GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

RECOMMENDATION ENGINES

I. DATA TYPES

II. CONTENT-BASED FILTERING

III. COLLABORATIVE FILTERING

IV. THE NETFLIX PRIZE

A recommendation system aims to match users to products/items/brand/etc that they likely haven’t experienced yet and/or predict a users preference based on past observations.

A recommendation system aims to match users to products/items/brand/etc that they likely haven’t experienced yet and/or predict a users preference based on past observations.

A **ranking** or **prediction** is produced by analyzing other user/item ratings (and sometimes item characteristics) to provide personalized recommendations to users.

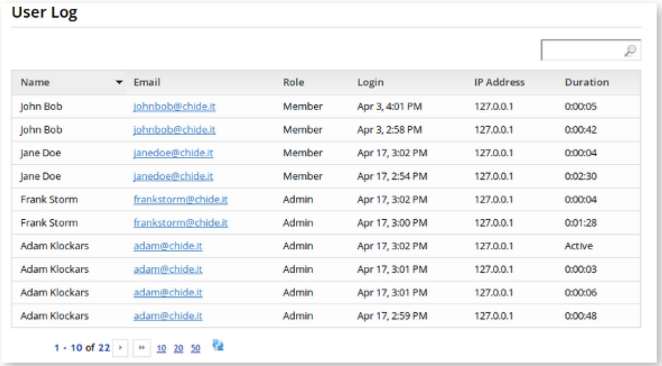
I. TYPES OF DATA

THE KIND OF RECOMMENDATIONS YOU CAN GIVE, ARE DEPENDENT ON THE DATA YOU HAVE.

WE NEED DATA TO RECOMMEND.

- Preferences
- Ratings
- Item meta-data
- User Behavior

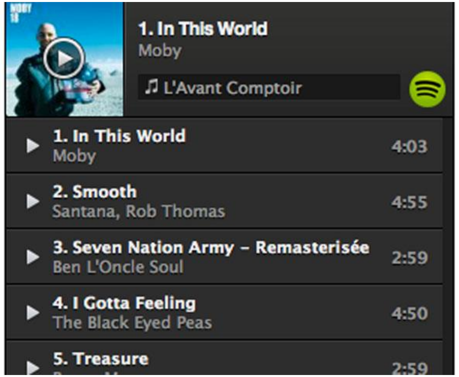
EXAMPLES – TYPES OF DATA



User Log

Name	Email	Role	Login	IP Address	Duration
John Bob	johnbob@chide.it	Member	Apr 3, 4:01 PM	127.0.0.1	0:00:05
John Bob	johnbob@chide.it	Member	Apr 3, 2:58 PM	127.0.0.1	0:00:42
Jane Doe	janedoe@chide.it	Member	Apr 17, 3:02 PM	127.0.0.1	0:00:04
Jane Doe	janedoe@chide.it	Member	Apr 17, 2:54 PM	127.0.0.1	0:02:30
Frank Storm	frankstorm@chide.it	Admin	Apr 17, 3:02 PM	127.0.0.1	0:00:04
Frank Storm	frankstorm@chide.it	Admin	Apr 17, 3:00 PM	127.0.0.1	0:01:28
Adam Klockars	adam@chide.it	Admin	Apr 17, 3:02 PM	127.0.0.1	Active
Adam Klockars	adam@chide.it	Admin	Apr 17, 3:01 PM	127.0.0.1	0:00:03
Adam Klockars	adam@chide.it	Admin	Apr 17, 3:01 PM	127.0.0.1	0:00:06
Adam Klockars	adam@chide.it	Admin	Apr 17, 2:59 PM	127.0.0.1	0:00:48

1 - 10 of 22 › » 10 20 50



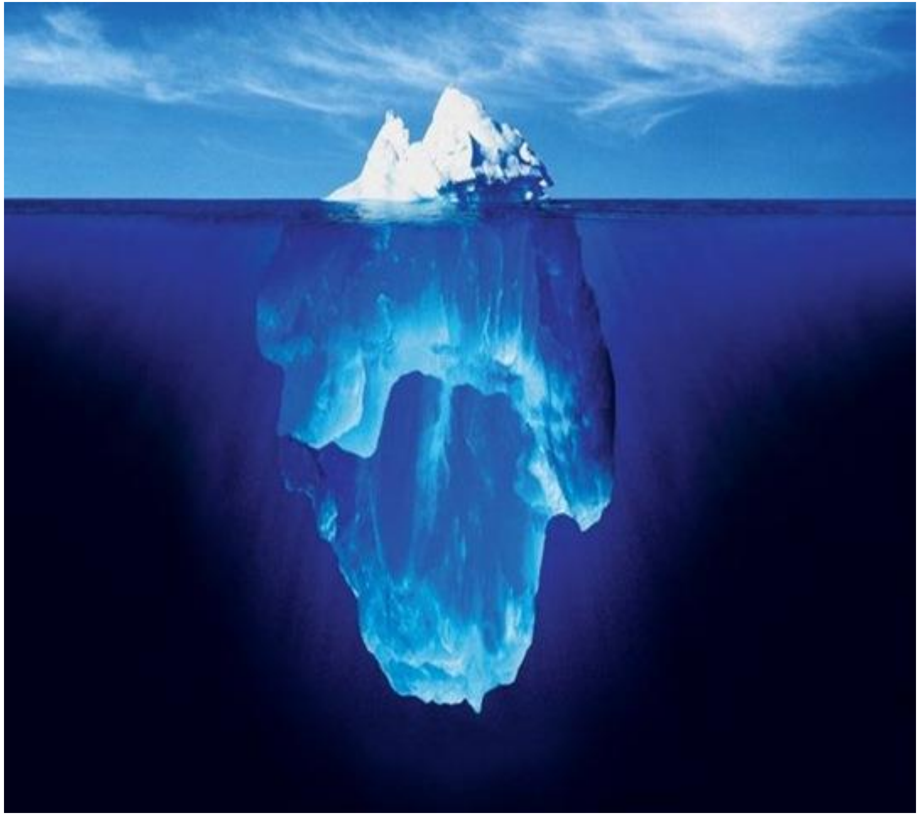
Ratings
Upvotes / Downvotes
Weighted Scale
Grades
Relevance Feedback

Access Logs
Session Lengths
Time spent on a page
Clicks / Non-Clicks
Purchase History
Product Descriptions

Listening History
Playlist Creates
Follows / Unfriend
Impressions
Email Reads / Impressions

EXAMPLES – TYPES OF DATA

Recommenders need feedback to be useful.

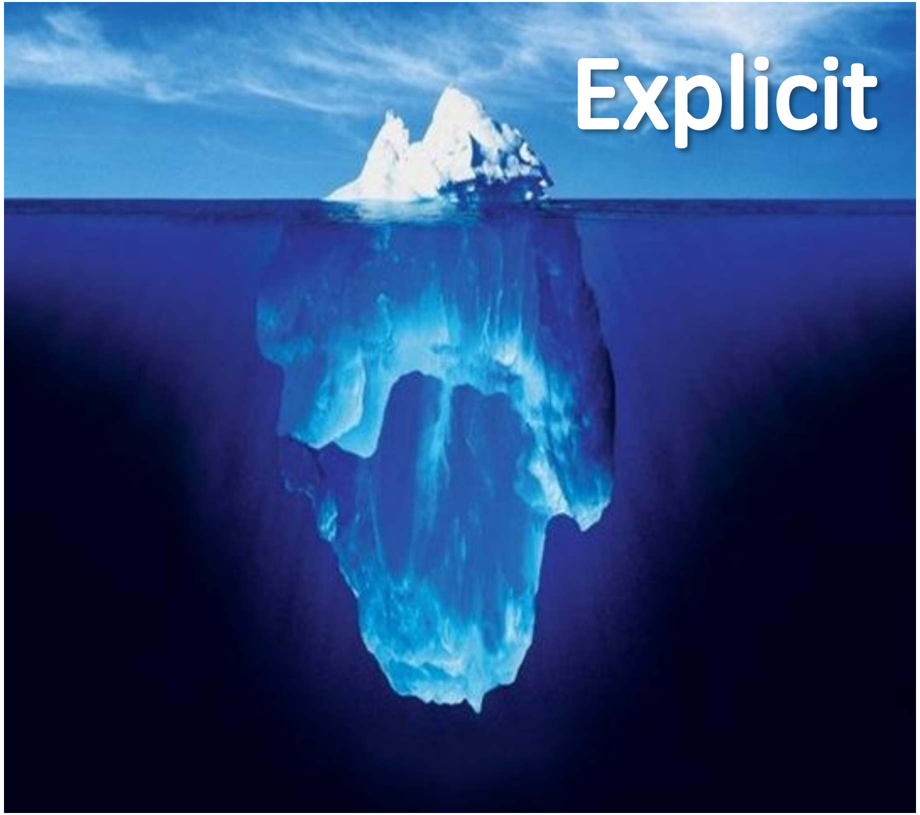


EXAMPLES – TYPES OF DATA

Recommenders need feedback to be useful.

Explicit

- Explicitly given
- Pro-actively acquired
- Expensive to collect



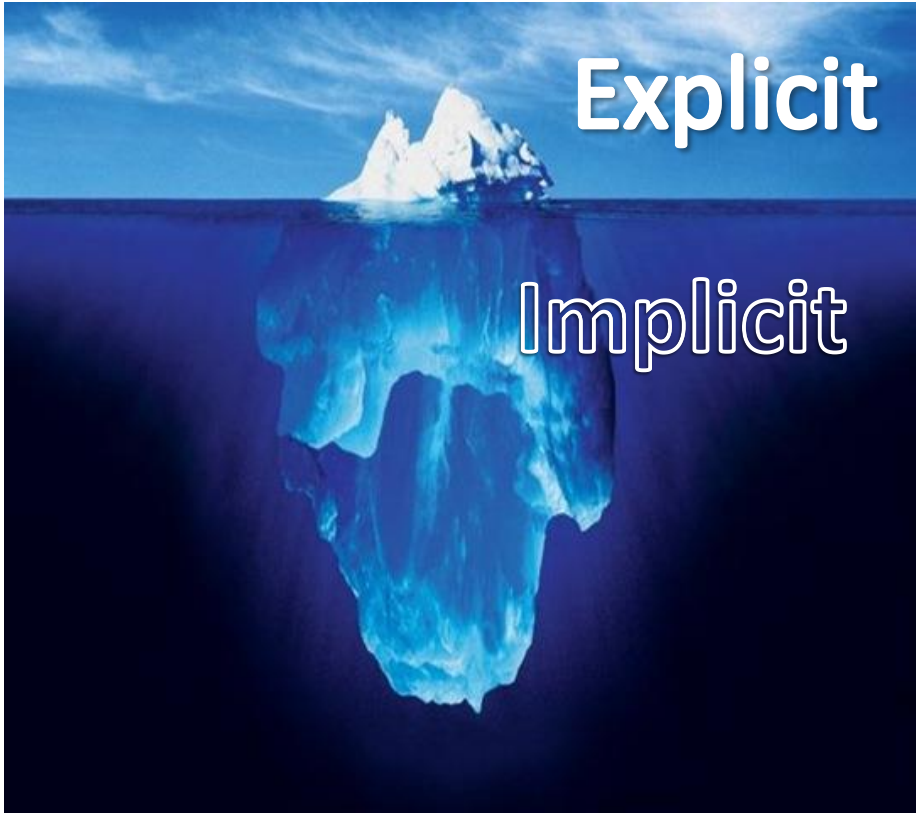
Recommenders need feedback to be useful.

Explicit

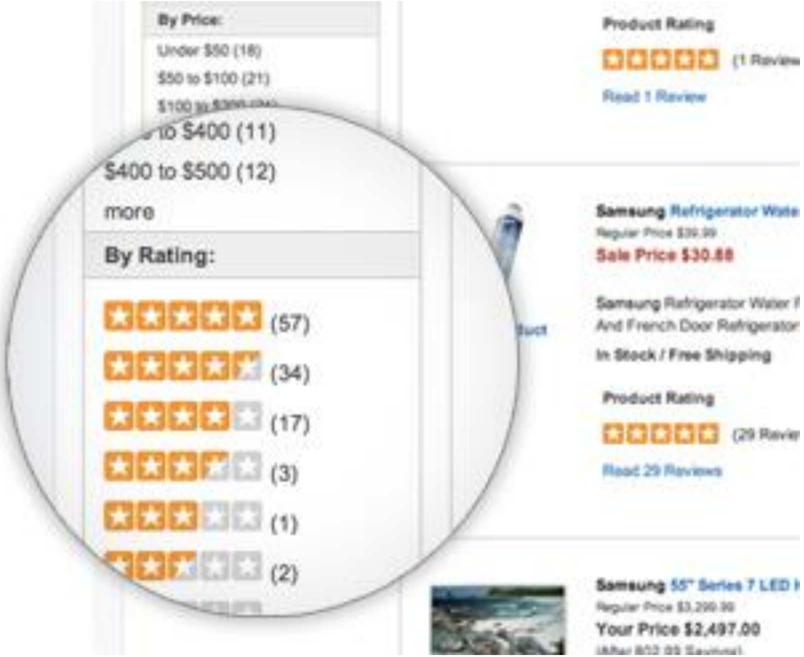
- Explicitly given
- Pro-actively acquired
- Expensive to collect

Implicit

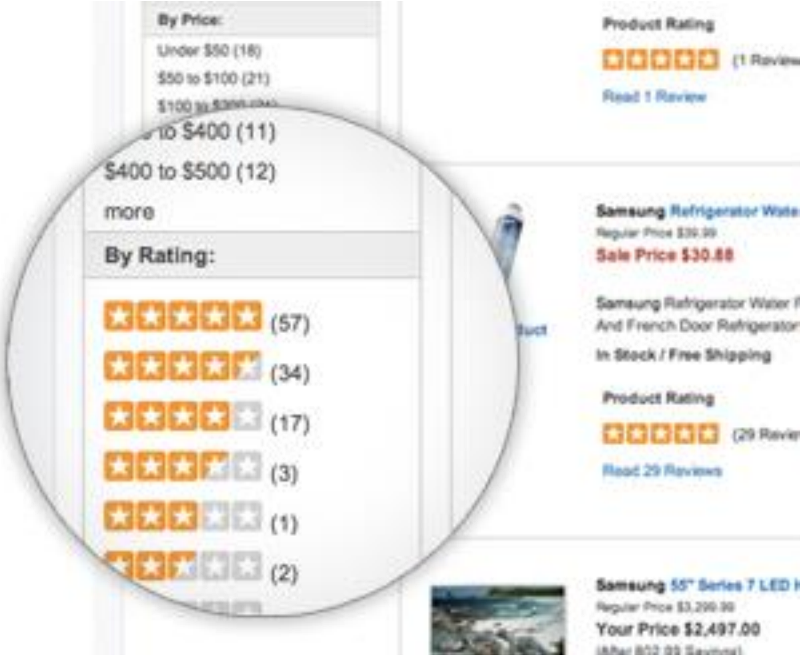
- Indirectly given
- Larger quantity
- Latent qualities



Explicit or Implicit?

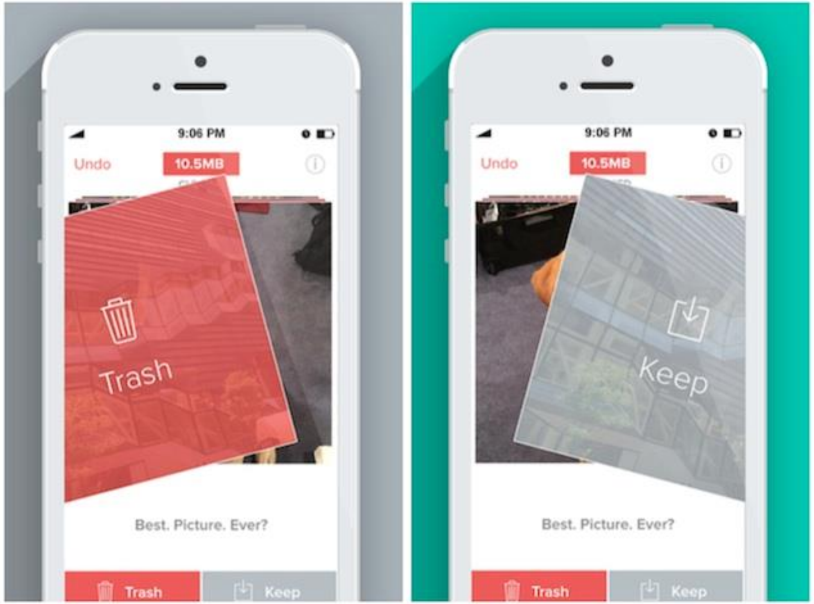


Explicit or Implicit?

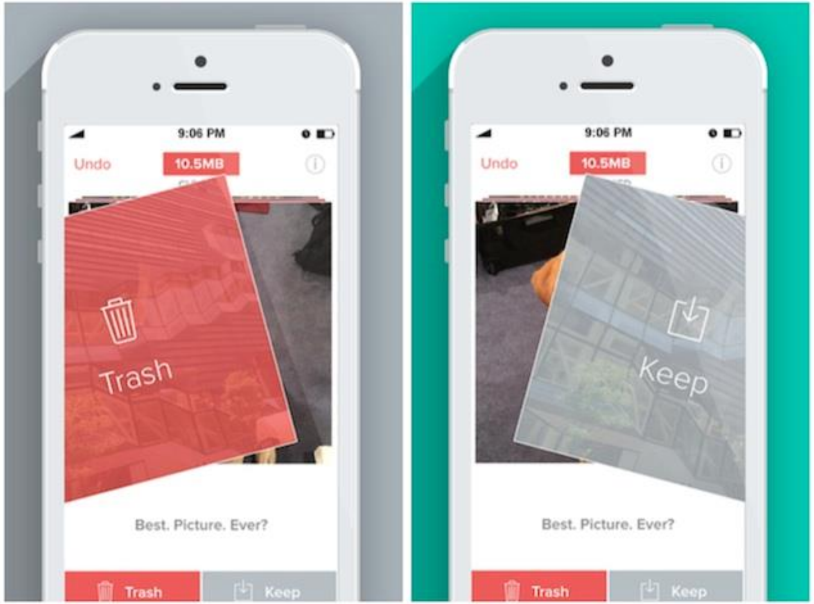


Explicit or Implicit?

Ratings: *Explicit*



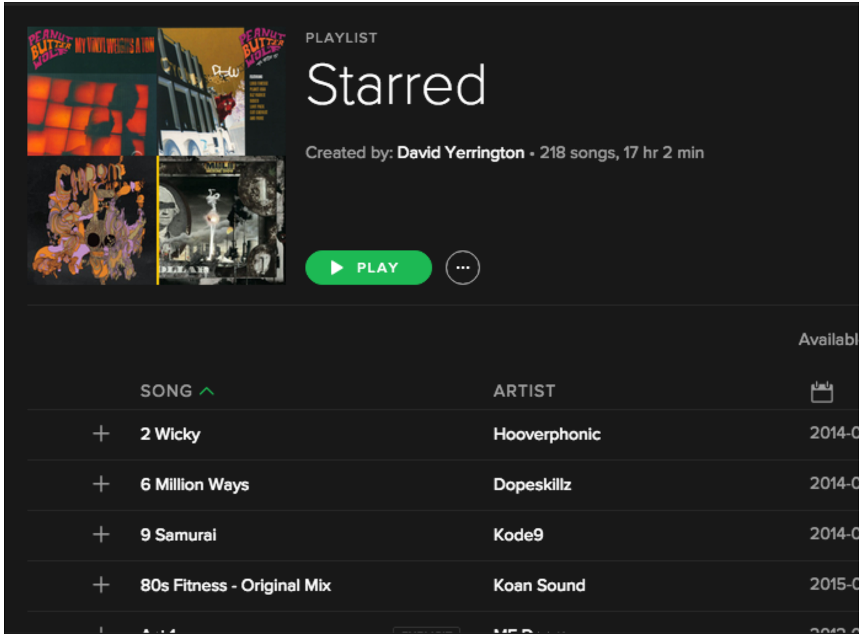
Explicit or Implicit?



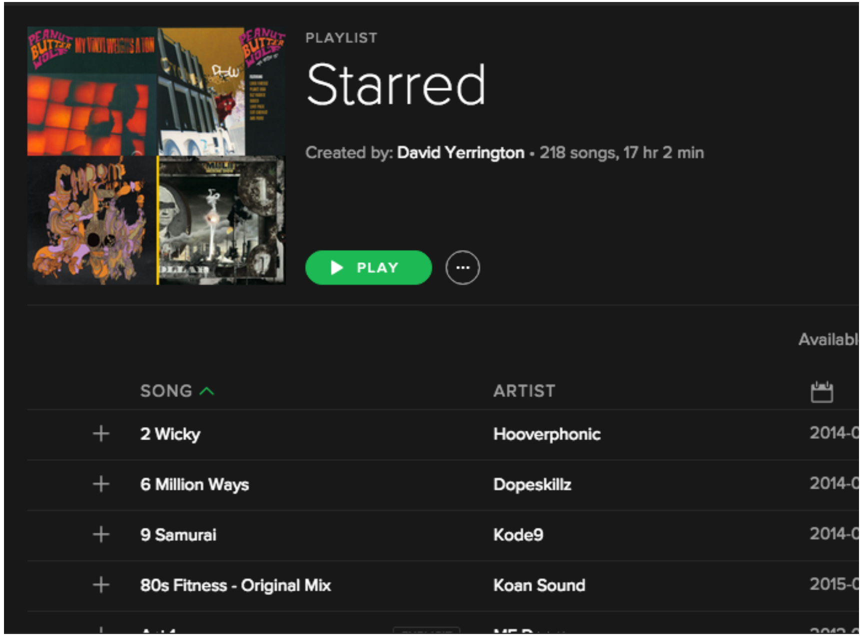
Explicit or Implicit?

Swipes: *Explicit*

EXAMPLES – TYPES OF DATA

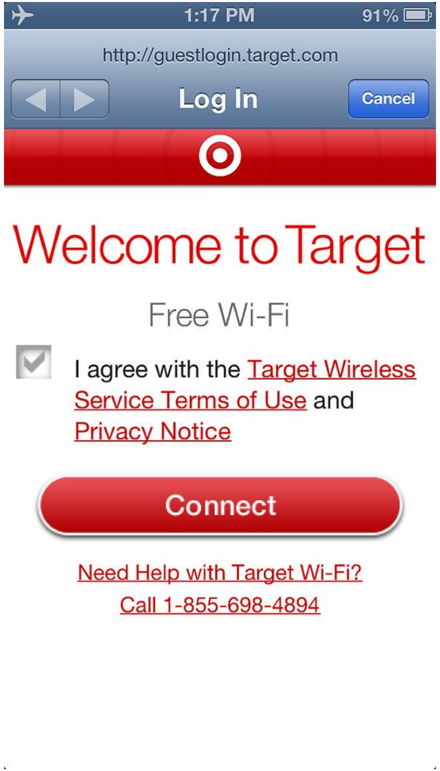


Explicit or Implicit?

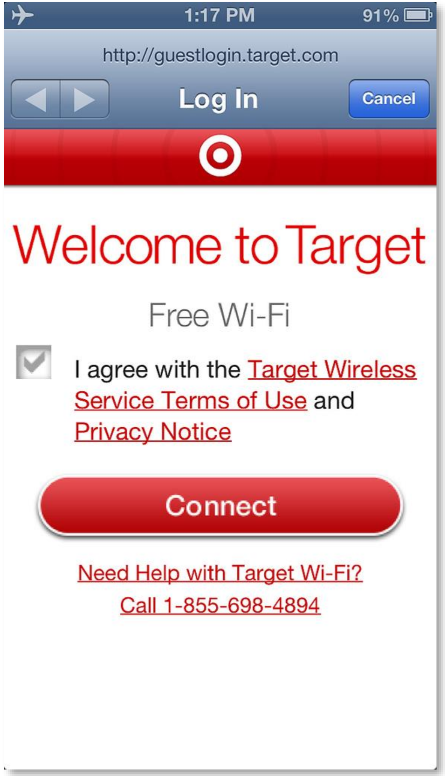


Explicit or Implicit?

Both!



Explicit or Implicit?



Explicit or Implicit?

Wifi logs: *Implicit*!

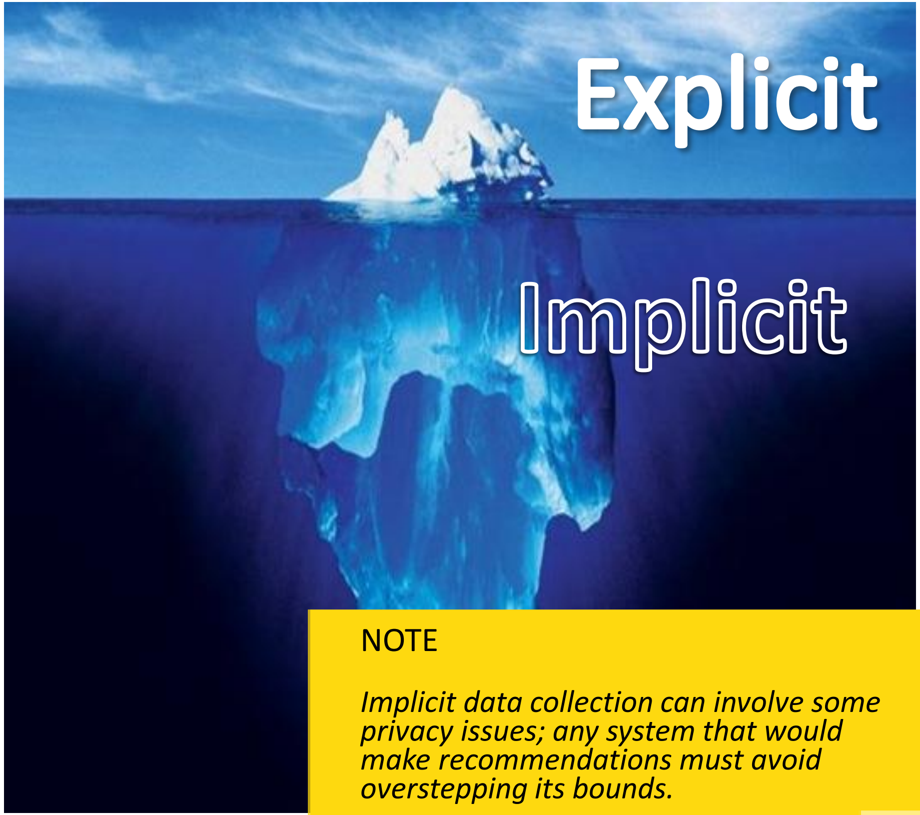
EXAMPLES – TYPES OF DATA

Explicit

- Ie: Ratings, surveys, reviews
- Easy to interpret
- Expensive

Implicit

- Ie: Activity logs, clicks, impressions
- Hard to interpret
- Cheap



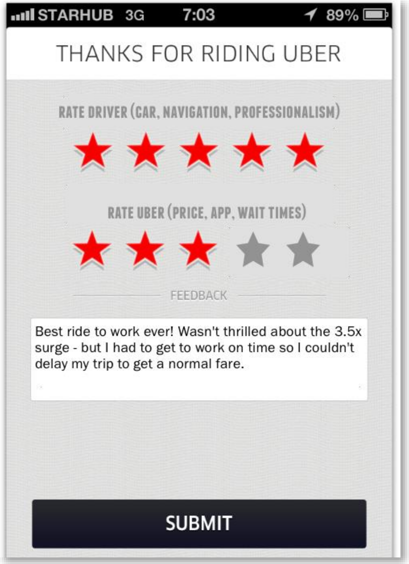
NOTE

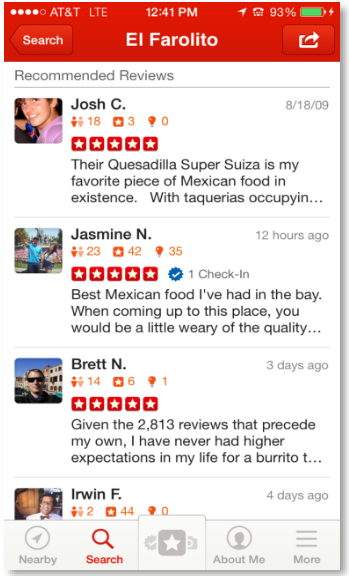
Implicit data collection can involve some privacy issues; any system that would make recommendations must avoid overstepping its bounds.

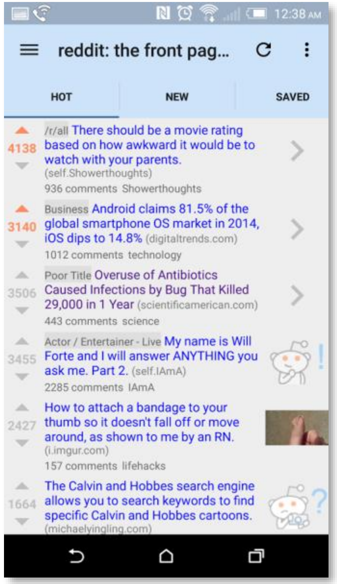
IA. EXPLICIT AND IMPLICIT FEEDBACK

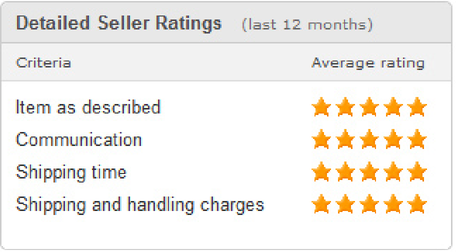
EXAMPLES – TYPES OF DATA

Ratings, Votes, Reviews









Uber Yelp Reddit Ebay

Explicit Feedback

- Frequently in the form of ratings
- Granularly represents preferences
- Requires extra effort from the user

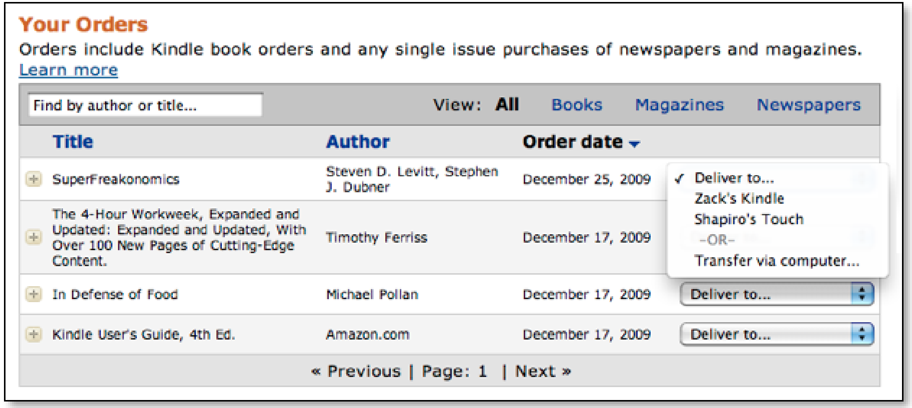
Explicit Feedback Questions

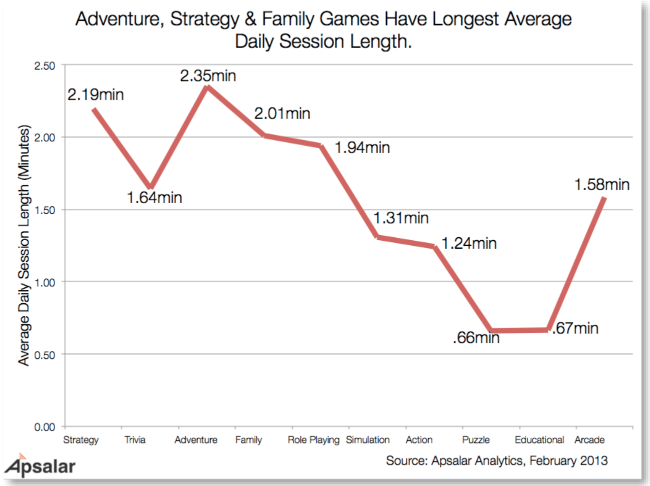
- What does a rating mean?
- Do user preferences change?
- Is what is known about the data accurate?
  - Is what is collected reflect a preference at all?
  - Is it representative to the goal or only reflective of a singular characteristic?

Explicit Feedback - Considerations

- Consistent scale for all ratings
- Can ratings be skewed by self/selection-bias
- Consider the ephemeral nature of preferences
- When the data was collected
  - Before or after experience
- Context of presentation

Implicit Examples

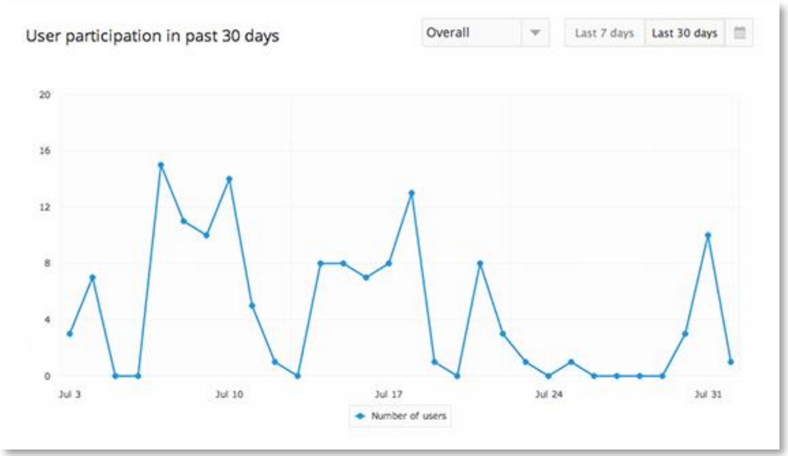




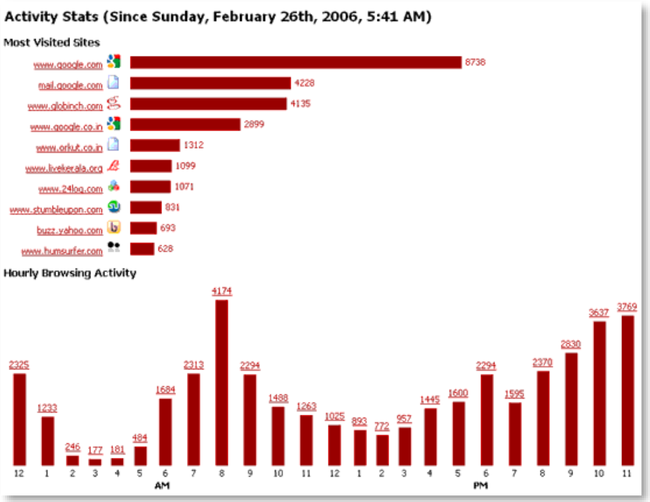
Order History

Session Length

Implicit Examples



Engagement Metrics



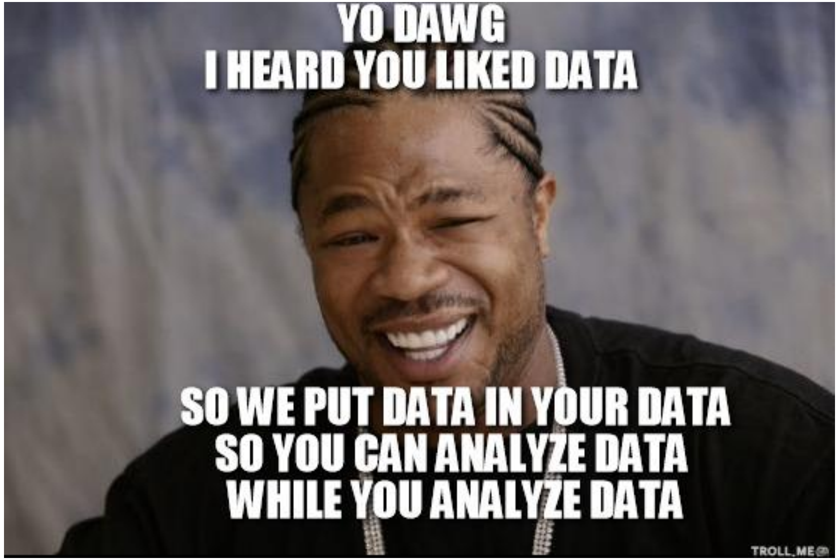
Session Length

Implicit Feedback

It’s still possible to make recommendations when no rating data is explicitly collected from a user.

The goal is to convert user behavior into user preferences, but it entails one challenge: How exactly does one infer preference based on actions in a system? This can be a difficult question to answer.

Implicit Feedback



There's tons of it!

Implicit feedback is everywhere.

- Email impressions
- Email click-throughs
- Conversions
- Demographic
- Session lengths
- Login attempts
- Track plays
- Money spent
- Ad impressions
- Ad clicks
- Ad click-purchase
- Web “click depth”
- # of swipes
- Profile views
- Message initiations
- Poll Votes
- Friend / unfriend
- Follow / unfollow
- *Like
- Post text
- Image EXIF
- Friends in common
- Message text
- Food purchases
- Geospatial data
- Store cameras
- Wifi logins / MAC
- Time series
- Objects in photos
- Driving record
- Credit history
- Topics most read

Implicit feedback is valuable depending on how you look at it.

?

Implicit Feedback Caveats

(ie: Users don’t tell you what you want to know.)

- Preferences can be vague
- You may need to process tons of data to get what you want
- Analysis can be complicated / meaning hard to find
- Identities can be indistinguishable
- Users don’t tell you what you want to know
- Easy to project bias onto data
- Positive / negative experience hard to assess

Implicit Feedback General Advice: Question Everything.

- Can a preference actually be observed?
- Is the lack of data actually a negative preference?
- Is there enough data to describe feedback or only a portion of it?
- Is the data scaled properly?
- Are there hidden correlations?
- Are there contradictory patterns?
- What’s missing?
- Can new features be created?

Implicit + Explicit Feedback: Work together

If a user rates an item, can you can use implicit feedback to validate credibility

- Did they read the article?
- Do they own the item?
- Did they rate before or after experience?
- Do other users mention them?
- Does user tend to rate high or low?
- How likely was the rating automated?

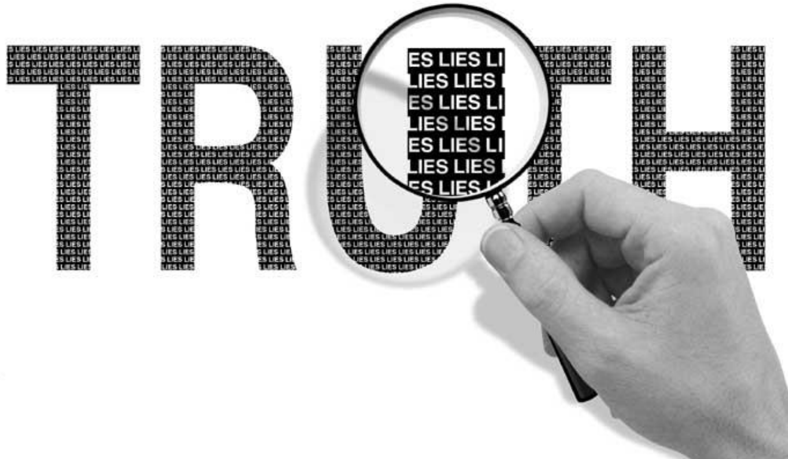
Use implicit data to understand the context and characteristics of a rating.

- Does time of day affect rating?
- Which kinds of reviews do they typically write?
- Are the reviews positive or negative?
- Do other users like their reviews?

Implicit + Explicit Feedback: Final Caveat

Take care when if creating explicit data from implicit data.

- Does the set of actions reflect a preference?
- Does the scale make sense?
- Is the outcome prediction (ratings) or recommendation?



Explicit

- Higher value with respect to preferences
- Usually collected as a “rating”
- Collection is responsibility of user
- More direct evaluation of items

Implicit

- Easy to collect in large quantities
- More difficult to work with
- Assumes nothing about the user (could be anyone!)
- Goal is to convert into preferences

IB. GENERAL DESIGN

There are two general approaches to the design:

There are many approaches to the design, but these are commonly modeled techniques:

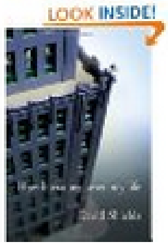
In **content-based filtering**, items are mapped into a feature space, and recommendations depend on *item characteristics*.

In contrast, an important assumption underlying all of **collaborative filtering**, is: *users who have similar preferences in the past are likely to have similar preferences in the future.*

EXAMPLES – AMAZON CONTENT-BASED

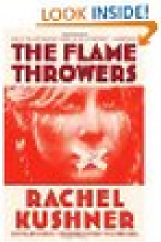
Customers Who Bought This Item Also Bought

Pitch Dark (NYRB Classics)
› Renata Adler
Paperback
$11.54



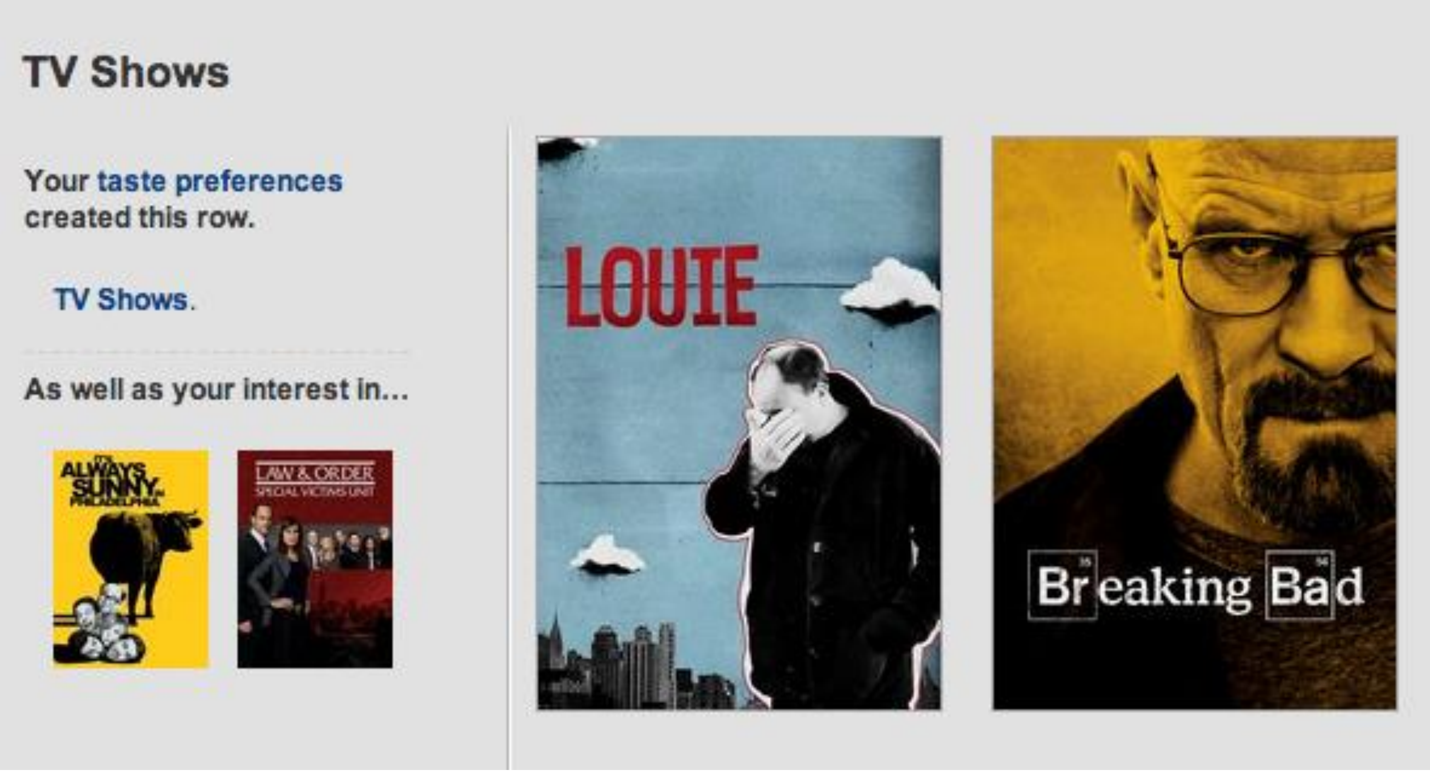
How Literature Saved My Life
› David Shields
(60)
Hardcover
$18.08

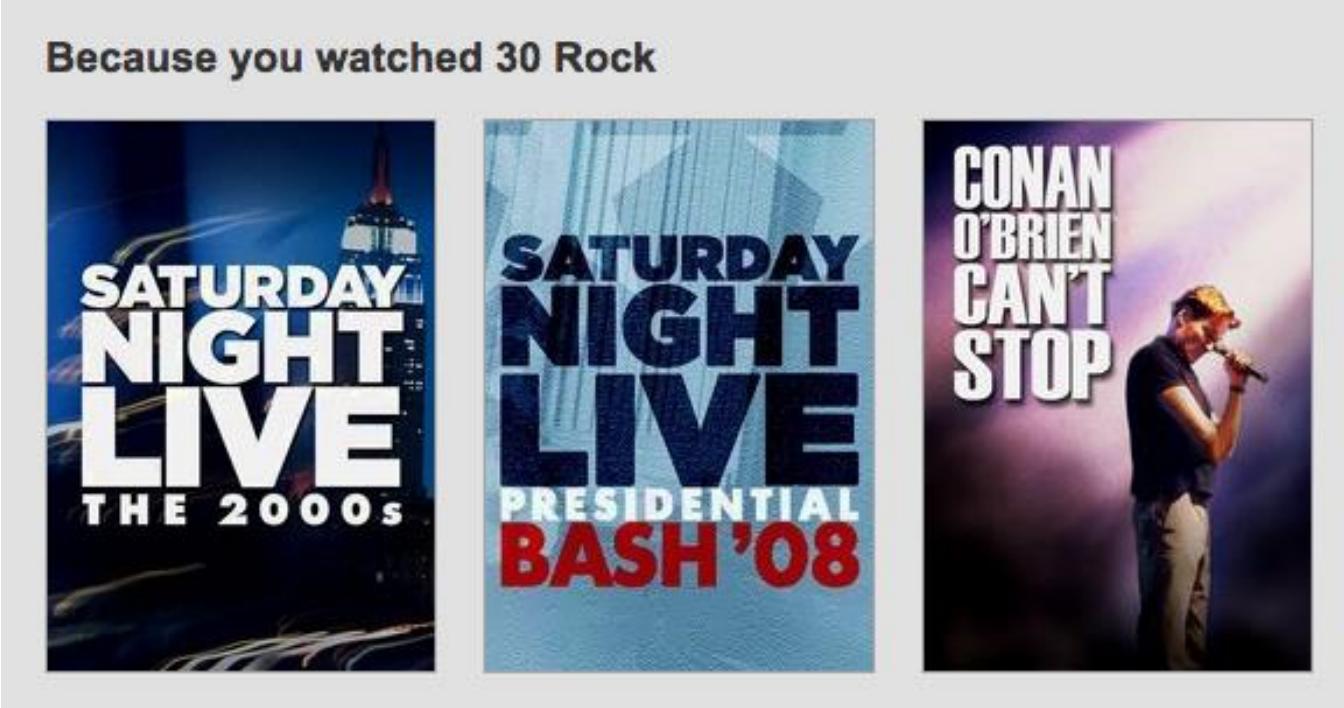
Bleeding Edge
Thomas Pynchon
Hardcover
$18.05



The Flamethrowers: A Novel
› Rachel Kushner
(17)
Hardcover
$15.79

EXAMPLES – NETFLIX




Because you watched 30 Rock
SATURDAY NIGHT LIVE
THE 2000s
SATURDAY NIGHT LIVE
PRESIDENTIAL
BASH '08
CONAN O'BRIEN CAN'T STOP

EXAMPLES – YOUTUBE

Recommended for you because you watched
Sugar Minott - Oh Mr Dc (Studio One)

Mikey Dread - Roots and Culture

by klaxonklaxon · 1,164,133 views

Lyrics:
Now here comes a special request
To each and everyone

Recommended for you because you watched
Thelonious Monk Quartet - Monk In Denmark

Bill Evans Portrait in Jazz (Full Album)

by hansgy1 · 854,086 views

Bill Evans Portrait in Jazz 1960
1. Come Rain or Come Shine - 3.19 (0:00)
2. Autumn Leaves - 5.23 (3:24)

Recommended for you because you watched
Bob Marley One Drop

Bob Marley - She's gone

by Dionysios29 · 1,058,704 views

This is one of the eleven songs of album Kaya that Bob Marley and The Wailers creative in 1978.
Lyrics:

How can we find good recommendations?

- Manual Curation
- Manually Tag Attributes
- Audio Content, Metadata, Text Analysis
- Collaborative Filtering

content-based filtering

EXAMPLES – NYTIMES.COM

MOST E-MAILED | RECOMMENDED FOR YOU

1. **How Big Data Is Playing Recruiter for Specialized Workers**
2. SLIPSTREAM
   When Your Data Wanders to Places You've Never Been
3. MOTHERLODE
   The Play Date Gun Debate
4. **For Indonesian Atheists, a Community of Support Amid Constant Fear**
5. **Justice Breyer Has Shoulder Surgery**
6. BILL KELLER
   Erasing History

8. How do you determine my Most Read Topics?

Back to top ▲

Each NYTimes.com article is assigned topic tags that reflect the content of the article. As you read articles, we use these tags to determine your most-read topics.

To search for additional articles on one of your most-read topics, click that topic on your personalized Recommendations page. To learn more about topic tags, visit Times Topics.

NOTE

Collaborative or Content based?

EXAMPLES – NYTIMES.COM

8. How do you determine my Most Read Topics?

Back to top ▲

Each NYTimes.com article is assigned topic tags that reflect the content of the article. As you read articles, we use these tags to determine your most-read topics.

To search for additional articles on one of your most-read topics, click that topic on your personalized Recommendations page. To learn more about topic tags, visit Times Topics.

NOTE

Collaborative or Content based?

CONTENT BASED ☺

II. CONTENT-BASED FILTERING

Content-based filtering begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.

Content-based filtering begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.

Item vectors measure the degree to which the item is described by each feature, and ***user vectors*** measure a user's preferences for each feature.

Content-based filtering begins by mapping each item into a feature space. Both users and items are represented by vectors in this space.

Item vectors measure the degree to which the item is described by each feature, and ***user vectors*** measure a user's preferences for each feature.

Ratings are generated by taking **dot products** of user & item vectors.

features = (big box office, aimed at kids, famous actors)

Items (movies):

Finding Nemo = (5, 5, 2)

Mission Impossible = (3, -5, 5)

Jiro Dreams of Sushi = (-4, -5, -5)

features = (big box office, aimed at kids, famous actors)

Items (movies):

Finding Nemo = (5, 5, 2)
Mission Impossible = (3, -5, 5)
Jiro Dreams of Sushi = (-4, -5, -5)

Users:

Alice = (-3, 2, -2)
Bob = (4, -3, 5)

features = (big box office, aimed at kids, famous actors)

Items (movies):

Finding Nemo = (5, 5, 2)

Mission Impossible = (3, -5, 5)

Jiro Dreams of Sushi = (-4, -5, -5)

Prediction (for Alice)

5*-3 + 5*2 + 2*-2 = -9

User:

Alice = (-3, 2, -2)

features = (big box office, aimed at kids, famous actors)

Items (movies):	**Prediction (for Alice)**	
Finding Nemo = (5, 5, 2)	5*-3 + 5*2 + 2*-2	= -9
Mission Impossible = (3, -5, 5)	3*-3 + -5*2 + 5*-2	= -29
Jiro Dreams of Sushi = (-4, -5, -5)		

User:

Alice = (-3, 2, -2)

features = (big box office, aimed at kids, famous actors)

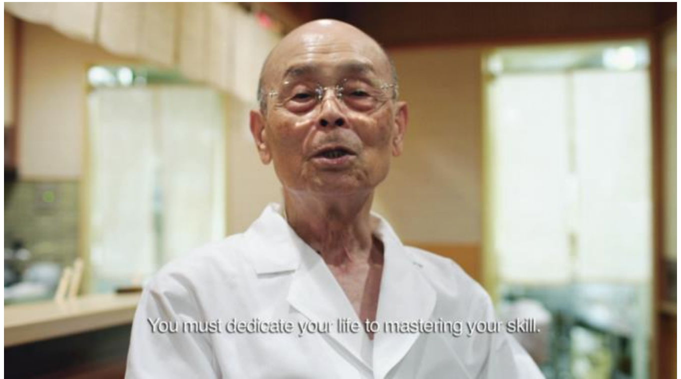
Items (movies):	**Prediction (for Alice)**	
Finding Nemo = (5, 5, 2)	5*-3 + 5*2 + 2*-2	= -9
Mission Impossible = (3, -5, 5)	3*-3 + -5*2 + 5*-2	= -29
Jiro Dreams of Sushi = (-4, -5, -5)	-4*-3 + -5*2 + -5*-2	= +12

User:

Alice = (-3, 2, -2)

features = (big box office, aimed at kids, famous actors)

Items (movies):	**Prediction (for Alice)**	
Finding Nemo = (5, 5, 2)	5*-3 + 5*2 + 2*-2	= -9
Mission Impossible = (3, -5, 5)	3*-3 + -5*2 + 5*-2	= -29
Jiro Dreams of Sushi = (-4, -5, -5)	-4*-3 + -5*2 + -5*-2	= +12



User:

Alice = (-3, 2, -2)

features = (big box office, aimed at kids, famous actors)

Items (movies):

Finding Nemo = (5, 5, 2)

Mission Impossible = (3, -5, 5)

Jiro Dreams of Sushi = (-4, -5, -5)

Prediction (for Bob)

User:

Bob = (4, -3, 5)

features = (big box office, aimed at kids, famous actors)

Items (movies):	**Prediction (for Bob)**	
Finding Nemo = (5, 5, 2)	5*4 + 5*-3 + 2*5	= +15
Mission Impossible = (3, -5, 5)	3*4 + -5*-3 + 5*5	= +52
Jiro Dreams of Sushi = (-4, -5, -5)	-4*4 + -5*-3 + -5*5	= -26

User:

Bob = (4, -3, 5)

features = (big box office, aimed at kids, famous actors)

Items (movies):	**Prediction (for Bob)**
Finding Nemo = (5, 5, 2)	5*4 + 5*-3 + 2*5 = +15
Mission Impossible = (3, -5, 5)	3*4 + -5*-3 + 5*5 = +52
Jiro Dreams of Sushi = (-4, -5, -5)	-4*4 + -5*-3 + -5*5 = -26

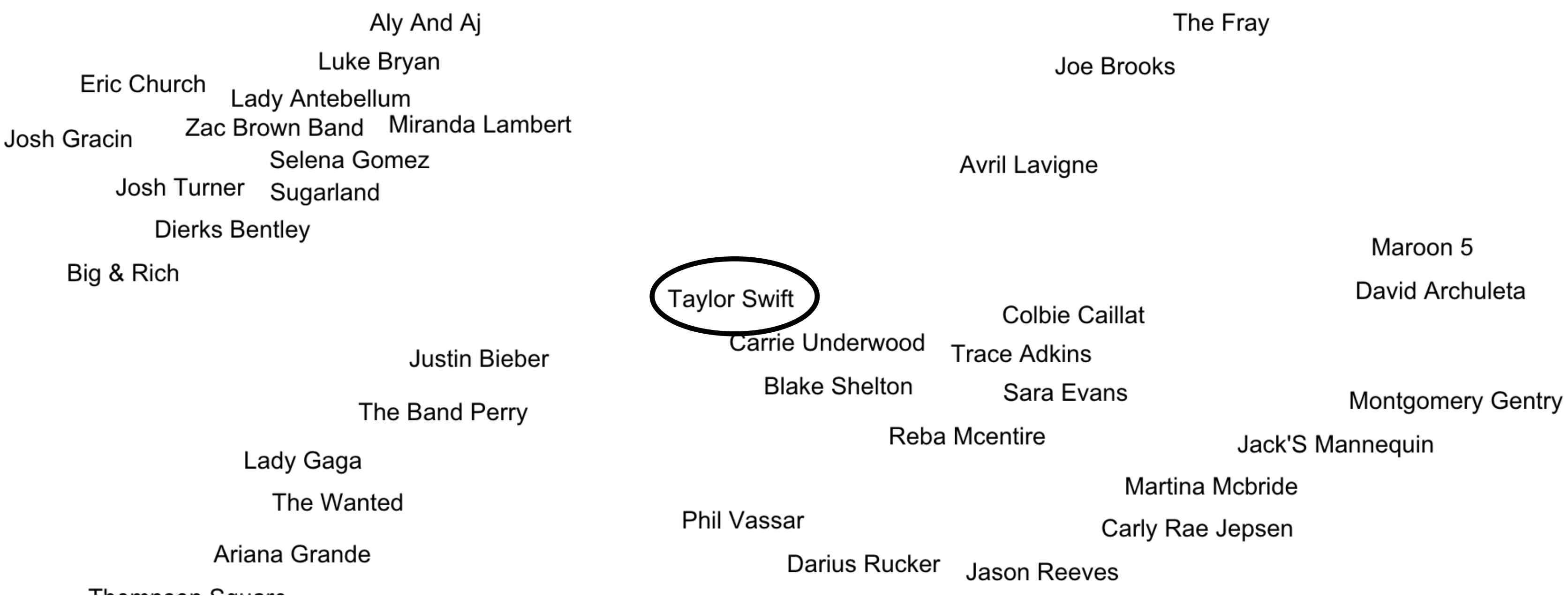
User:

Bob = (4, -3, 5)

One notable example of content-based filtering is Pandora, which maps songs into a feature space using features (or “genes”) designed by the Music Genome Project.

Using song vectors that depend on these features, Pandora can create a station with music having similar properties to a song the user selects.

VISUALIZATION OF SIMILAR ARTISTS



http://www.music-map.com/

Content-based filtering has some difficulties:

Content-based filtering has some difficulties:

- Must map items into a feature space (usually by hand!)

- Recommendations are limited in scope (items must be similar to each other)

- Hard to create cross-content recommendations (eg books/music films…this would require comparing elements from different feature spaces!)

III. COLLABORATIVE FILTERING

Collaborative filtering refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are *only* interested in the existing user-item ratings themselves.

Collaborative filtering refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are *only* interested in the existing user-item ratings themselves.

In this case, our dataset is a *ratings matrix* whose columns correspond to items, and whose rows correspond to users.

Collaborative filtering refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are *only* interested in the existing user-item ratings themselves.

In this case, our dataset is a *ratings matrix* who columns correspond to items, and whose rows correspond to users.

NOTE

The idea here is that users get value from recommendations based on other users with similar *tastes*.

18,000 movies

480,000 users

x	1	1	x	...	x
x	x	x	5	...	x
x	x	3	x	...	x
x	4	3	x	...	2
...	x	x	x	...	x
x	5	x	1	...	x
x	x	3	3	...	x
x	1	x	x	...	2

NOTE

This matrix will always be *sparse*!

source: http://www.eecs.berkeley.edu/~zhanghao/main/publications/subfolder/netflix.png

Main difference between content and collaborative filtering:

Content Based:

maps items and users into a feature space

Collaborative:

relies on previous user-item ratings

We will look at collaborative filtering in a user-user sense.

We will look at collaborative filtering in a user-user sense.

We will take a given user, and find the K most similar users, and then recommend brands from the similar users!

We will look at collaborative filtering in a user-user sense.

We will take a given user, and find the K most similar users, and then recommend brands from the similar users!

NOTE

Sound familiar? It's similar to KNN!


Customers Who Bought This Item Also Bought
LOOK INSIDE!
Pitch Dark (NYRB Classics)
Renata Adler
Paperback
$11.54
LOOK INSIDE!
How Literature Saved My Life
David Shields
(60)
Hardcover
$18.08
No image available
Bleeding Edge
Thomas Pynchon
Hardcover
$18.05
LOOK INSIDE!
THE FLAME THROWERS
RACHEL KUSHNER
The Flamethrowers: A Novel
Rachel Kushner
(17)
Hardcover
$15.79

The system cannot draw inferences because it hasn’t gathered enough information yet.

The cold start problem arises because we’ve been relying only on ratings data, or on explicit feedback from users.

The cold start problem arises because we’ve been relying only on ratings data, or on explicit feedback from users.

Until users rate several items, we don’t know anything about their preferences!

The cold start problem arises because we’ve been relying only on ratings data, or on explicit feedback from users.

Until users rate several items, we don’t know anything about their preferences!

We can get around this by enhancing our recommendations using implicit feedback, which may include things like item browsing behavior, search patterns, purchase history, etc.

While explicit feedback (ratings, likes, purchases) leads to high quality ratings, the data is sparse and cold starts are problematic.

While explicit feedback (ratings, likes, purchases) leads to high quality ratings, the data is sparse and cold starts are problematic.

Meanwhile implicit feedback (browsing behavior, etc.) leads to less accurate ratings, but the data is much more dense (and less invasive to collect).

How do we define "similarity" of users?

This is required if we want to do user-based collaborative filtering

MATH ALERT!!

How do we define "similarity" of users?

Jaccard Similarity:

Defines similarity between two sets of objects

How do we define “similarity” of users?

Jaccard Similarity:

Defines similarity between two sets of objects

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

How do we define "similarity" of users?

Jaccard Similarity:

Defines similarity between two sets of objects

Number of similar elements

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Number of distinct elements

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

J ({1, 2, 3}, {2,3,4}) = {2, 3}   2

------   = ----

{1, 2, 3, 4}   4

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Exercise:

User one: {“Target”, “Banana Republic”, “Old Navy”}
User two: {“Banana Republic”, “Gap”, “Kohl’s”}

JS (User one, User two) =

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Exercise:

User one: {“Target”, “Banana Republic”, “Old Navy”}
User two: {“Banana Republic”, “Gap”, “Kohl’s”}

JS (User one, User two) = 1 / 5 **= .2**

PYTHON ALGORITHM STEPS

1. Get list of known users in a dictionary where the key is the user ID, an the value is a list of brands they like

    Example: { ‘83065’ : ["Kohl's", 'Target’] }

2. For a given user, we will calculate their closeness to every user in csv
3. We will choose the K most similar users
4. Recommend brands liked by similar users

PYTHON ALGORITHM STEPS

1. Get list of known users in a dictionary where the key is the user ID, an the value is a list of brands they like

   Example: { ‘83065’ : ["Kohl's", 'Target’] }

2. For a given user, we will calculate their closeness to every user in csv
3. We will choose the K most similar users
4. Recommend brands liked by similar users

Consider this a kind of KNN but instead of Euclidean Distance, we are using the Jaccard Similarity

IV. THE NETFLIX PRIZE

The Netflix prize was a competition to see if anyone could make a 10% improvement to Netflix's recommendation system (accuracy measured by RMSE).

The grand prize was $1m dollars

The Netflix prize was a competition to see if anyone could make a 10% improvement to Netflix's recommendation system (accuracy measured by RMSE).

The grand prize was $1m dollars

The ratings matrix contained >100mm numerical entries (1-5 stars) from ~500k users across ~17k movies. The data was split into train/quiz/test sets to prevent overfitting on the test data by answer submission (this was a clever idea!)

The competition began in 2006, and the grand prize was eventually awarded in 2009. The winning entry was a stacked ensemble of 100's of models (including neighborhood & matrix factorization models) that were blended using boosted decision trees.

Ultimately, the competition ended in a photo finish. The winning strategy came down to last-minute team mergers & creative blending schemes to shave 3rd & 4th decimals off RMSE (concerns that would not be important in practice).

The competition began in 2006, and the grand prize was eventually awarded in 2009. The winning entry was a stacked ensemble of 100’s of models (including neighborhood & matrix factorization models) that were blended using boosted decision trees.

Ultimately, the competition ended in a photo finish. The winning strategy came down to last-minute team mergers & creative blending schemes to shave 3rd & 4th decimals off RMSE (concerns that would not be important in practice).

The competition did much to spur interest and research advances in recsys technology, and the prize money was donated to charity.

Though they adopted some of the modeling techniques that emerged from the competition, Netflix never actually implemented the prizewinning solution.

Why do you think that’s true?