedures and measures!**

# How an ideal recommendation should be?

- Recommender systems use past behaviours of users to suggest items.
  - Obvious recommendations may become boring and create what is called the "filter bubble effect".
- **What do you expect from a recommender system**?
- **Do you like receiving recommendations with novel and unexpected items**?



NETFLIX Spotify® amazon LinkedIn

# Your perception about Recommender Systems

- What about our personal experience, when surfing online and looking for something to buy, a movie to watch, news to read or a new song to enjoy.
  - Do you trust recommender systems and do you take recommended items into account? **Why?**
  - What kind of personal data are you willing to share for this purpose?

Make accurate predictions is needed but recommender system also need to account for factors like **diversity**, **context**, **evidence**, **freshness**, and **novelty**


CAUTION
RecSys
can change
user behaviors

# Quality indicators

# Quality Indicators

- Relevance 
- Coverage 
- Novelty 

- Diversity 

- Confidence/Trust 
- Serendipity 
- Scalability 
- Robustness/Stability 

# Relevance

Relevance is measures the recommendation
***accuracy***
but ***accuracy*** often provide
an incomplete picture of the conversion rate

Higher **accuracy** does **not** always
mean higher **satisfaction**

Being accurate is not enough: how accuracy metrics have hurt recommender systems
SM McNee, J Riedl, JA Konstan
CHI'06 extended abstracts on Human factors in computing systems, 1097-1101
680 2006

# Coverage

- One of the main reasons for implementing a recommender system is to enable users to navigate the **full catalog**, which is known as **content coverage**.
- Even, when the recommender is highly accurate, it may not be able to recommend a set of items, or it may not be able to recommend some items to a set to users. This measure is referred as **coverage.**
- This is a typical artifact due to the sparsity of rating matrices.
- **trick**: fully coverage by simply predicting random ratings for user-item combinations, whose rating are not reliable to be predicted.

# Coverage

- Measures the ability of recommender system to recommend long-tail items
  - **Item coverage** is the percentage of items included in the recommendation list over the number of potential items.
  - **User coverage** is the percentage of users for whom the recommender was able to generate a recommendation lists over the number of potential users.
  - **Catalog coverage** is the percentage of recommended user-item pairs over the total number of potential pairs. The number of recommended user-item pairs can be represented by the length of the recommender lists **L.**
  - **User interaction coverage** is the percentage of rated predictions over the total number of ratings.

# Diversity

- Diversity measures how dissimilar recommended items are for a user

# Diversity

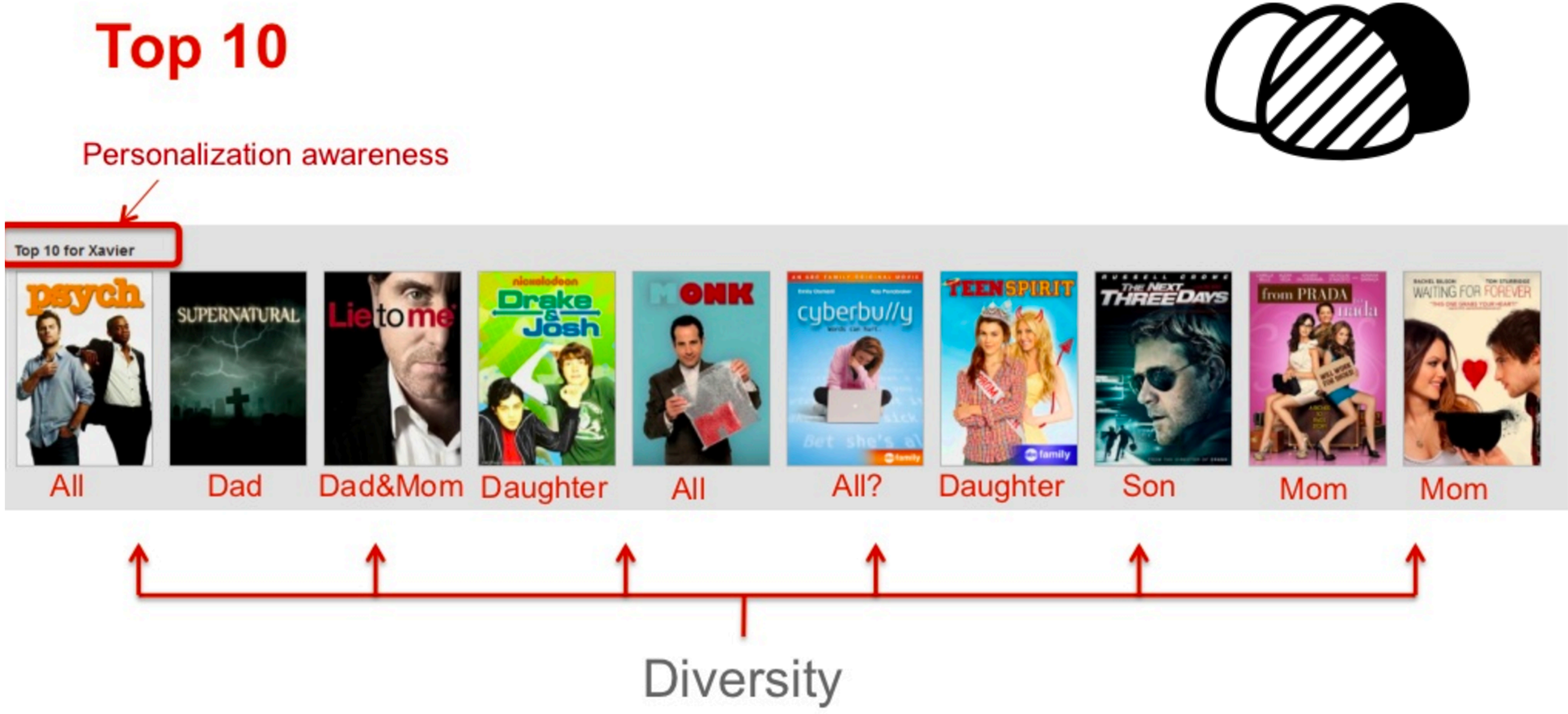
Top Movies for **Dave**


**A clear bad example**

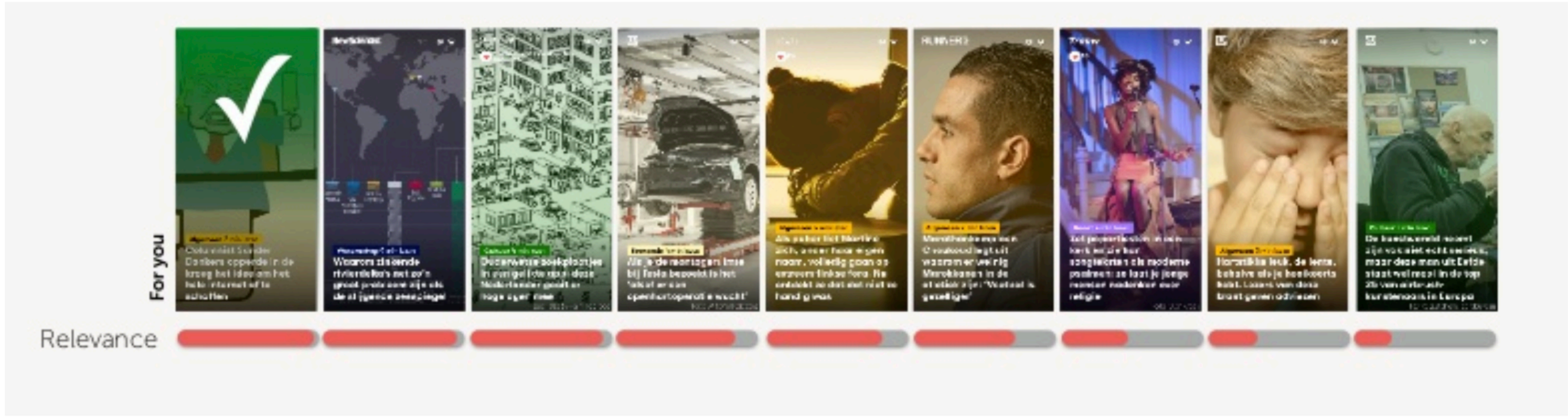
# Diversity

- The list of recommend items should be different between them.
- Diversity can be **measured** in terms of **content-centric similarity** between pair of items. The vector-space of each item description is used for the similarity computation.

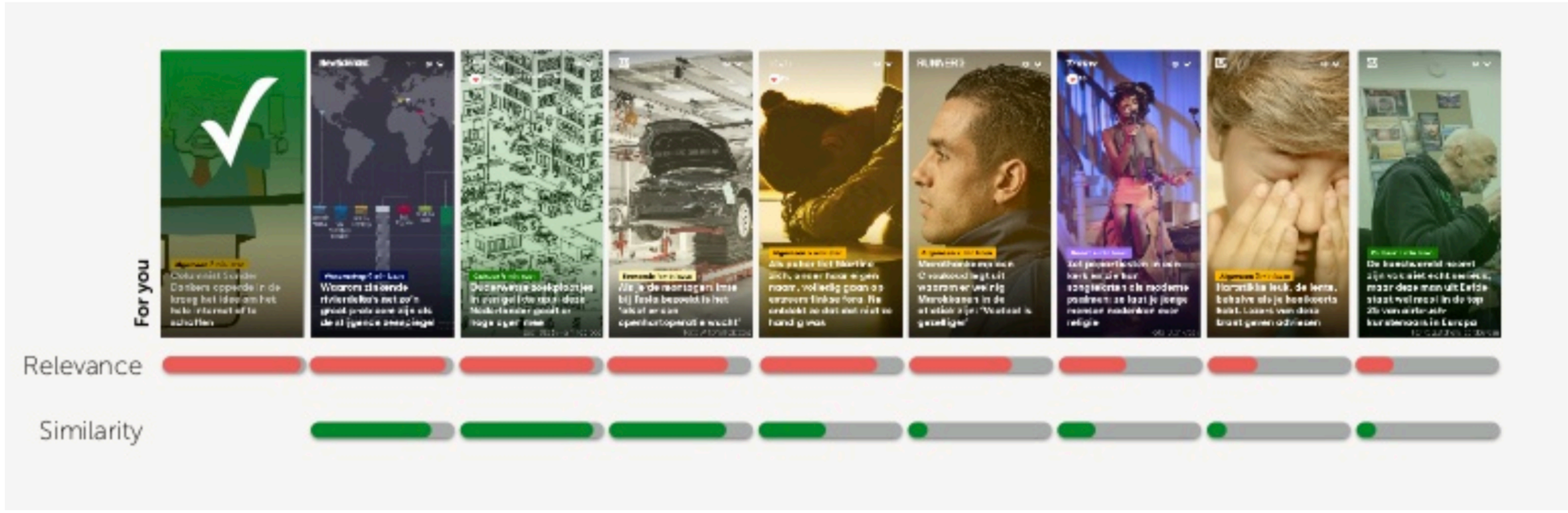
# Diversity


Top 10
Personalization awareness
Top 10 for Xavier
All
Dad
Dad&Mom
Daughter
All
All?
Daughter
Son
Mom
Mom
Diversity

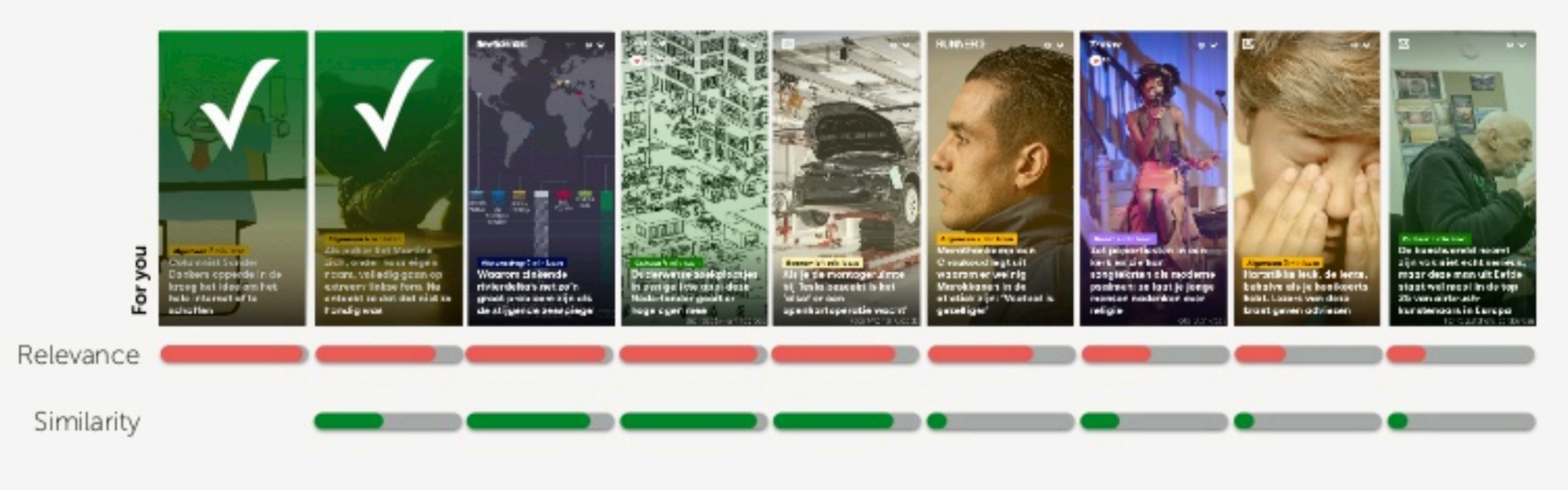
# Diversity



# Diversity



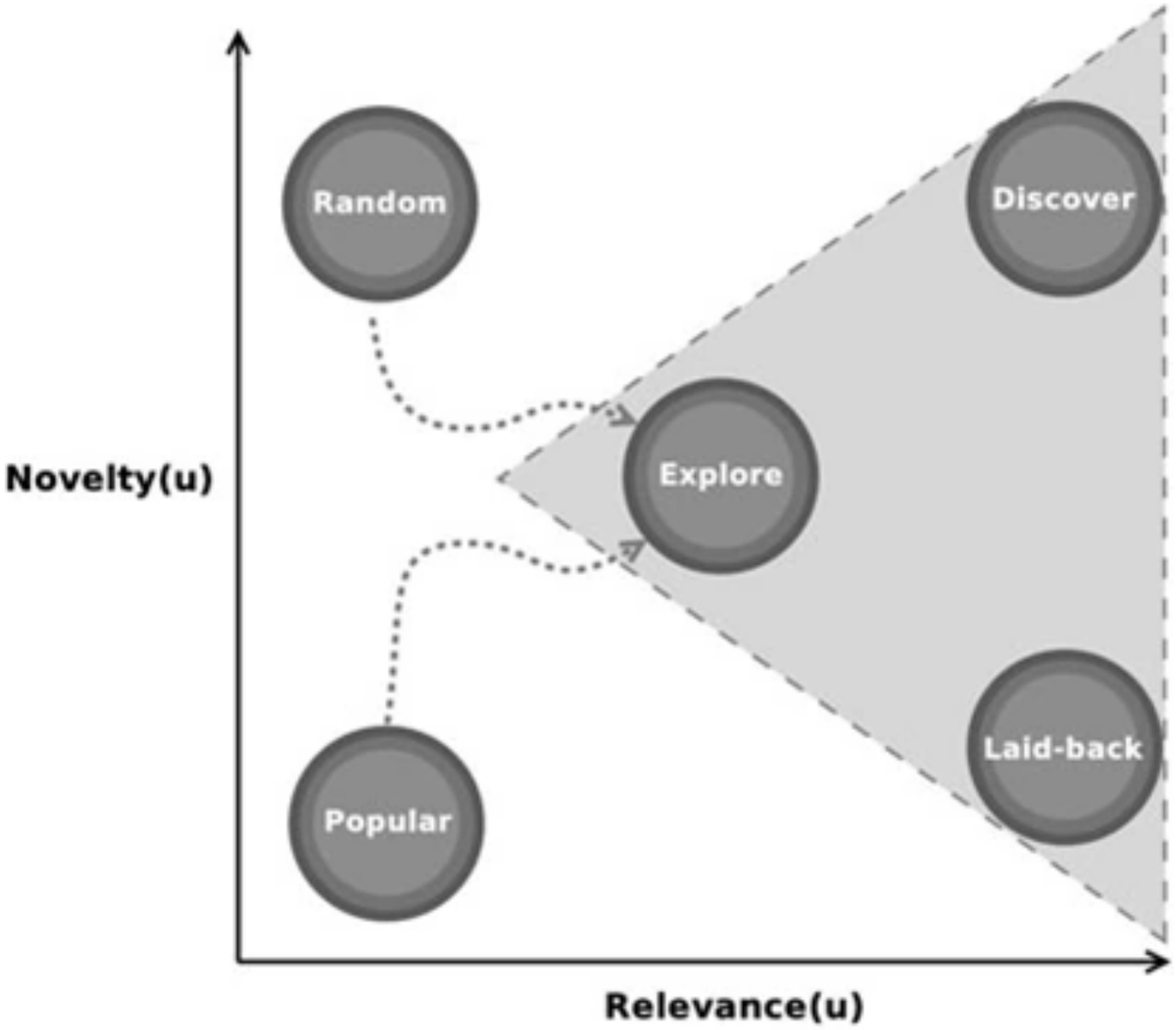
# Diversity



# Novelty


- Novelty determines **how unknown recommended items are to a user**.
- The novelty of a recommender system evaluate the likelihood of a recommender system to give a recommendations to the user what they are not aware of, or what they have not seen before.
- Increase the ability of the user to discover important insights into their likes and dislikes.
- The most natural way to evaluate novelty is done through online experiments.
- In some measure, it is assumed that popular items are less likely to be novel.

# Novelty


Random
Discover
Explore
Laid-back
Popular
Novelty(u)
Relevance(u)

# Novelty


- To measure the surprise of an item, we can determine its probability as a function of its rank for all users.

- As defined by Zhang et al., a user's average novelty over the *n* items in their recommended list ***L*** can then be defined as:

$$Novelty = \sum_{i \in L} \frac{-log_2(P_i)}{n}$$

where

$$P_i = \frac{n - rank_i + 1}{n - 1}$$

# Serendipity

- Serendipity means “**lucky discovery**”. It is a measure that evaluates the level of surprise in successful recommendations.

**Serendipitous** item = **surprising** + **relevant**

# Confidence/Trust

- Even, if the system is accurate, the user can not believe the predictions. When explanations are provided by the recommendations system, the users is more likely to trust the system (only if the explanations are logical).

- Trust is really complicated to be evaluated, usually, is performed by **online surveys**.

- Estimating confidence interval of the predicted ratings.

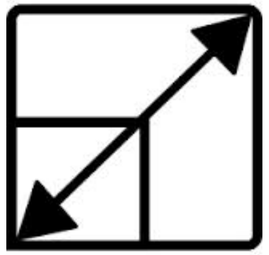
- Trick: show prediction rating of already seen movies

# Robustness/Stability

- How secure/stable is the recommendation system agains of attacks such as fake ratings or when the patterns in the data evolve significantly over time.

Imagine a recommendation system for **restaurants**

# Scalability



- Training time
- Prediction time
- Memory requirements

Too many metrics!
Which is most important?

# How to do trade-off

- Business goal
- Our belief
- Making new algorithm by 3 steps experiments:
  - **Offline** testing
  - **User** survey
  - **Online** testing

# Evaluation Paradigms

**Offline Evaluation**

Historical, such as ratings, are used.
In some cases, temporal information is also provided with the ratings, such as the time-stamp at which the information was obtained.

**User Studies**

Test subjects are actively recruited, and asked to interact
with the recommendation system to perform some actions. Example: satisfaction questionaries

**Online Evaluation**

Online evaluation also leverage user studies except that users are real users of
fully developed  or commercial system.
The user directly plays with the system,
usually different methods are compared with different random uses

# Offline-Evaluation

# Offline methods

- Most common methods for evaluating recommender systems in **research** and **practice perspective**
- **Historical data** sets are used
- We need to define:
  - Task
  - Dataset
  - Partioning
  - Metrics

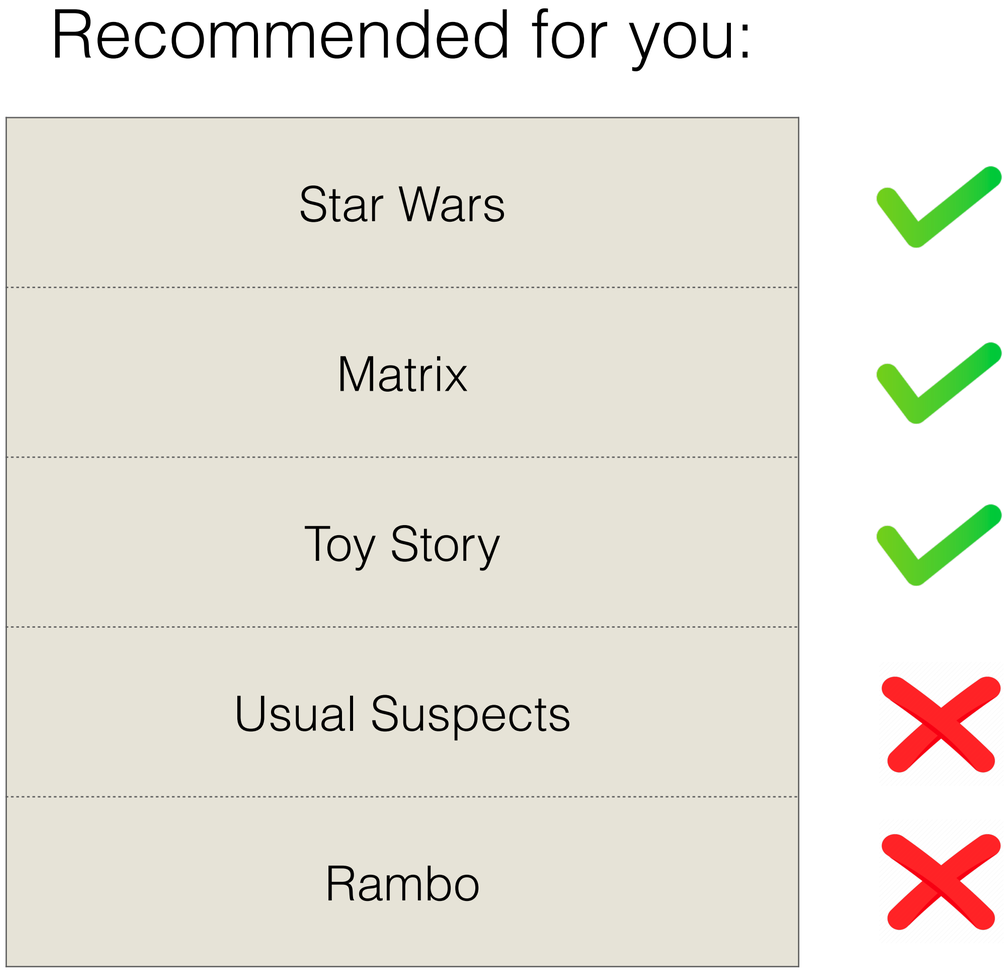
# Offline methods: Task

- Rating prediction

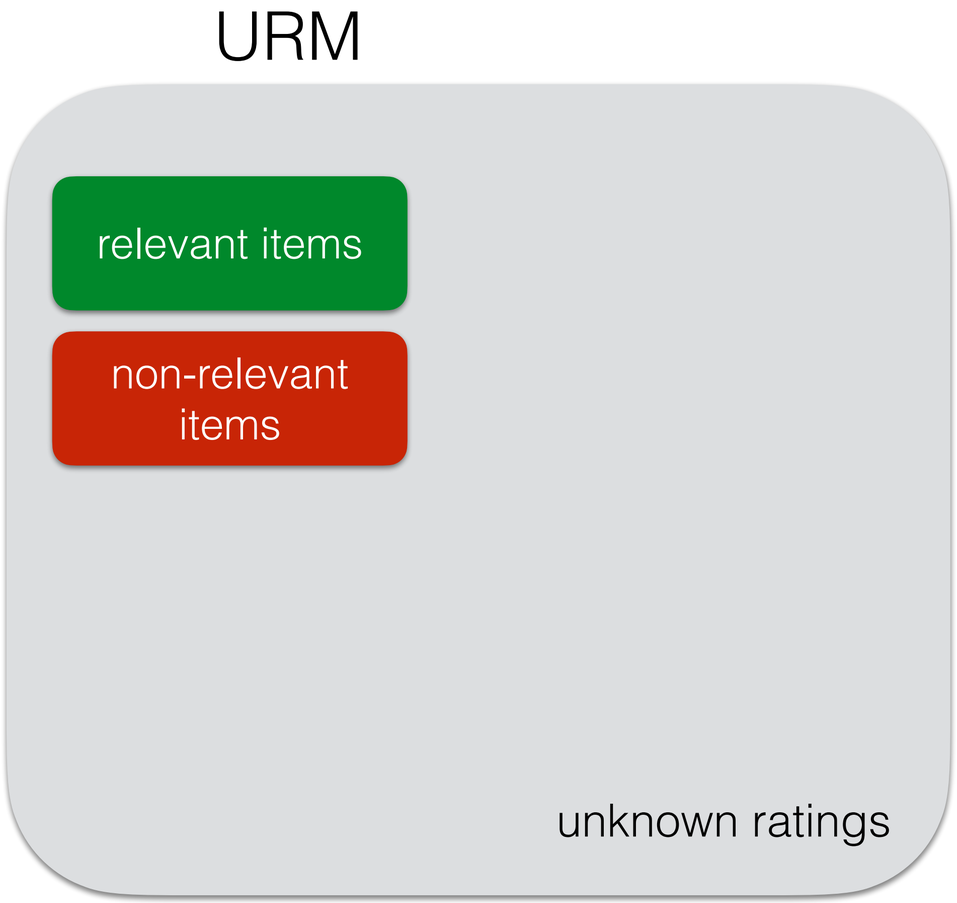
True value = 4
vs.
Predicted value = 3.7

# Offline methods: Task

- Rating prediction
- Top-N recommendation



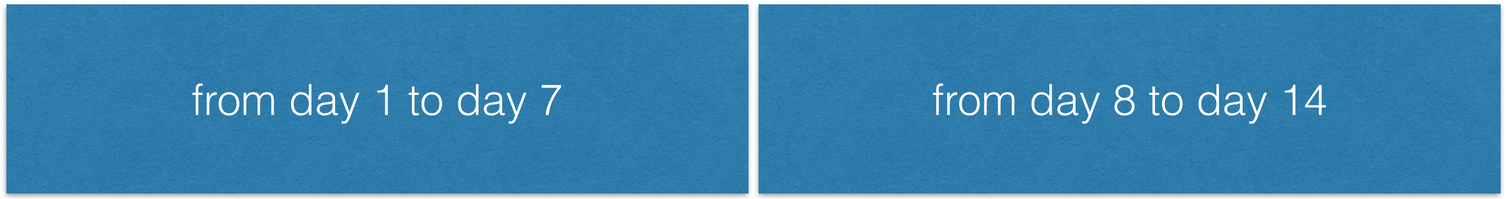
# Offline methods: Dataset



# Offline methods: Data Partition

- You'll need to go into tedious details about the data and how to divide it.
  - In classical supervised problem data is divided in a random fashion. (e.g. training: 90%- test:10%)
  - You must ensure that your training set follows the same distribution of the test set.
    - Stratified sampling can be an option
    - What about new users?
    - What about timestamps
    - You can only learn from past actions

# **Offline methods: Task -** Twitter data




Training + Val
100% data

Test
10%

# Offline methods: Task

- Split by user

| | user 1 | user 2 | user 3 | user 4 |
|---|---|---|---|---|
| 1 | 0068646 | 0422720 | 1300854 | 1300854 |
| 2 | 0113277 | 0454876 | 2170439 | |
| 3 | | 0790636 | 2203939 | |
| 4 | | 0816711 | | |

- Split by timestamp

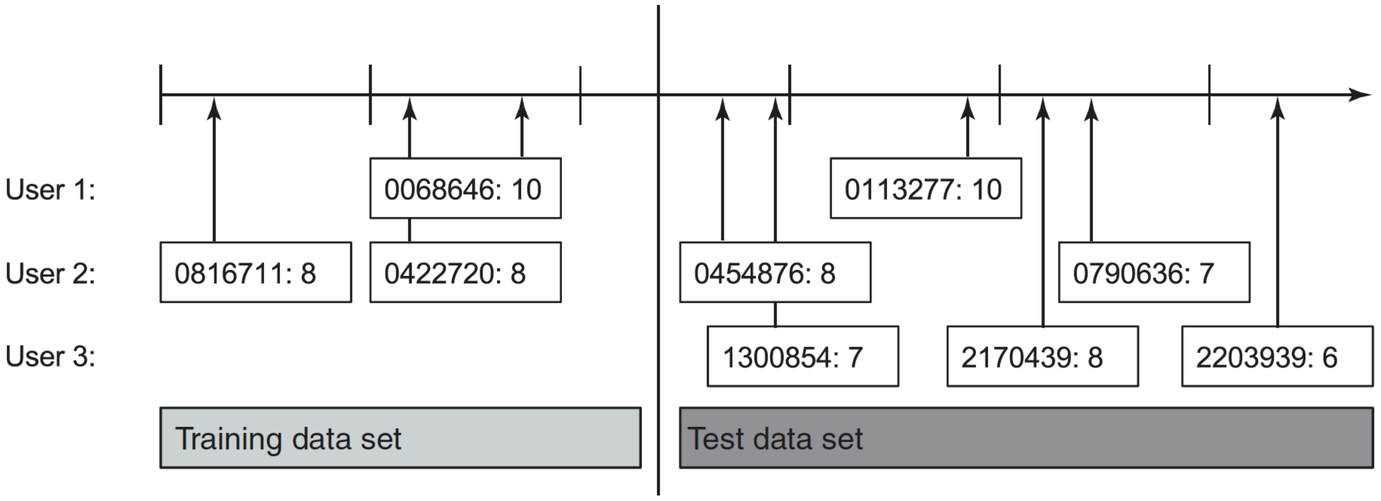



**Figure 9.11 Splitting data by time creates timed snapshots of the data.**

# Offline methods: Metrics

- It will depend on the task:
  - Error metrics such as RMSE or MAE among the most popular

# Offline methods: Metrics

- Usually, ratings are numerical values that need to be estimated.
- **Ratings prediction:**
  - Mean Absolute Error (MAE) computes the deviation between predicted rating and the true rating

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

  - Root Mean Square Error (RMSE) is similar to MAE, but makes more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2}$$

# Offline methods: Metrics

Any **problem** with RMSE or MAE?

these metrics do **not** really **measure the user experience**

# House of Cards

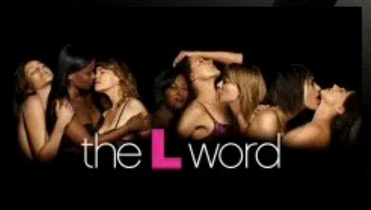
2013 TV-MA 1 Season HD 5.1

Sharks gliding ominously beneath the surface of the water? They're a lot less menacing than this Congressman.

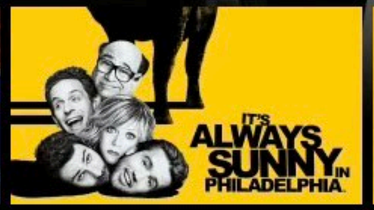
*This winner of three Emmys, including Outstanding Directing for David Fincher, stars Kevin Spacey and Robin Wright.*


## Because you watched Orange Is the New Black



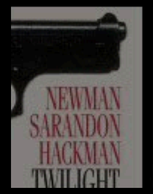








## Because you watched Red Lights








# Offline methods: Metrics

# Offline methods: Metrics

## Rank **Accuracy**

1. Evaluate the rating by using correlation.

# Offline methods: Metrics

## Rank **Accuracy**

- Spearman rank coefficient. Similar to Pearson correlation when using ranking as input.
  - What happens with similar predicted values?
- Kendall rank correlation

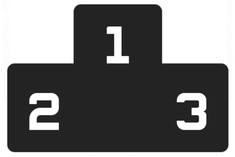
$$\tau_u = \frac{\text{Number of concordant pairs} - \text{Number of discordant pairs}}{\text{Number of pairs in } I_u}$$

# Offline methods: Metrics

## Rank **Accuracy**

1. Evaluate the rating by using correlation.

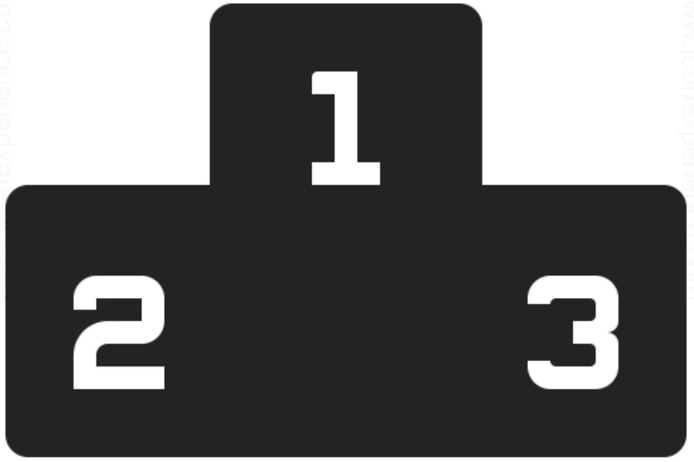
2. **Top-N recommendations**



# Offline methods: Metrics

## Rank **Accuracy**

- **Top-N** performance metric can be also used and in fact, the evaluation with this metric is more closer to what is important for the user, **just those top recommendations**.



# Offline methods: Metrics

## Rank **Accuracy**

**Top-N Recommendations**

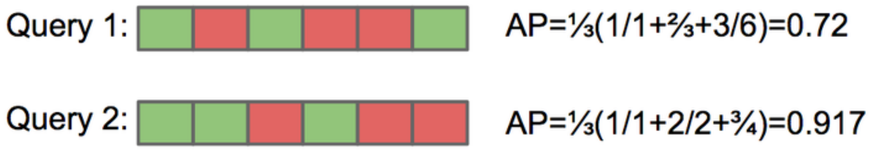
- Populars measures are the following:
  - Mean Average Precision (MAP)
  - Normalized Discounted cumulative gain (NDCG)

# Offline methods: Metrics

## Rank **Accuracy**

- **Mean Average Precision (MAP)** calculates the precision at the position of every corrected item in the ranked results list

$$\text{AveP} = \frac{\sum_{k=1}^{n}(P(k) \times \text{rel}(k))}{\text{number of relevant documents}} \qquad \text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP(q)}}{Q}$$




MAP= (0.72+0.917)/2=0.8185

# Offline methods: Metrics

Rank **Accuracy**

# Case: Kaggle Challenge Expedia

Submissions are evaluated according to the Mean Average Precision @ 5 (MAP@5):

$$MAP@5 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{min(5,n)} P(k)$$

where |U| is the number of user events, P(k) is the precision at cutoff k, n is the number of predicted hotel clusters.

# Offline methods: Metrics

## Rank **Accuracy**

- **Normalized Discounted cumulative gain** (**NDCG**). Two assumptions are made :

  - Highly relevant documents are more useful when appearing earlier in a search engine result list (have higher ranks)

  - Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than non-relevant documents.

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

$$\mathrm{IDCG_p} = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

# Offline methods: Metrics

## Rank **Accuracy**

- We can use the strategy followed by P. Cremonesi et.al.

- In order to measure the precision recall, first the models is trained using the training data, and then, for each item ***i*** rated with 5 stars in the test data set:

  - A set of 100 random unseen movies for the user of the item ***i*** are selected. We assume that these random movies will not be at the same interest than the 5 star movie

  - We predict the rating of the movie of item ***i*** and 100 random unseen movies.

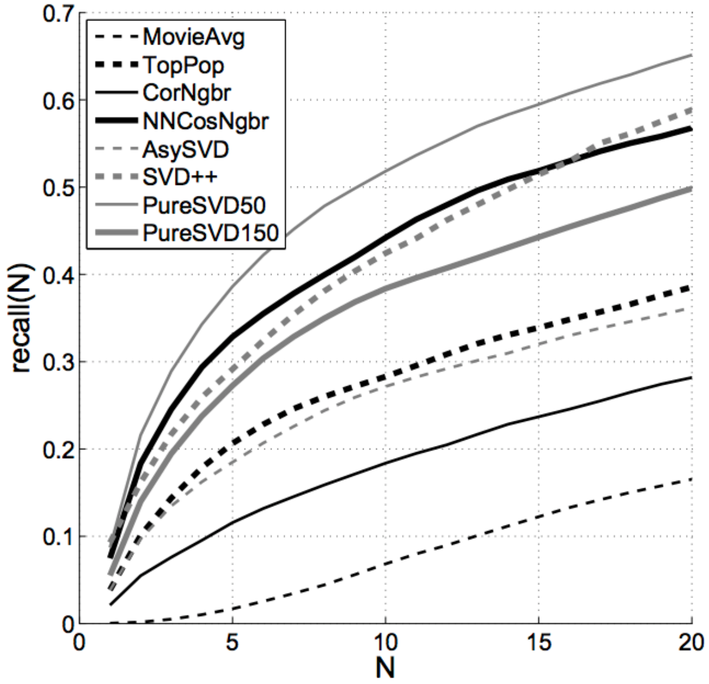
  - We form a rank list by ordering all the 101 item according to the predicted rating. Let denote ***p*** the rank of the test item ***i*** within the list. The best results corresponds to the case the test item ***i*** precedes all the random items (i.e., p=1).

  - A top-N recommendation list by piking the **N** top ranked items from the list. If ***p≤N*** we have a hit. Otherwise we have a miss. Chances of hit increases as **N** is higher.

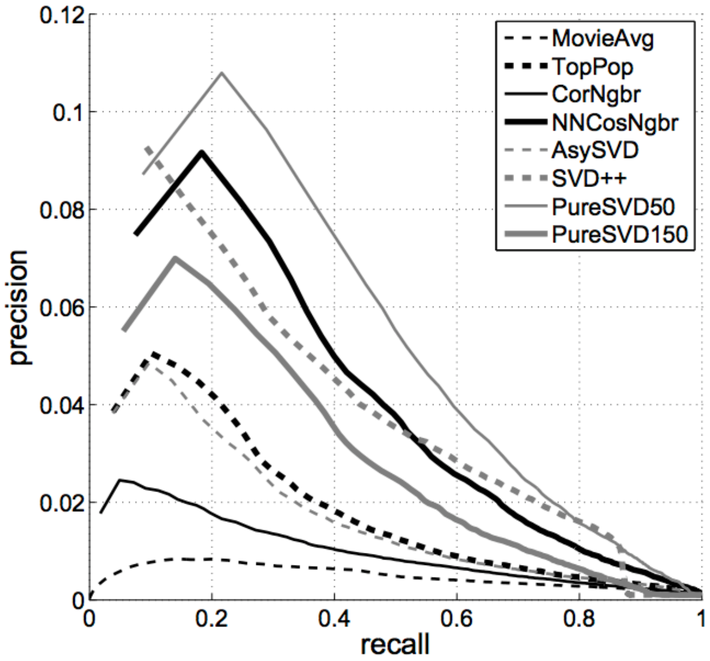
Performance of recommender algorithms on top-n recommendation tasks
P Cremonesi, Y Koren, R Turrin
Proceedings of the fourth ACM conference on Recommender systems, 39-46 612 2010

# Offline methods: Metrics

## Rank **Accuracy**




(a) recall




(b) precision vs recall

**Figure 2: Movielens: (a) recall-at-$N$ and (b) precision-versus-recall on all items.**

Performance of recommender algorithms on top-n recommendation tasks
P Cremonesi, Y Koren, R Turrin
Proceedings of the fourth ACM conference on Recommender systems, 39-46
612 2010

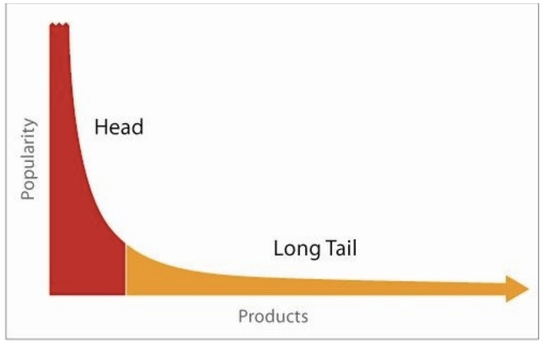
# Offline methods: Metrics

## Rank **Accuracy**

- The accuracy measures are usually heavily influence by rating from populars item. Item that received very few items are practically ignored.

- However, the non-popular items are typically the ones which provides higher profit to the companies.

***Trick**: weight items according to the profit, or remove the those items which are too popular.*



# Offline methods

**Advantage**

Only rely on a dataset. Easy to be obtained

**Disadvantage**

Offline metrics can not reflect business goal

# User surveys & **Online** Evaluation

# Evaluation Framework

# Evaluation Framework

The algorithm counts for only 5% of the relevance of a recommender system

–Francisco Martin, Keynote Speaker RecSys 2009

**Solution**: try other aspects

# Evaluation Framework

# Evaluation Framework

"Testing a recommender against a random videoclip system, the number of clicked clips and total viewing time went down!"

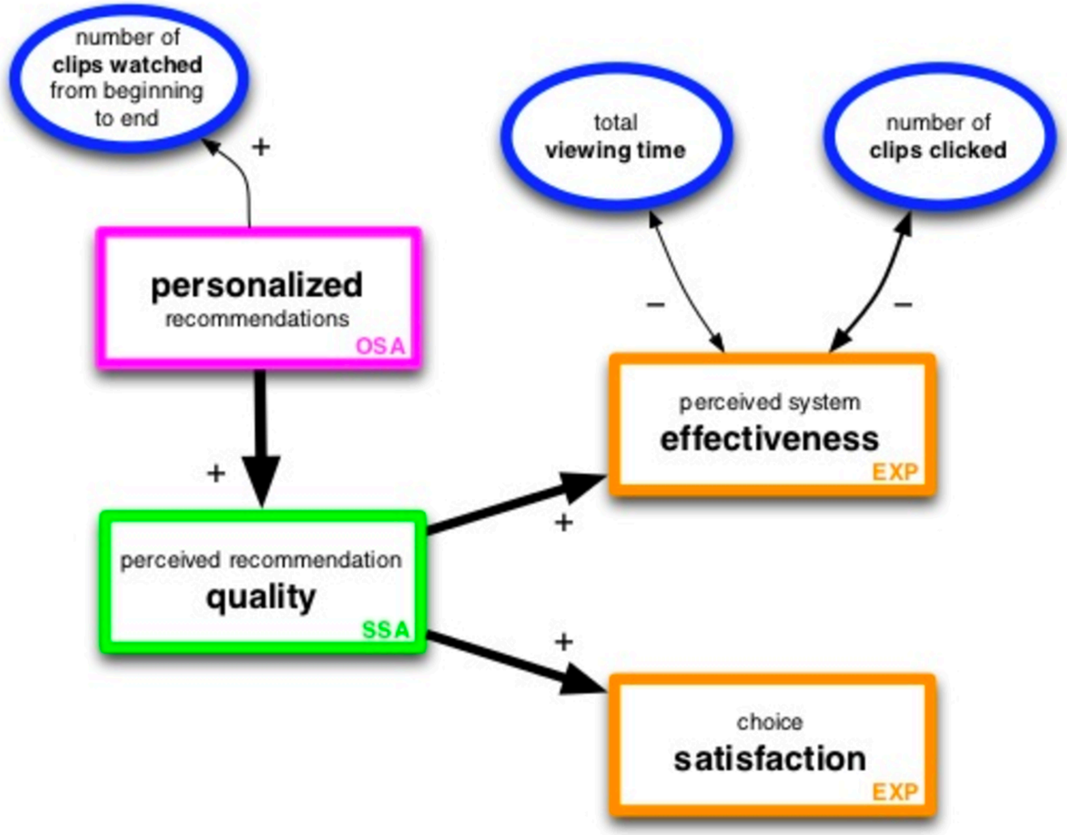
Receiving recommendations and providing feedback: The user-experience of a recommender system
BP Knijnenburg, MC Willemsen, S Hirtbach
International Conference on Electronic Commerce and Web Technologies, 207-216

15 2010

# Evaluation Framework

- Behavior is **hard** to understand
  - Relationship between behavior and satisfaction is not always trivial

# Evaluation Framework




Receiving recommendations and providing feedback: The user-experience of a recommender system
BP Knijnenburg, MC Willemsen, S Hirtbach
International Conference on Electronic Commerce and Web Technologies, 207-216

15 2010

# User Studies

Users are actively recruited and asked to interact with the recommender systems to perform specific task.

Feedback is collected from them before and after the interactions.

These data are then used to make inferences about likes or dislikes about the user.

# Participants

Your colleagues?

Study link on Facebook/Twitter?

# Participants

Are connections or colleagues the typical users of your system?

- They may have more knowledge of the field of study
- They may feel more excited about the system
- They may know what the experiment is about
- They probably want to please you

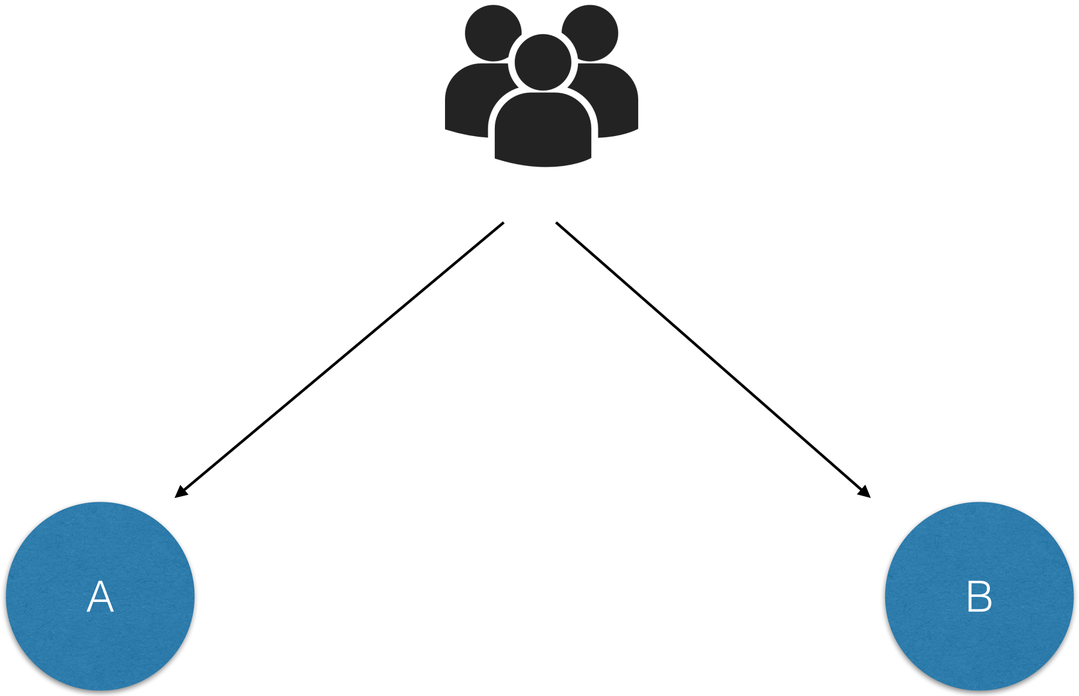
You should sample from a **target population**

- **An unbiased sample of users can heart your experiment**

# Make some questions!

**“Are our users more satisfied if our news recommenders shows only recent items?”**

# Make some questions!




The first 40 participants will get the baseline, the next 40 will get the treatment “

# Make some questions!

The first 40 participants will get the baseline, the next 40 will get the treatment “

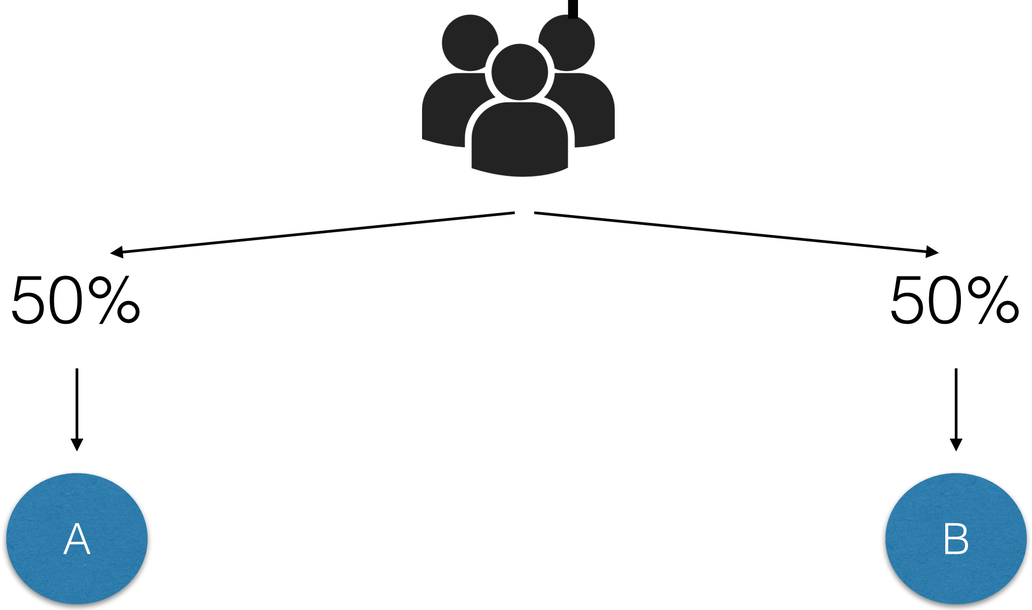
**Caution:**

**These to groups cannot be expected to be similar!**

Some new item may affect one group but not the other

**Randomize** the assignments of conditions to participants. It may reduce participation variation (but not eliminate)

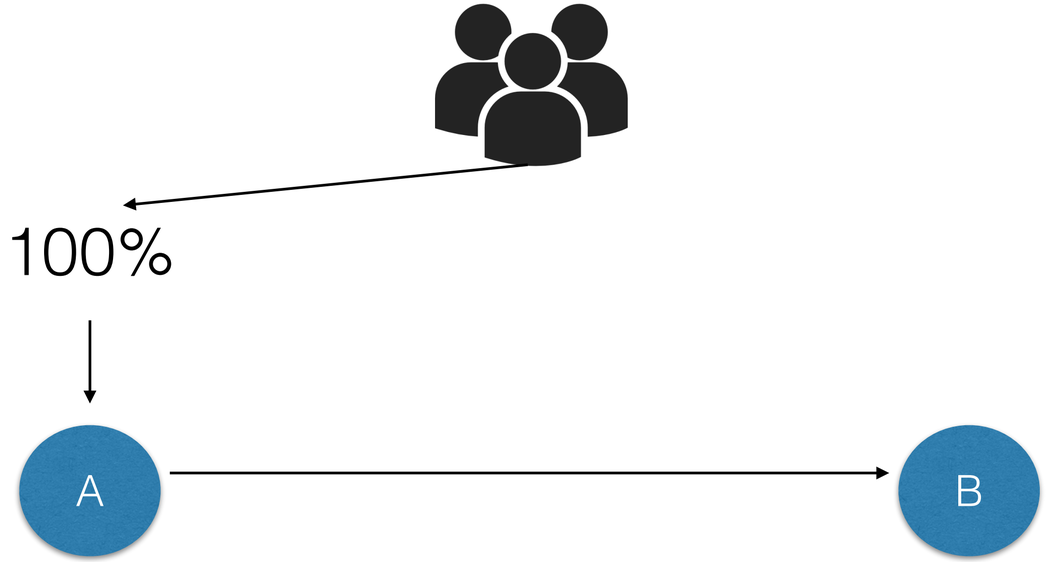
# Make some questions!




**Between-subjects design:** Randomly assign half of the participants to A and half to B.

Realistic interaction; Manipulation hidden from the user; Many participants needed

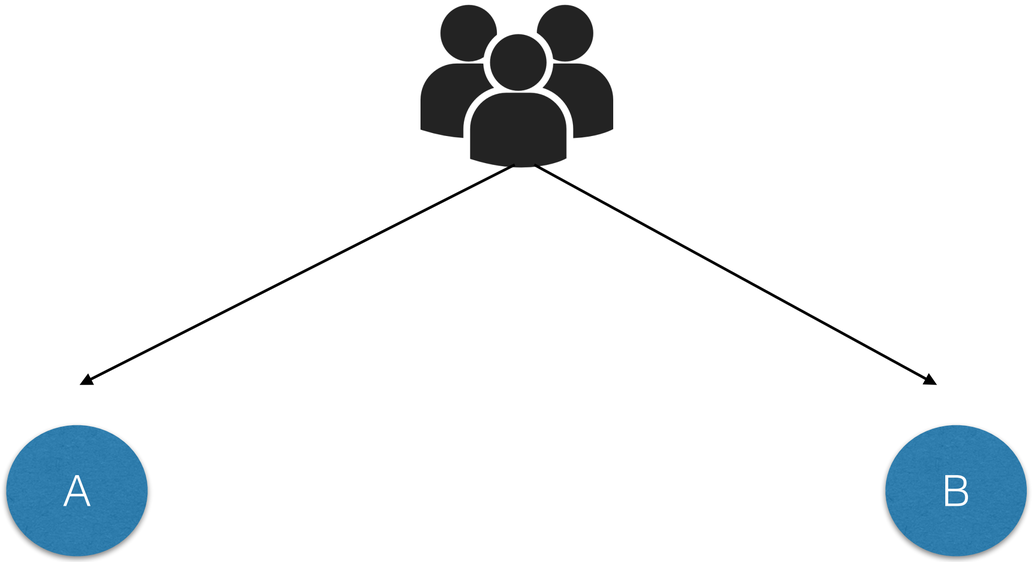
# Make some questions!




**Within-subjects design:** Give Participants A first, then B

Remove subject subjectivity, Participants may see the manipulation, spill-over effect

# Make some questions!




**Within-subjects design:** Show A and B simultaneously

Remove subject subjectivity, Participants can compare conditions,
Not realistic interaction

# Manipulations

- Let’s test an algorithm against a random recommendations!!
- What should we say to the participant?

Beware of the **placebo** effect!

"We were demonstrating our new recommender to a client. They were amazed by how well it predicted their preferences!"

"Later we found out that we forgot to activate the algorithm: the system was giving completely random recommendations."

(anonymized)

# How do we measure the test?

“To measure satisfaction we can ask users whether they liked the system (on a 5-stars rating scale)”

# How do we measure the test?

- Does the question **mean** the same to everyone?

Peter liked the system because it is **convenient**

Mary liked the system because it is **easy to be used**

David liked the system because **recommendations are good**

A single question is not enough to establish content validity

**We use to need a multi-item measurements scale**

# Measurements

- Perceived system effectiveness:
  - Using the system is annoying
  - The system is useful
  - Using the system make me happy
  - Overall, I’m satisfied with the system
  - I would recommend the system to other
  - I would quickly abandon using the system

# Measurements

- Which is best for a questionary?

  A. The recommendations were not really novel

  B. The recommendations felt outdated

# Measurements

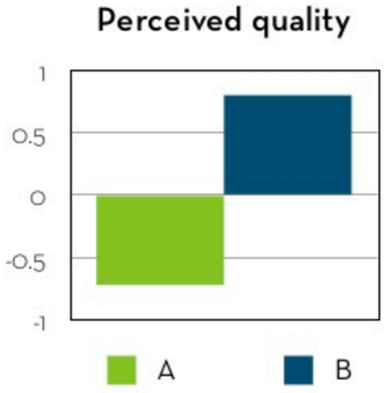
- Chose simple over-specialized words
  - Participants may not know that they are dealing with a recommender system.
- Avoid double-barreled questions:
  - **BAD**: “The recommendations were relevant and fun”

# Analysis!

Statistical evaluation of the results?

# Analysis!

- Does these two algorithms lead to a different level of perceived quality?

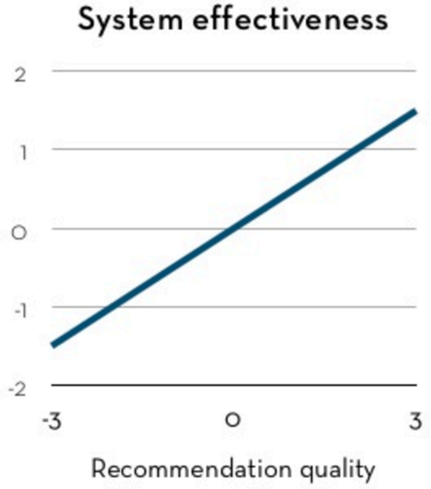



- T-test

# Analysis!

- Does the perceived recommendation quality influenced the system effectiveness

- Linear regression

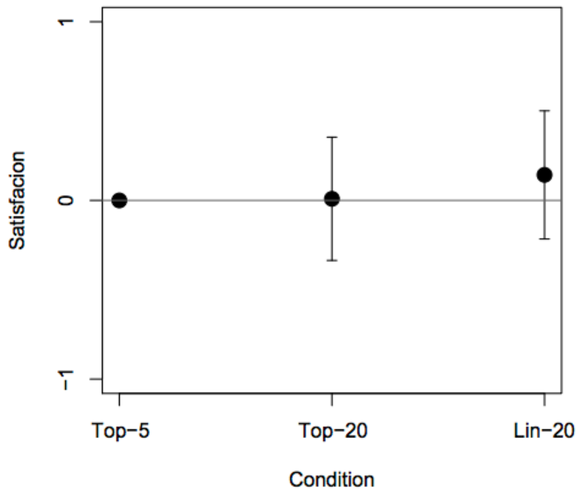


# Analysis!

- What is the effect of the number of recommendations?
- What about the composition of the recommendation list?

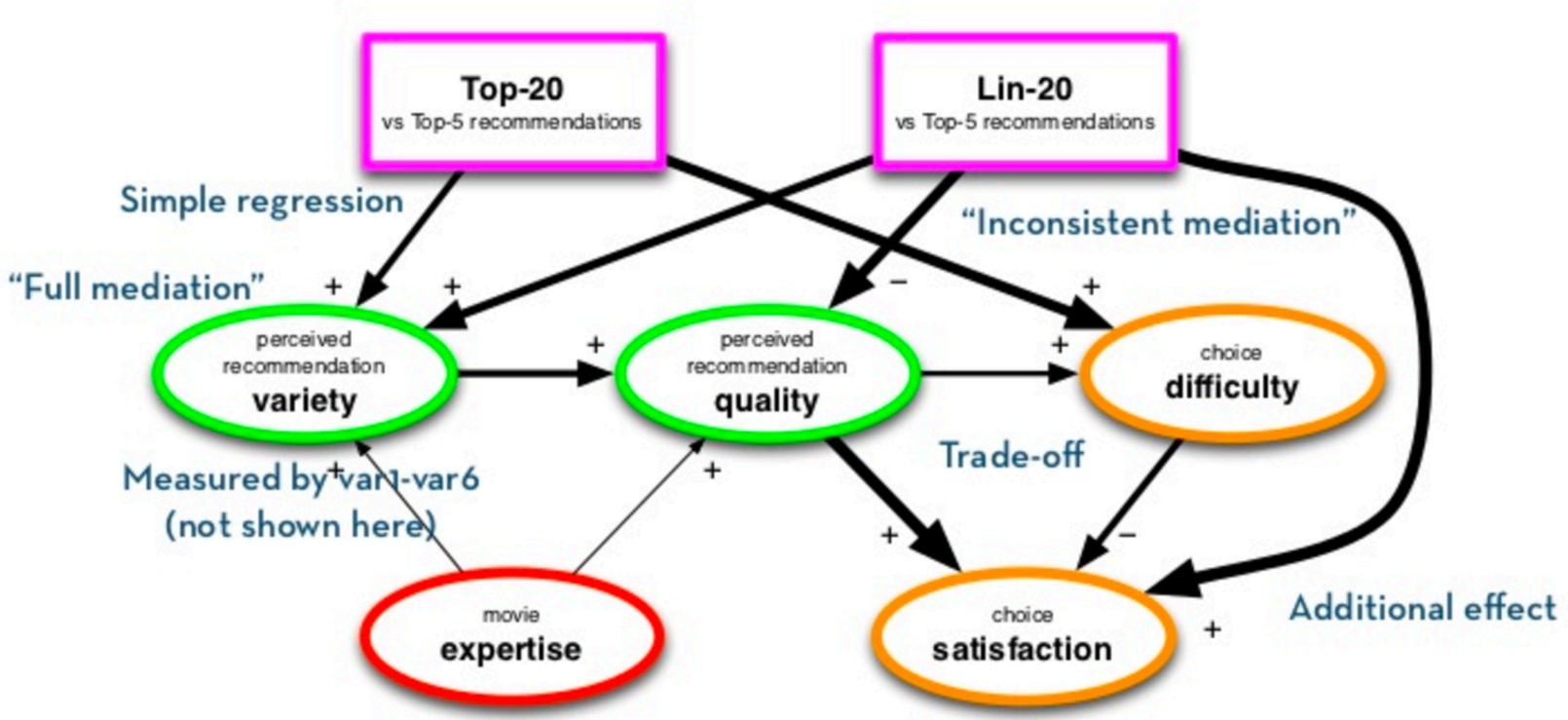
3 different options:

- Top-5 : 1,2,3,4,5
- Top-20: 1,2,3,4,5,6,7,8,9,10,11…20
- Lin-20: 1,2,3,4,5,99,199,299,399,499,599..




Understanding choice overload in recommender systems
D Bollen, BP Knijnenburg, MC Willemsen, M Graus
Proceedings of the fourth ACM conference on Recommender systems, 63-70
99 2010

# Analysis!




Understanding choice overload in recommender systems
D Bollen, BP Knijnenburg, MC Willemsen, M Graus
Proceedings of the fourth ACM conference on Recommender systems, 63-70

99 2010

# Analysis!

- What is the combined effect of a list diversity and list length on perceived recommendations?

# User Studies

**Advantage**

Can get subjective measures

Low risk than evaluating online

**Disadvantage**

Higher cost than offline experiments

Some results may have not statistical significance

Users may have different behaviors under test environment than on real environment

It’s difficult to design a double blink experiments

# Online methods

Once you're satisfied that your recommender algorithm,
it's time to deploy the system and test it out on real humans

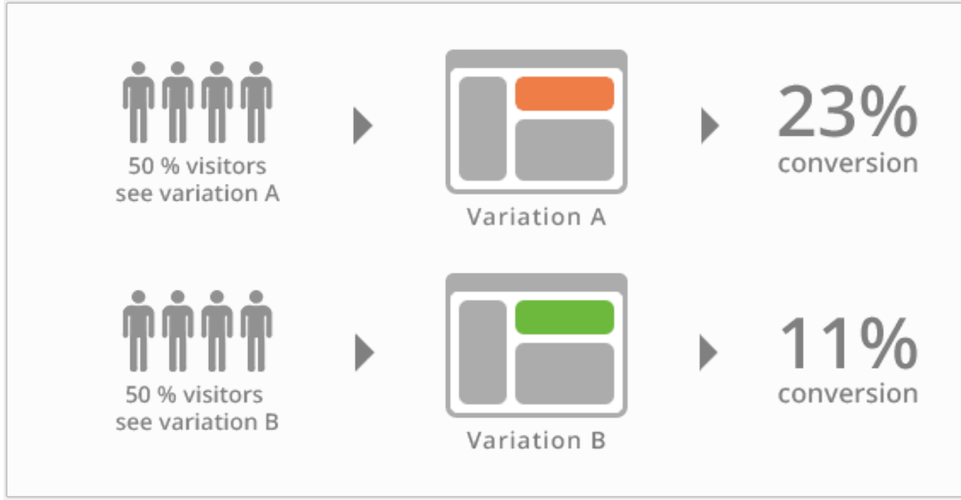
Only when you have lots of users !!

# Online methods

- User participation is essential
- E.g. convergence ratio of users clicking
- Methods like ***A/B Testing*** measure the direct impact of the recommender system on the users
  - Goal: improve conversion rate on profitable items is one of the most important goals

# Online methods

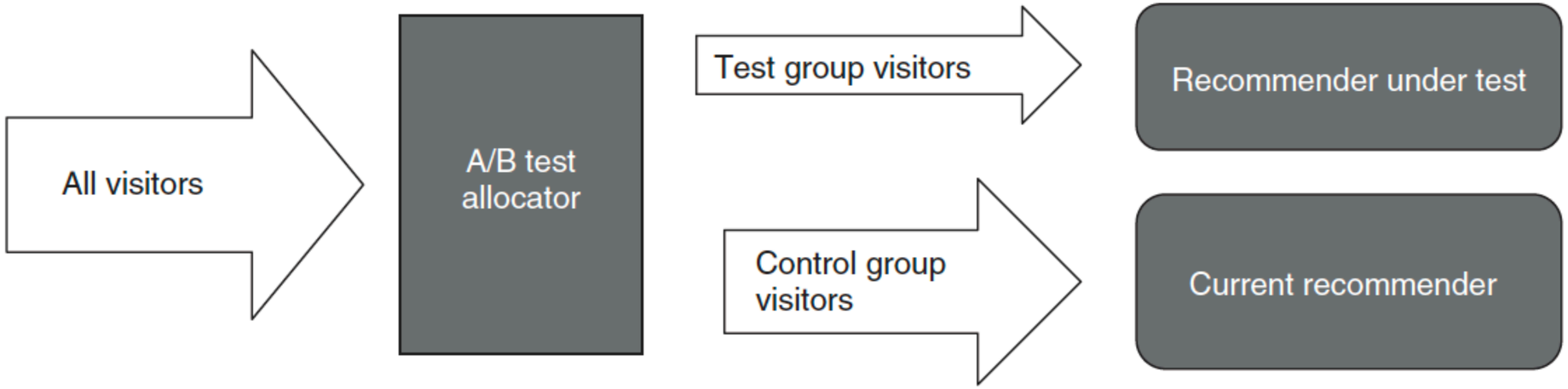
- A/B testing
  - Measure differences in metrics across **statistically identical  populations** that each experience a different algorithm
  - Overall Evaluation Criteria (OEC) == " member retention"
    - Use long-term metrics whenever possible
    - Short-term metrics can be informative and allow faster decisions
      - But, not always aligned with OEC



# A/B Testing

- With A/B testing, you can test a new recommender algorithm by redirecting **a small part of the traffic to the new recommender** and letting the customers indicate which is better.

- Users won't know that they're part of an experiment. And that's the whole point.



# It's All A/Bout Testing: The Netflix Experimentation Platform

Netflix Technology Blog Follow
Apr 29, 2016 · 11 min read

Ever wonder how Netflix serves a great streaming experience with high-quality video and minimal playback interruptions? Thank the team of engineers and data scientists who constantly A/B test their innovations to our adaptive streaming and content delivery network algorithms. What about more obvious changes, such as the complete redesign of our UI layout or our new personalized homepage? Yes, all thoroughly A/B tested.

“All models are wrong, but some are are useful”

–George Box, 1976

# Other methods

## Uplift-based Evaluation and Optimization of Recommenders

Masahiro Sato
sato.masahiro@fujixerox.co.jp
Fuji Xerox

Janmajay Singh
janmajay.singh@fujixerox.co.jp
Fuji Xerox

Sho Takemori
takemori.sho@fujixerox.co.jp
Fuji Xerox

Takashi Sonoda
takashi.sonoda@fujixerox.co.jp
Fuji Xerox

Qian Zhang
qian.zhang@fujixerox.co.jp
Fuji Xerox

Tomoko Ohkuma
ohkuma.tomoko@fujixerox.co.jp
Fuji Xerox

### ABSTRACT

Recommender systems aim to increase user actions such as clicks and purchases. Typical evaluations of recommenders regard the purchase of a recommended item as a success. However, the item may have been purchased even without the recommendation. An uplift is defined as an increase in user actions caused by recommendations. Situations with and without a recommendation cannot both be observed for a specific user-item pair at a given time instance, making uplift-based evaluation and optimization challenging. This paper proposes new evaluation metrics and optimization methods for the uplift in a recommender system. We apply a causal inference framework to estimate the average uplift for the offline evaluation of recommenders. Our evaluation protocol leverages both purchase and recommendation logs under a currently deployed recommender system, to simulate the cases both with and without recommendations. This enables the offline evaluation of the uplift for newly generated recommendation lists. For optimization, we need to define positive and negative samples that are specific to an uplift-based approach. For this purpose, we deduce four classes of items by observing purchase and recommendation logs. We derive the relative priorities among these four classes in terms of the uplift and use them to construct both pointwise and pairwise sampling methods for uplift optimization. Through dedicated experiments with three public datasets, we demonstrate the effectiveness of our optimization methods in improving the uplift.

### 1 INTRODUCTION

One of the major goals of recommender systems is to induce positive user interactions, such as clicks and purchases. Because increases in user interactions directly benefit businesses, recommender systems have been utilized in various domains of industry.

Recommendations are typically evaluated in terms of purchases[1] of recommended items. However, these items may have been purchased even without recommendations. For a certain e-commerce site, more than 75% of the recommended items that were clicked would have been clicked even without the recommendations [42]. We argue that the true success of recommendations is represented by the increase in user actions caused by recommendations. Such an increase affected purely by recommendations is called an uplift. The development of a recommender should focus more on the uplift than the accurate prediction of user purchases.

However, evaluating and optimizing the uplift is difficult, owing to its unobservable nature. An item is either recommended or not for a specific user at a given time instance, so the uplift cannot be directly measured for a given recommendation. This means that there is no ground truth for training and evaluating a model.

Previous studies targeting uplift construct purchase prediction models incorporating recommendation effects [2, 40]. The items recommended are ones that have the largest differences between the predicted purchase probabilities for cases with and without recommendations. Another approach builds two prediction models: