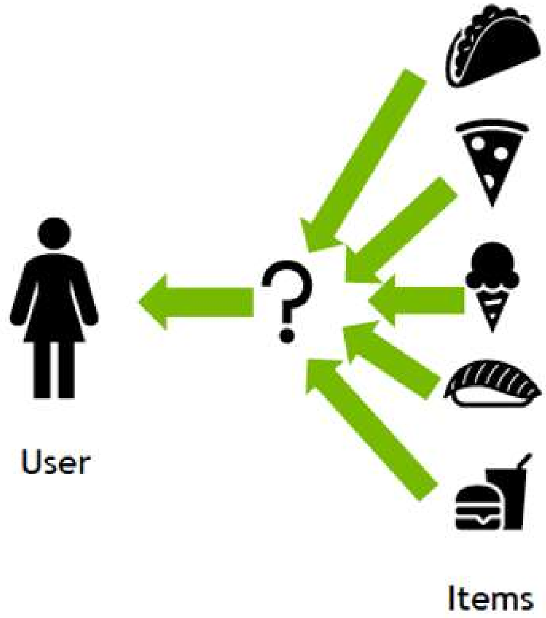
Graduate Certificate in Big Data Analytics

Recommender Systems

Hybrid Systems, Commercial Platforms, Closing Remarks



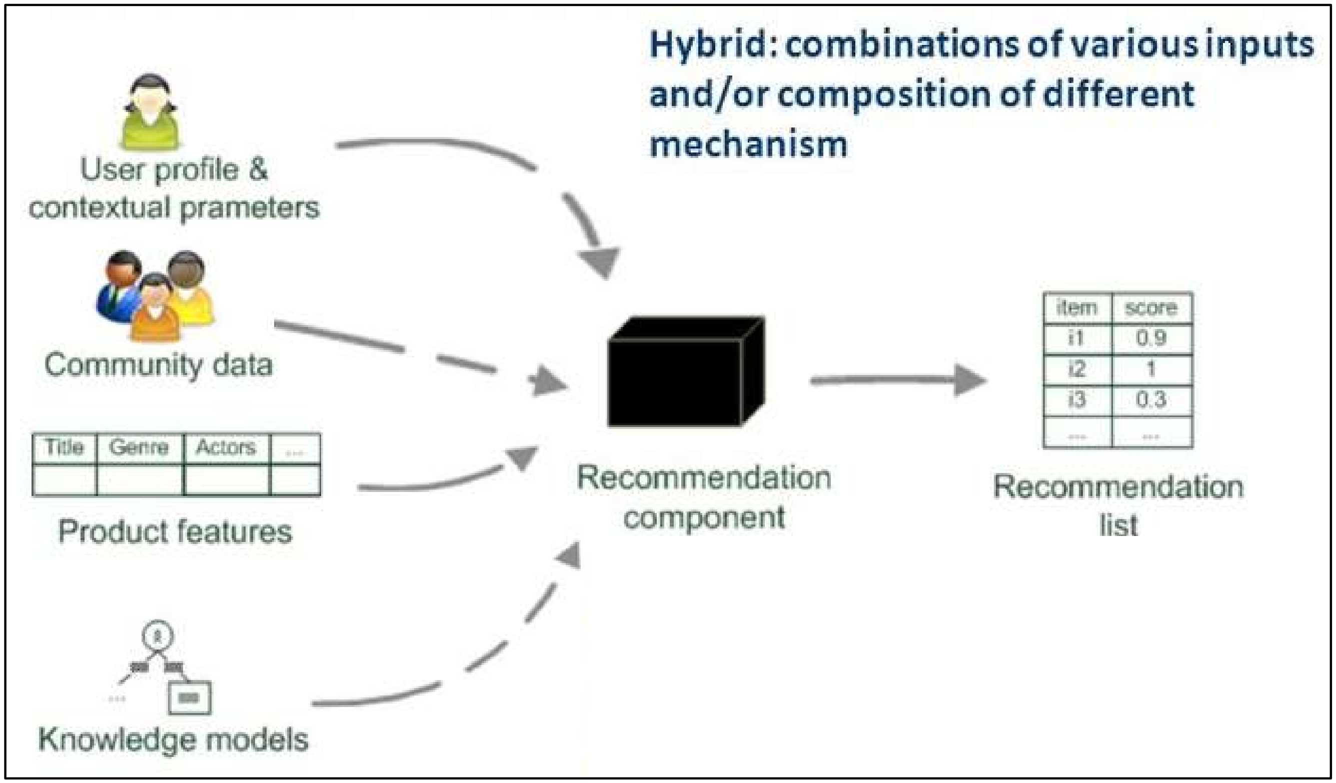
Dr. Barry Shepherd
Institute of Systems Science
National University of Singapore
Email: barryshepherd@nus.edu.sg

Hybrid Recommender Systems

- Often combining many different systems can generate better results
- E.g. Pandora Recommendation Engine
  - "The recommender uses about 70 different algorithms: 10 analyze content, 40 process collective intelligence, and then another 30 do personalized filtering.

    Celma said, "This is challenging from an engineering point of view. We have the goal that when you thumb down a song, the recommendation for the next song occurs in less than 100 milliseconds. It is hard to do this in a way that scales across all users."
- E.g. Netflix Prize Winner
  - The winner of the Netflix prize was a combination of 107 algorithms

Hybrid Recommender Systems

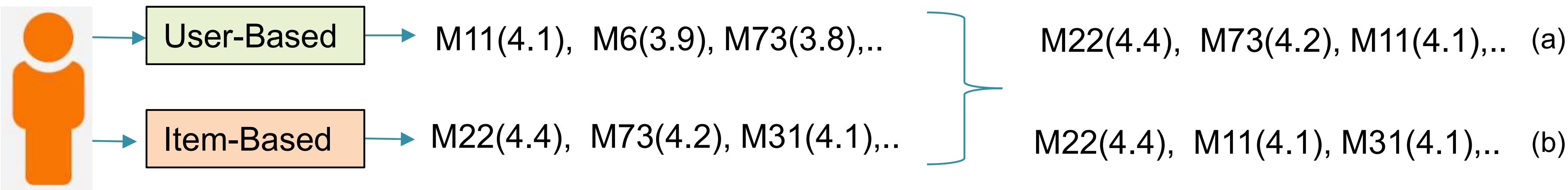


- **Mixed:** Present all recommender results together
- **Weighted:** Numerically combine the scores
- **Switching:** Switch between different recommendations according to user profile & context
- **Cascade:** Assign recommenders a priority, the low priority ones break ties between the higher ones
- **Feature Combination:** Combine features from different sources and input to a single recommender
- **Feature Augmentation:** One recommender generates features that form part of the input to another
- **Meta-level:** One recommendation technique is applied and produces a model, this is then the input to the next technique.

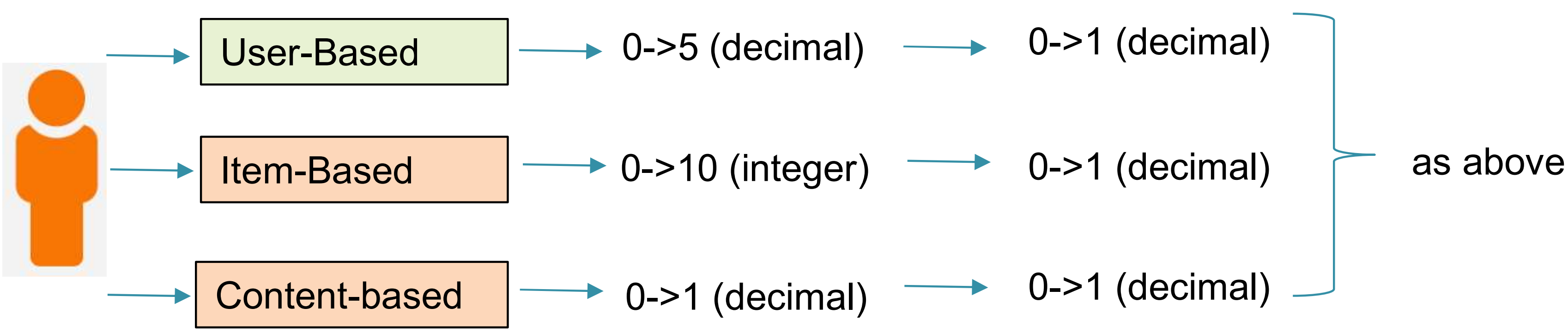
Hybridization techniques from Wikipedia

Numerical Combination of Scores

- If recommender outputs have the same range (e.g. predicted rating 1->5) then they can be numerically combined, e.g.
  a) Pick the topN items with the highest predicted ratings, or
  b) Average the predicted ratings for each item, pick the topN highest



- If the outputs have different ranges then can normalise all to be the same range



Other Combination Methods

- Pick the most frequent items in the topN , with tie break:

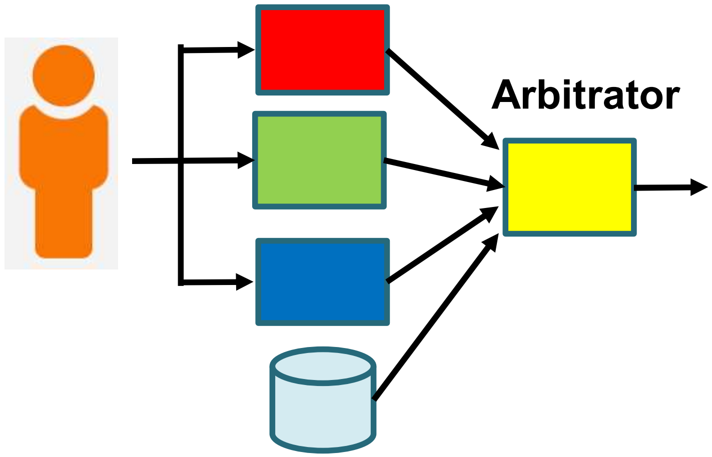
User-based		Item-based		User-based (implicit)		Combined
M11 (4.1)	+	M22 (4.4)	+	M31 (0.91)	⇨	M73 (3 times)
M6 (3.9)		M73 (4.2)		M11 (0.87)		M11 (2 times)
M73 (3.8)		M31 (4.1)		M82 (0.86)		M6 (2 times)
M19 (3.7)		M6 (4.0)		M73 (0.81)		M31 (2 times)

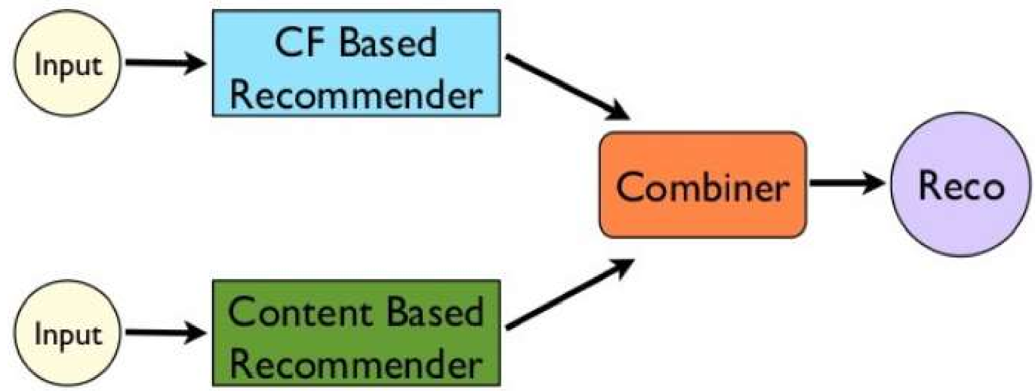
  – For small N and many items its possible (likely) than the top items are all different hence one solution is to make N larger and then select the best K items (K < N)

- Or, pick a selection from the topN of each recommender:

User-based		Item-based		User-based (implicit)		Combined (top 2 from each)
M11 (4.1)	+	M22 (4.4)	+	M31 (0.91)	⇨	M11
M6 (3.9)		M73 (4.2)		M11 (0.87)		M6
M73 (3.8)		M31 (4.1)		M82 (0.86)		M22
M19 (3.7)		M6 (4.0)		M73 (0.81)		M73
						M31
						M82

Weighted Combination





- Feature Weighted Stacking
  - Many different recommenders (models) are stacked up
  - Another model is used to weight their votes
  - Can select weights manually via experimentation
  - Or, use a ML algorithm to learn the weights
    - E.g. linear regression
    - Training signal could be rating prediction error

Weighted Combination of Similarities

- Combining User Ratings with Content-Based Filtering (example)*

User Profile

~ characteristics of the items viewed (& liked)

<u>genre</u>
(romance, scifi, action, comedy,..)
0.3 0.4 0.2 0.1
Normalised counts of items viewed

<u>synopsis</u>
(word1, word2, word3,..)
0.023 0.12 0.02
TF-IDF's (across all viewed items)

<u>director</u>
<u>actress</u>
<u>actor</u>

Cluster to get user cliques/groups (good for cold start)

User Ratings

~ ratings given to viewed items

$$\text{Sim}_{\text{userJ,userK}} = \text{sim}_{\text{userJ,userK}} * (1\text{-c}) + \text{sim}_{\text{groupJ,groupK}} * \text{c}$$

C ~ 0.3 was found to give best results

Collaborative Filtering

A new approach for combining content-based and collaborative filters
Byeong Man Kim, Qing Li, Chang Seok Park, Si Gwan Kim, Ju Yeon Kim
Journal of Intelligent Information Systems
July 2006, Volume 27, Issue 1, pp 79–91 | Cite as

*https://link.springer.com/article/10.1007/s10844-006-8771-2

Feature Combination

- Combine the features together into one large user-record and input into a single recommender
- E.g.

user1: (m1, m2, m3, m4, m5, …) + (romance, scifi, action, comedy,..) + (word1, word2, word3,..)

3 4 - 2 - … 0.3 0.4 0.2 0.1 0.023 0.12 0.02

Normalised ranges Normalised ranges Normalised ranges

User-user or item-item similarity

Cascading Example

- Assign recommenders a priority, the low priority ones break ties between the higher ones
- E.g.

CF (Item-Based) → Tie break → CF (User-Based) → Tie break → Content-Based

CF (Item-Based) ↓ Recommendations

CF (User-Based) ↓ Recommendations

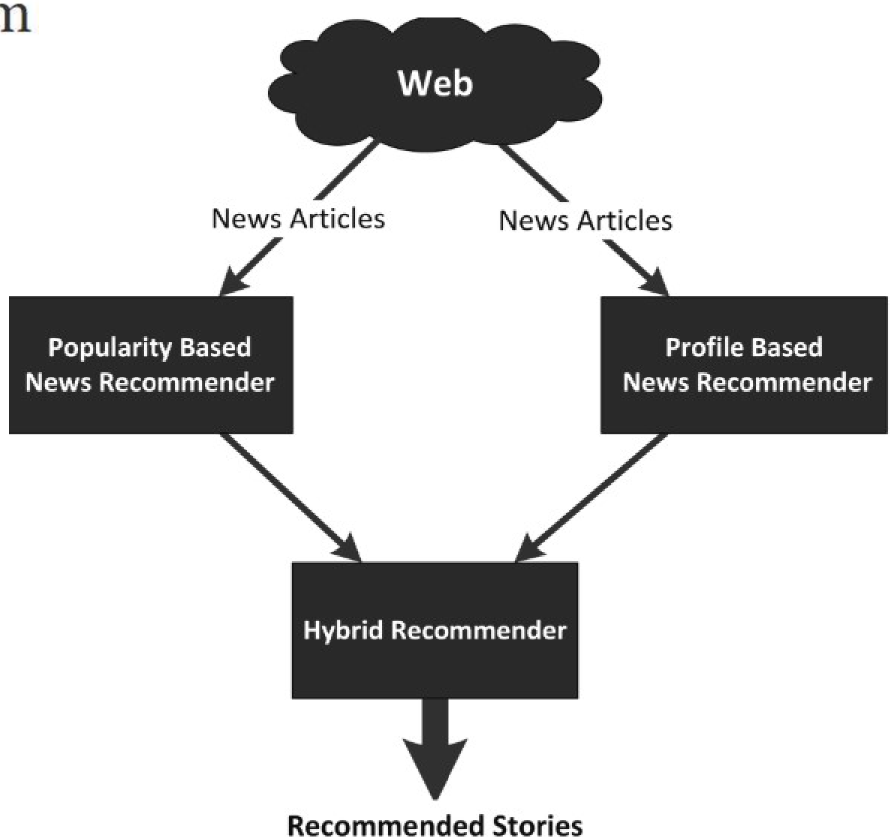
Content-Based ↓ Recommendations

Example: News Recommendation (1)

Incorporating popularity in a personalized news recommender system

Nirmal Jonnalagedda, Susan Gauch, Kevin Labille, Sultan Alfarhood

https://peerj.com/articles/cs-63/

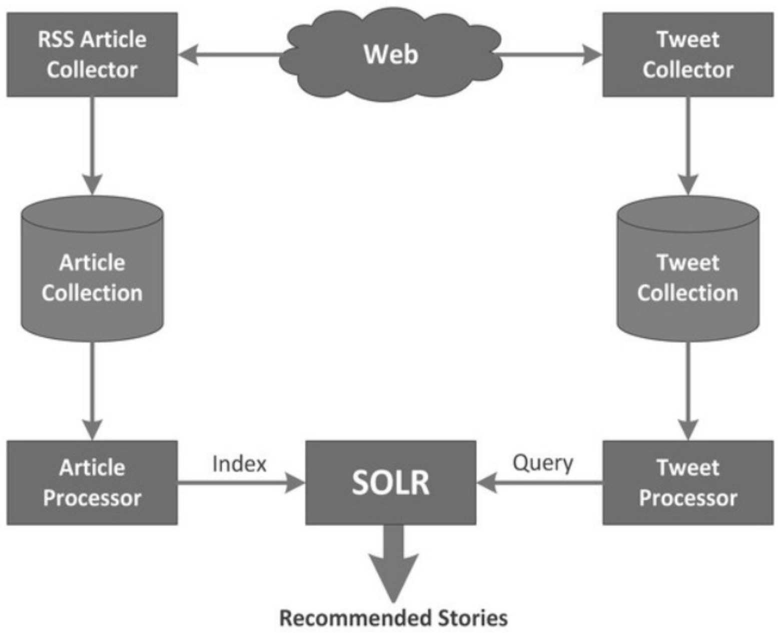


$$Hybrid\_Wt2_{ij} = \alpha * Popularity\_Wt_j + (1-\alpha) * Personal\_Wt_{ij}$$

Example: News Recommendation (2)

Popularity-Based Recommender
Popularity score is based on #tweets that get mapped to an article over a time period.

$$Popularity\_Wt_i = \sum_{t \in T} cosineSimilarity(Article_i, Tweet_t)$$



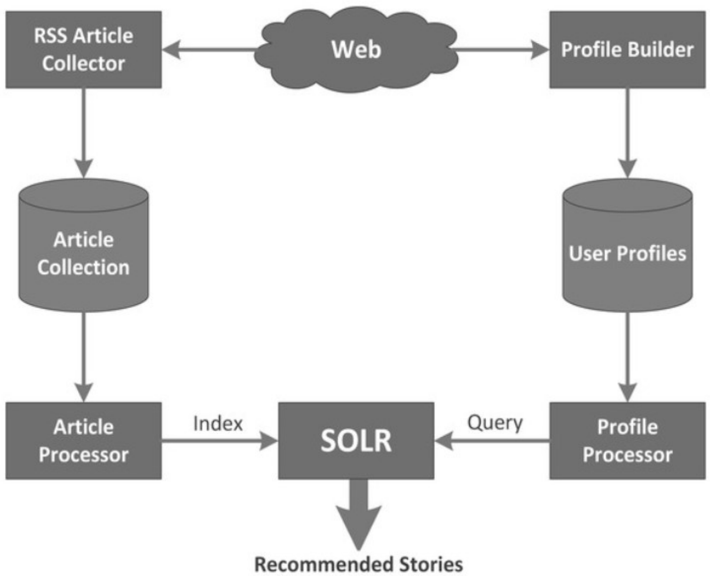
	Article B1	Article B2	Article B3	Article E4	Article S5	Article S6
Tweet 1	0.6					0.7
Tweet 2	0.3	0.1				
Tweet 3	0.5			0.9		
Tweet 4			0.5	0.4		
Tweet 5		0.2			0.2	
Tweet 6		0.1	0.1			
Tweet 7		0.1				0.3

Article	*Popularity_Wt*
B1	1.4
E4	1.3
S6	1.0
B3	0.6
B2	0.5
S5	0.2

Example: News Recommendation (3)

Profile-based Recommender

Personal score is based on the similarity between users profiles (supplied by user by manually scoring articles) and the articles available.



$$Personal\_Wt_{ij} = CosineSimilarity\left(ArticleProfile_i,\ UserProfile_j\right)$$

Category	Weight
Business	6
Entertainment	1
Sports	3

*

Articles	Business Wt	Entertainment Wt	Sports Wt
B1	0.3	0.2	0.0
B2	0.7	0.0	0.6
B3	0.4	0.7	0.0
E4	0.0	8.0	0.2
S5	0.6	0.1	0.0
S6	0.4	0.1	0.0

Article	*Personal_Wt*
B2	6.0
S5	3.7
B3	3.1
S6	2.5
B1	2.0
E4	1.4

Example: News Recommendation (4)

$$Hybrid\_Wt2_{ij} = \alpha * Popularity\_Wt_j + (1-\alpha) * Personal\_Wt_{ij}$$

Article	*Popularity_Wt*	Normalized *Popularity_Wt*	*Personal_Wt*	Normalized *Personal_Wt*
B1	1.4	1.00	2	0.33
B2	0.5	0.36	6	1.00
B3	0.6	0.43	3.1	0.52
E4	1.3	0.93	1.4	0.23
S5	0.2	0.14	3.7	0.62
S6	1.0	0.71	2.5	0.42

Article	*Hybrid_Wt*
B2	0.36
B1	0.33
S6	0.30
B3	0.22
E4	0.22
S5	0.09

Example: Spotify

Combines 3 different Approaches:

Collaborative Filtering
using implicit feedback, # times played etc

+

Songs & Artist Models
uses web crawling to get sentiment & buzz about songs and artist

+

Audio Modelling
use CNN to convert raw audio into a feature set (tempo, liveliness, danceability etc). Then match to users past listens

For each user, there are two listening histories we take into consideration: the set of all tracks a user listened to and the set of all artists a user listened to. Thus, we are able to compute a *artist similartiy (artistSim)* and a *track similarity (trackSim)* as shown in Equations 2 and 3.

$$artistSim_{i,j} = \frac{|artists_i \cap artists_j|}{|artists_i \cup artists_j|} \quad (2)$$

$$trackSim_{i,j} = \frac{|tracks_i \cap tracks_j|}{|tracks_i \cup tracks_j|} \quad (3)$$

The final user similarity is computed using a weighted average of both, the *artistSim* and *trackSim* as depicted in Equation 4.

$$sim_{i,j} = w_a * artistSim_{i,j} + w_t * trackSim_{i,j} \quad (4)$$

http://ceur-ws.org/Vol-1313/paper_7.pdf

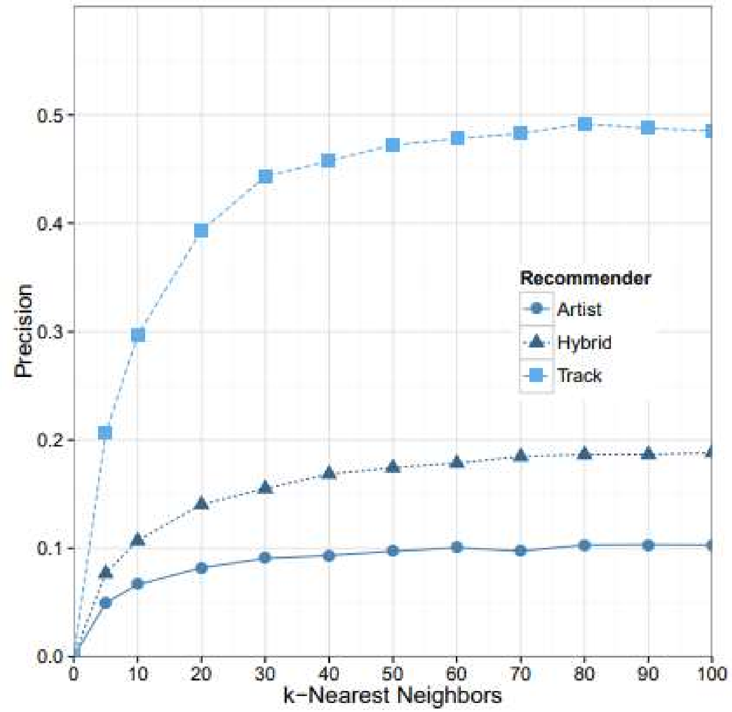


Figure 3: Precision and Recall of the Track-Based Recommender

https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

Commercial Tools and Platforms

- There are probably 100's of tools & products (and survey sites!) …….

https://adoric.com/blog/top-25-product-recommendation-tools/

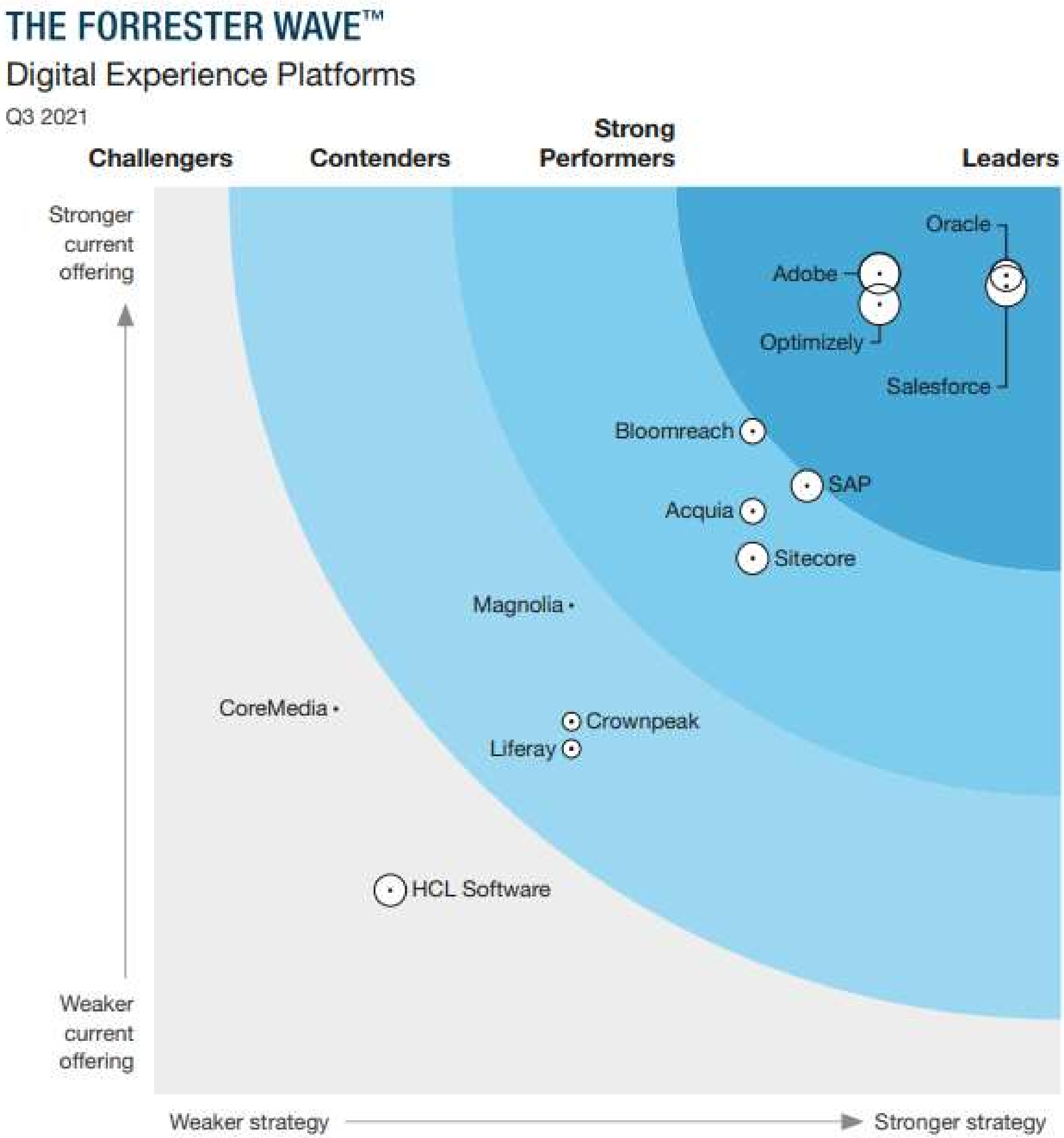
https://analyticsindiamag.com/5-open-source-recommender-systems-you-should-try-for-your-next-project/

https://aimultiple.com/recommendation-engine#top

Compare Recommendation Engines

AIMultiple is data driven. Evaluate 14 products based on comprehensive, transparent and objective AIMultiple scores. For any of our scores, click the icon to learn how it is calculated based on objective data.

Commercial Tools and Platforms



Vendor	Product evaluated
Acquia	Acquia Open DXP
Adobe	Adobe Experience Cloud
Bloomreach	Bloomreach Experience Platform (BRX)
CoreMedia	CoreMedia Content Cloud
Crownpeak	Crownpeak Digital Experience Platform
HCL Software	HCL Digital Experience
Liferay	Liferay Digital Experience Platform
Magnolia	Magnolia
Optimizely	Optimizely Digital Experience Platform
Oracle	Oracle Advertising & Customer Experience Platform
Salesforce	Salesforce Experience Cloud
SAP	SAP Customer Experience
Sitecore	Sitecore Experience Platform (XP)

https://reprints2.forrester.com/#/assets/2/367/RES161679/report

Experience (Optimization) Platforms

- *"Experience Optimization (EXO) is the ongoing process of understanding your customers and providing the best possible experience for them across all touchpoints. EXO allows businesses to improve the experiences of their customers through controlled experimentation and dynamic experience delivery. Using Experience Optimization, businesses are able to vet and quantify the impact of their ideas with real-time customer data in ways that were not possible before."*

https://www.optimizely.com/optimization-glossary/experience-optimization/

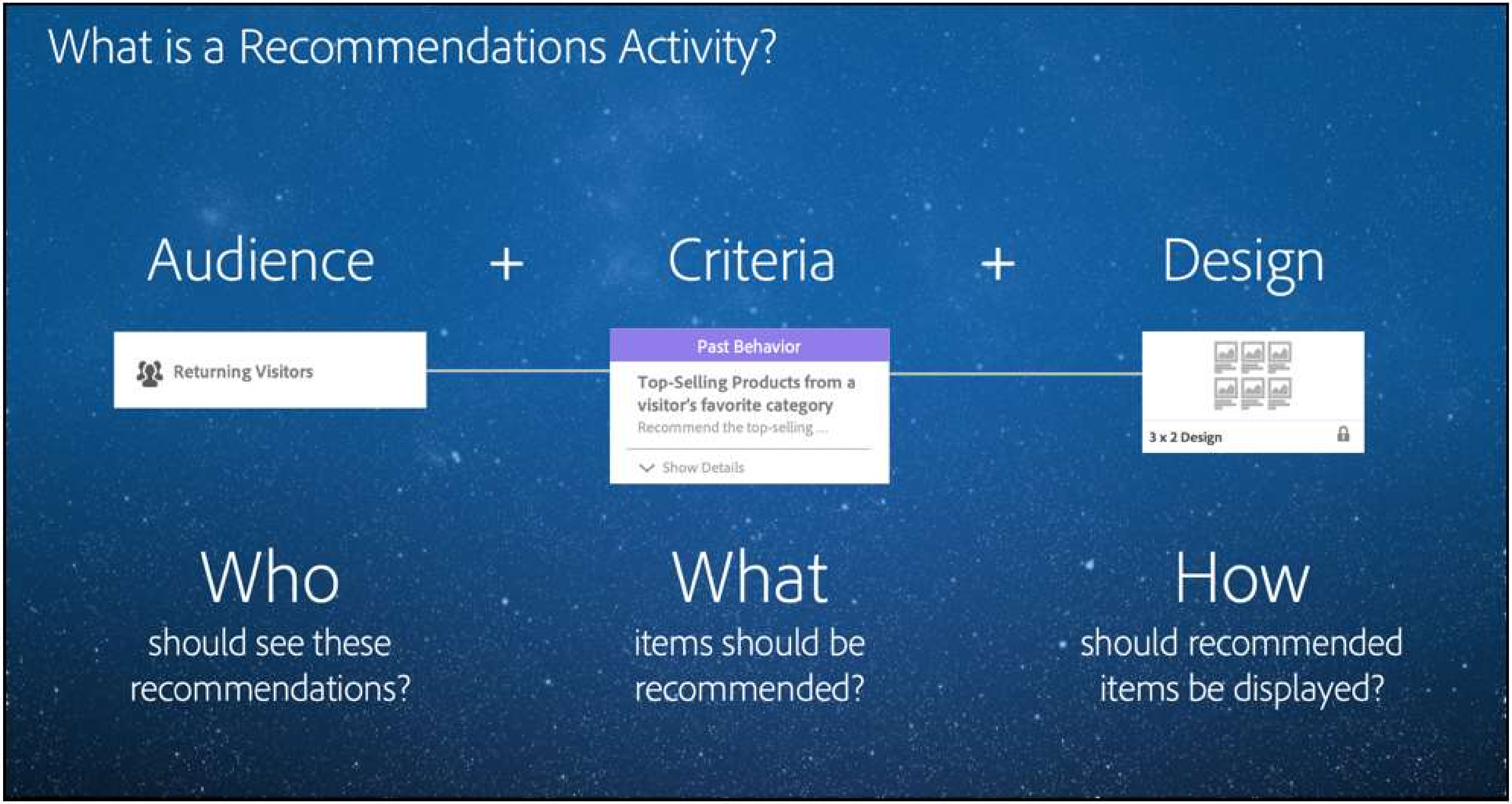
Experience

Commercial Tools and Platforms

Figure 2: Forrester Wave™: Experience Optimization Platforms Scorecard, Q4 2020

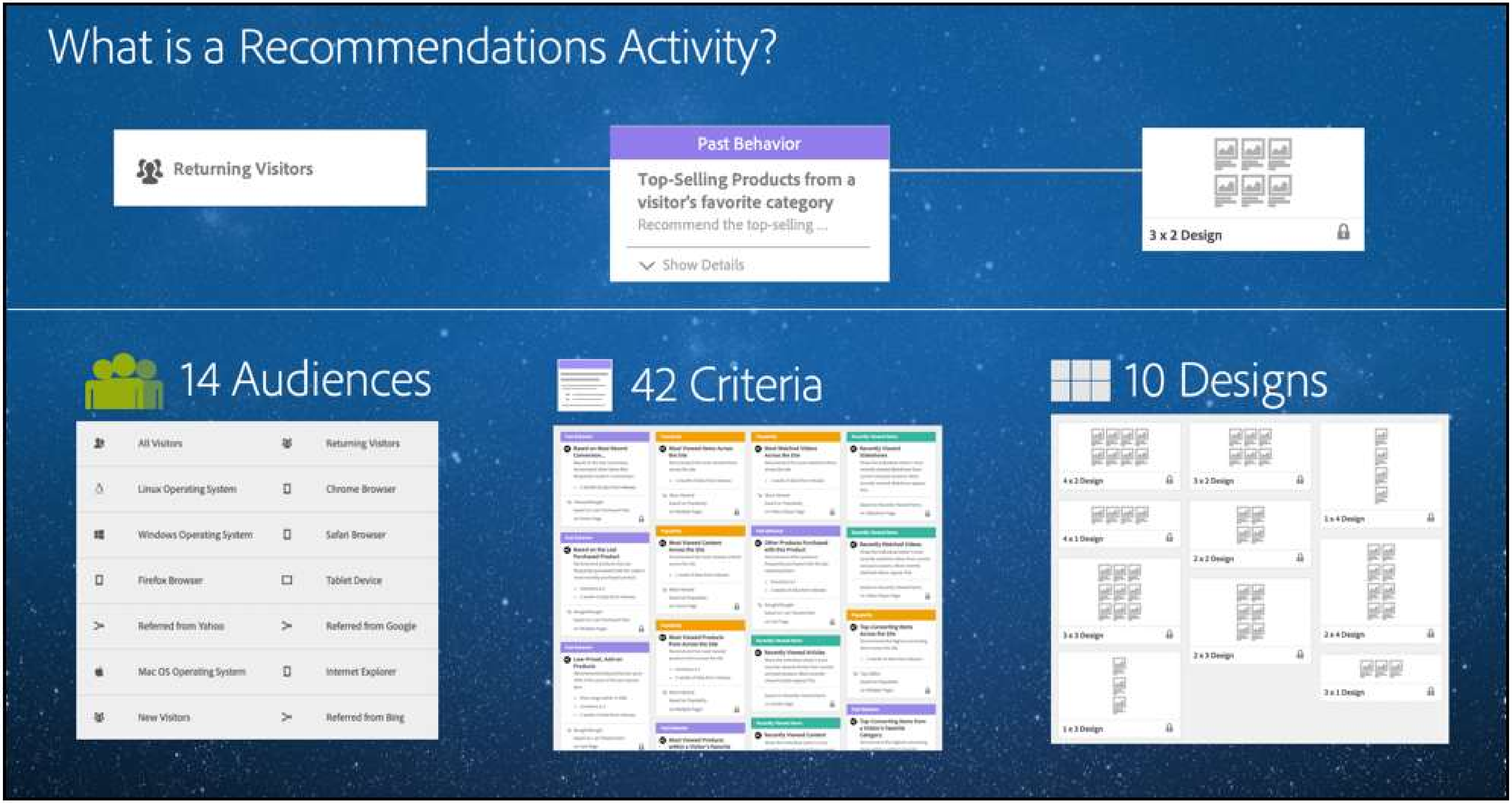
	Forrester's weighting	AB Tasty	Adobe	Dynamic Yield	Kameleoon	Kibo Commerce	Optimizely	Oracle	Salesforce	SAS*	SiteSpect
Current offering	50%	1.88	3.79	3.06	1.58	3.47	2.84	3.74	3.48	1.79	2.63
Online testing	40%	2.60	3.00	3.80	1.80	3.80	4.20	4.20	3.40	1.40	3.40
Behavioral targeting	20%	1.00	3.67	3.00	1.00	4.33	1.00	3.00	5.00	2.33	1.67
Recommendations	15%	1.00	4.60	3.00	1.00	3.80	1.00	3.00	4.20	2.20	1.40
Experience optimization technique innovation	5%	1.00	5.00	1.00	3.00	3.00	1.00	5.00	1.00	3.00	1.00
Platform experience	20%	2.20	4.60	2.20	1.80	1.80	3.80	3.80	2.20	1.40	3.40

Adobe Target Recommendations



https://experienceleague.adobe.com/docs/target/using/recommendations/introduction-to-recommendations.html?lang=en

Adobe Target Recommendations



Adobe Target Recommendations

Criteria in Adobe Target are rules that determine which products or content to recommend based on a predetermined set of visitor behaviors

Criteria	Description
Items/Media with Similar Attributes	Recommends items or media similar to items or media based on current page activity or past visitor behavior.
People Who Viewed This, Viewed That	Recommends items that are most often viewed in the same session that the specified item is viewed.
People Who Viewed This, Bought That	Recommends items that are most often purchased in the same session that the specified item is viewed.
People Who Bought This, Bought That	Recommends items that are most often purchased by customers at the same time as the specified item.
Site Affinity	Recommends items based on the certainty of a relationship between items.
Top Sellers	The items that are included in the most completed orders. Multiple units of the same item in a single order are counted as one order.
Most Viewed	The items or media that are viewed most often.
User-Based Recommendations	Recommends items based off of each visitor's browsing, viewing, and purchasing history. These items are generally referred to as "Recommended for You."

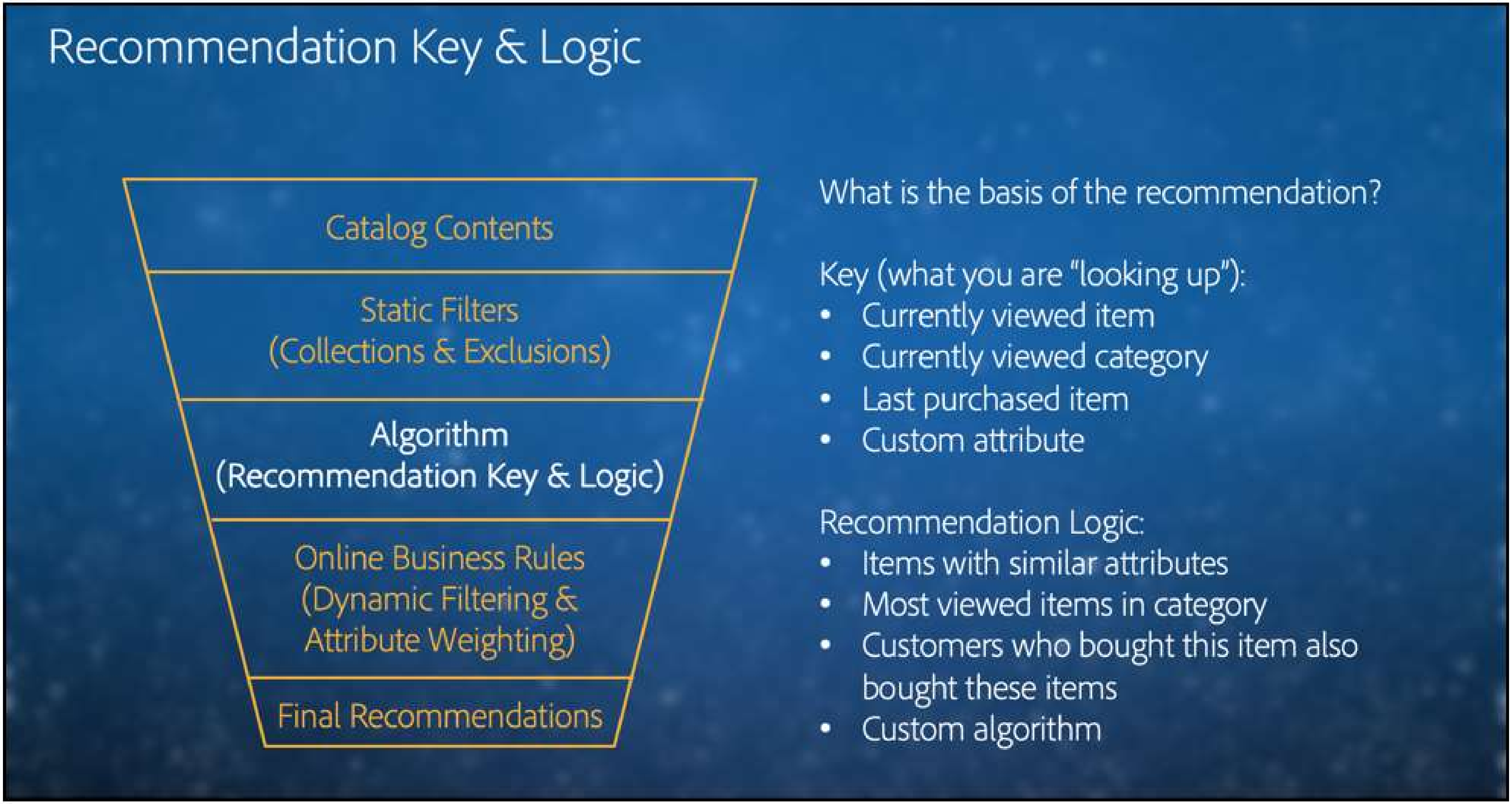
https://experienceleague.adobe.com/docs/target/using/recommendations/criteria/algorithms.html?lang=en

Adobe Target Recommendations

Out of the box, Target includes a portfolio of algorithms.

Last Updated: August 24, 2021

Adobe Target Recommendations



Adobe Target Recommendations

Challenges and Issues – User Experience

The success of a recommender systems depends on more than just accuracy:

- **Diversity** - users tend to be more satisfied with recommendations when there is a higher diversity, e.g. items from different artists. There is unlikely to be a single best recommendation, allow the user to treat the RS as a knowledge discovery tool
- **Serendipity** - how surprising are the recommendations?
- **Avoiding bad recommendations -** assigning a cost to them
- **Long versus short term recommendations.** How far ahead can the recommendation be? e.g. associations found between items over longer time
- **Repeat Recommendations** - sometimes it may be more effective to re-show recommendations or let users re-rate items, than showing new items. Users may ignore a recommendation first time but still like the item, e.g. if short of time
- **Recommending Sequences** – sometimes the sequence is important e.g. a compilation of musical tracks from slow to fast, episodes in Game of Thrones etc.

Challenges and Issues – Privacy

- **Privacy** - push-back by users if they feel the RS is collecting too much information about them.
  - The Netflix Challenge data was anonymised but in 2007 two researchers from the University of Texas were able to identify individual users by matching the data sets with film ratings on the Internet Movie Database.
  - As a result, in December 2009, an anonymous Netflix user sued Netflix in Doe v. Netflix.
  - This led in part to the cancellation of a second Netflix Prize competition in 2010.

RYAN SINGEL 03.12.10 02:40 PM

NetFlix Cancels Recommendation Contest After Privacy Lawsuit

Challenges and Issues - Trust

- Trust can be built by explaining how the recommendations are generated and why an item is being recommended
- **F**ake reviews and fake ratings erode trust

MENU FORTUNE SUBSCRIBE

Yelp's fake review problem

by Daniel Roberts @readDanwrite SEPTEMBER 26, 2013, 3:05 PM EST

A New York sting operation caught businesses paying for positive ratings on recommendation websites.

Challenges and Issues - Trust

The Guardian
International edition
For 200 years

Haroon Siddique

Wed 27 Oct 2021 00.01 BST

https://www.theguardian.com/travel/2021/oct/27/almost-1m-tripadvisor-reviews-in-2020-found-to-be-fraudulent

Almost 1m Tripadvisor reviews in 2020 found to be fraudulent

In total Tripadvisor penalised 34,605 properties for fraudulent activity and banned 20,299 members

▲ The report said: 'While our overall review contributions dropped in line with the slowdown in travel, fraudulent submissions did not follow the same trend.' Photograph: Jakub Porzycki/NurPhoto/REX/Shutterstock

Almost 1m reviews submitted for inclusion on Tripadvisor - equivalent to 3.6% of the total - were determined to be fraudulent by the website last year.

In its second transparency report - the first was released in 2019 - the travel guidance platform said 67.1% of the fake reviews had been caught before making it on to the platform by its pre-posting moderation algorithm.

In 2019, Tripadvisor rejected as "simplistic" analysis by consumer group Which? of 250,000 hotel reviews on its site, which found one in seven had "blatant hallmarks" of being fake.

The report, published on Wednesday, also provided details on paid reviews,

Challenges and Issues - Computability

- **Scalability** – handling very large numbers of users and items
- **Cold-start** – making recommendations to a new user
- **Sparsity** - recommending items in **the long-tail** is hard since there are very few ratings / purchases for them

ARTICLE INTERNET, MEDIA

Do recommendation systems make the 'tail' longer or shorter?

By Paul Belleflamme 26 April 2012 39 (Updated March 2015)

http://www.ipdigit.eu/2012/04/do-recommendation-systems-make-the-tail-longer-or-shorter/

Challenges and Issues - Computability

Innovation

Techdirt | Wireless News | Innovation | Case Studies | Startups | Net Neutrality | Techdirt Deals!

Why Netflix Never Implemented The Algorithm That Won The Netflix $1 Million Challenge

from the *times-change* dept

Innovation
by Mike Masnick
Fri, Apr 13th 2012
12:07am

- Despite all the plaudits and case studies, Netflix announced this week that despite paying $1 million dollars to a winning team of multinational researchers in 2009, they never bothered to implement their solution.
- Why? Because, according to Netflix the "additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment."

Instead

…they gave us the source code. We looked at the two algorithms with the best performance in the ensemble: *Matrix Factorization* (generally called SVD, *Singular Value Decomposition*) and *Restricted Boltzmann Machines* (RBM).To put these to use, we had to overcome some limitations, for instance that they were built to handle 100 million ratings, instead of the 5 billion+ that we have, and they were not built to adapt as members added more ratings. Once we overcame those challenges, we put the two algorithms into production, where they are still used as part of our recommendation engine.

https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429

Class End!