GENERAL ASSEMBLY

# HULU CASE STUDY

## RECOMMENDATION ENGINES

Let’s look at how the Hulu recommendation engine works. We’ll also see how they made a business case for it.

1. Jaccard Similarity
2. Hulu Recommendation Engine

# I. JACCARD SIMILARITY

How do we define “similarity” of users?

Jaccard Similarity:

- Defines similarity between two sets of objects.
- Does not require converting the objects to numbers!

How do we define “similarity” of users?

Jaccard Similarity:

Defines similarity between two sets of objects

$$JS(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

How do we define “similarity” of users?

Jaccard Similarity:

Defines similarity between two sets of objects

$$JS(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

Number of similar elements

Number of distinct elements

$$JS(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

JS ({1, 2, 3}, {2,3,4}) = 0.5.

- |A Intersection B| = |{2, 3}| = 2.
- |A Union B| = |{1, 2, 3, 4}| = 4.

$$JS(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

# Exercise:

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

JS (User one, User two) = ?

$$JS(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

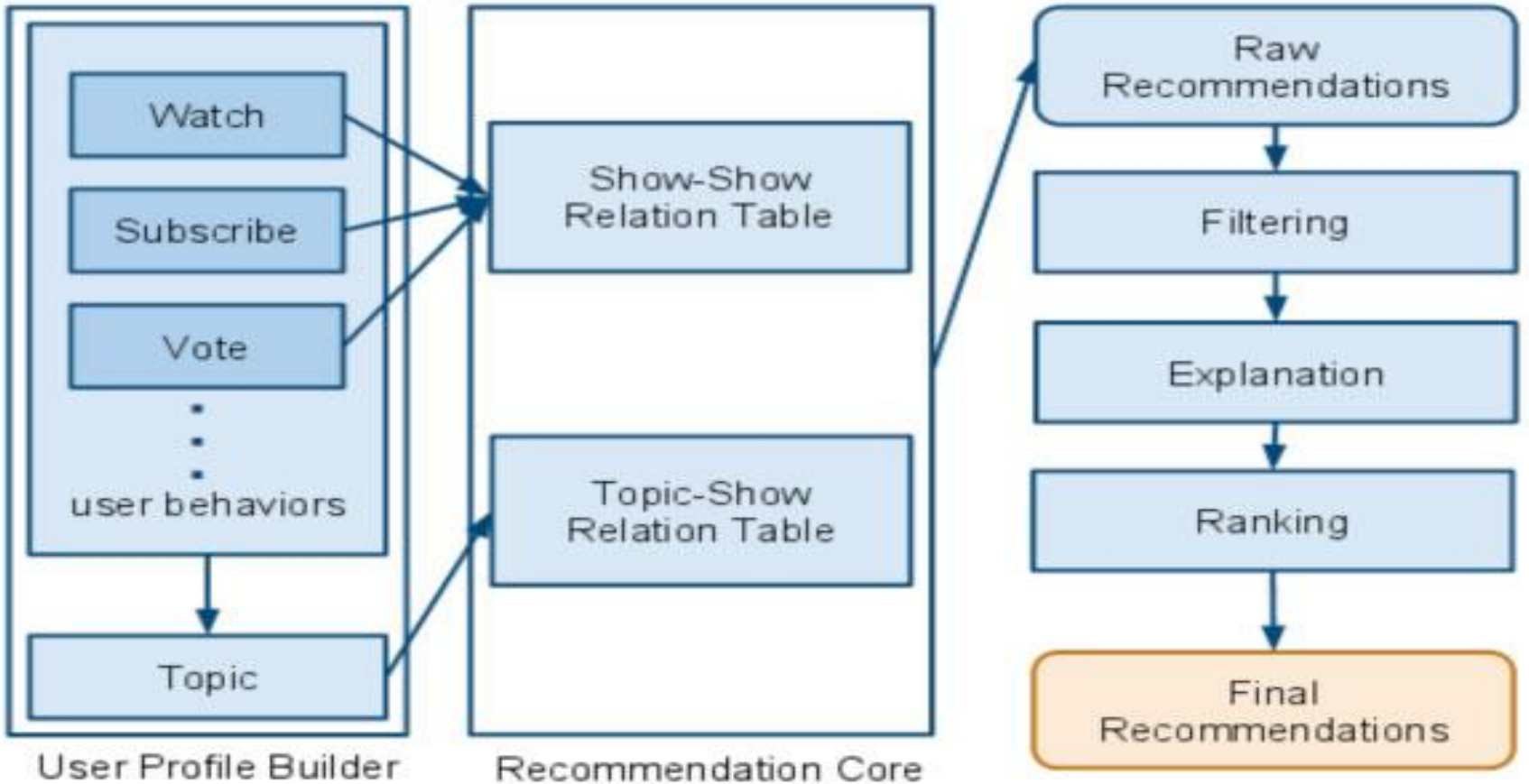
# Exercise:

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

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

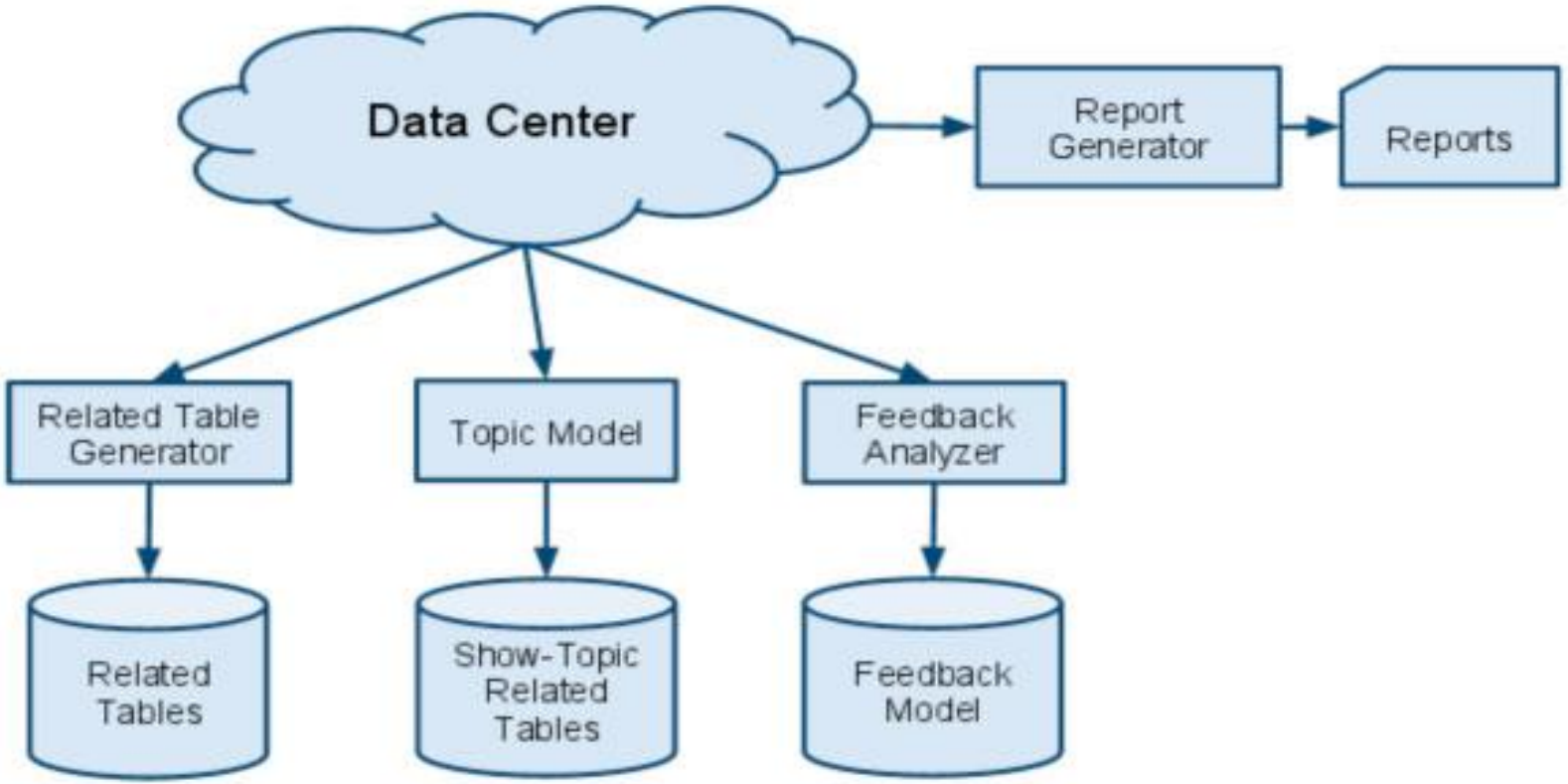
# II. HULU

# Hulu: On-line Architecture

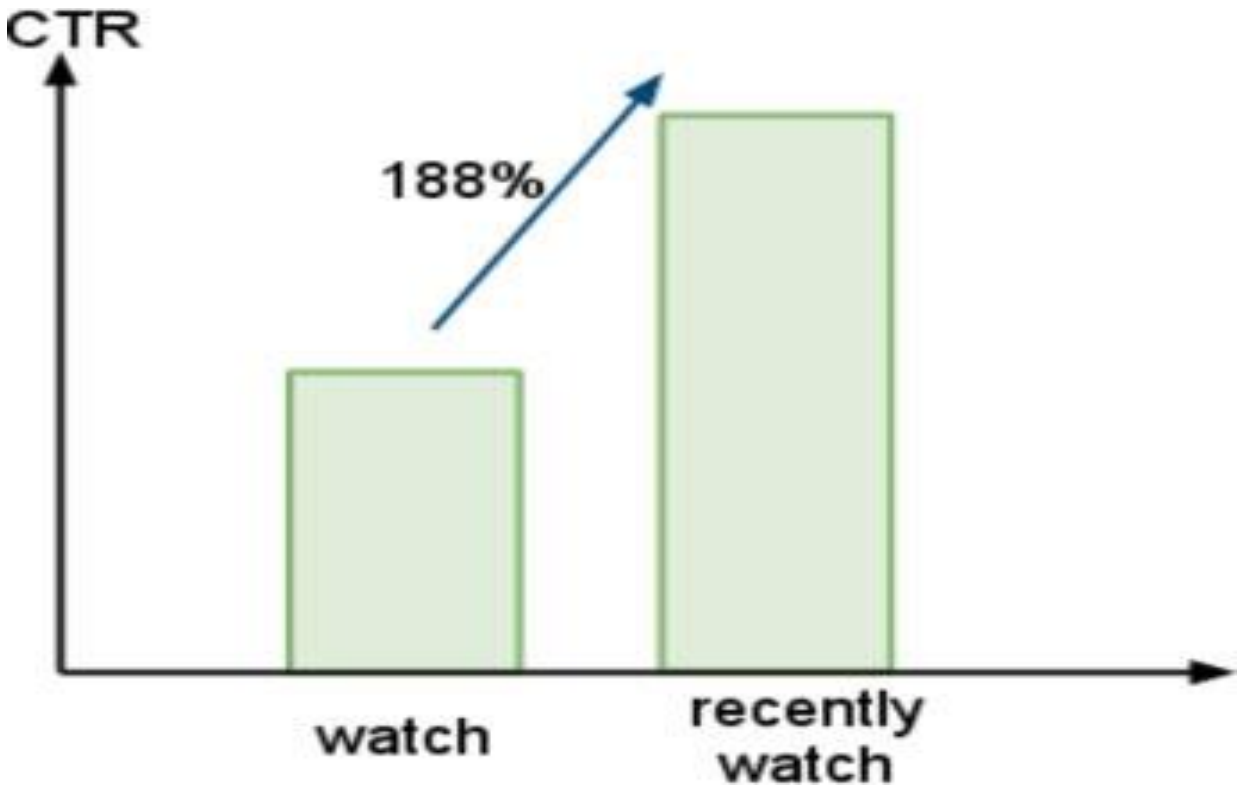



http://tech.hulu.com/blog/2011/09/19/recommendation-system/

# Hulu: Off-line Architecture

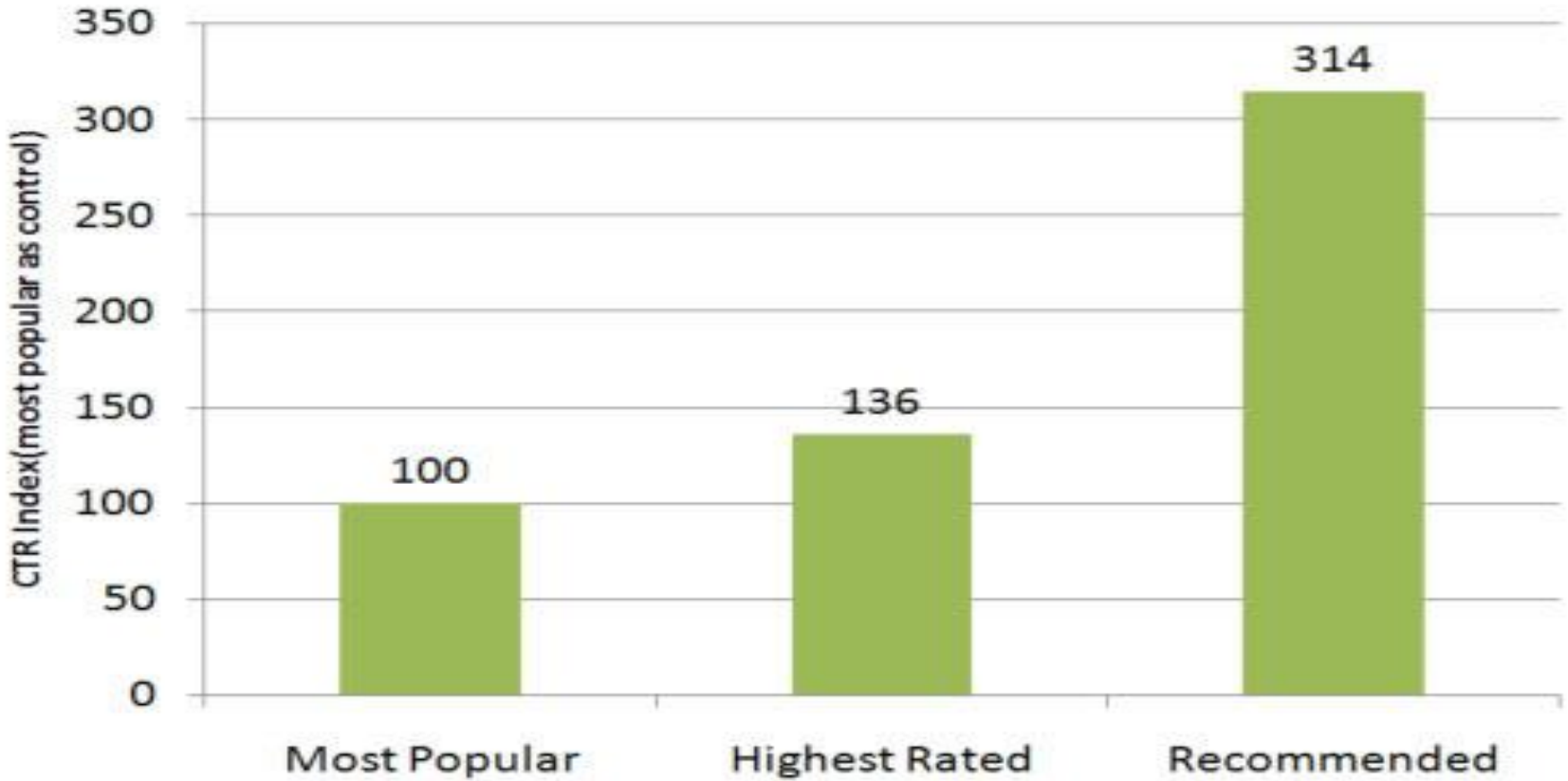



http://tech.hulu.com/blog/2011/09/19/recommendation-system/



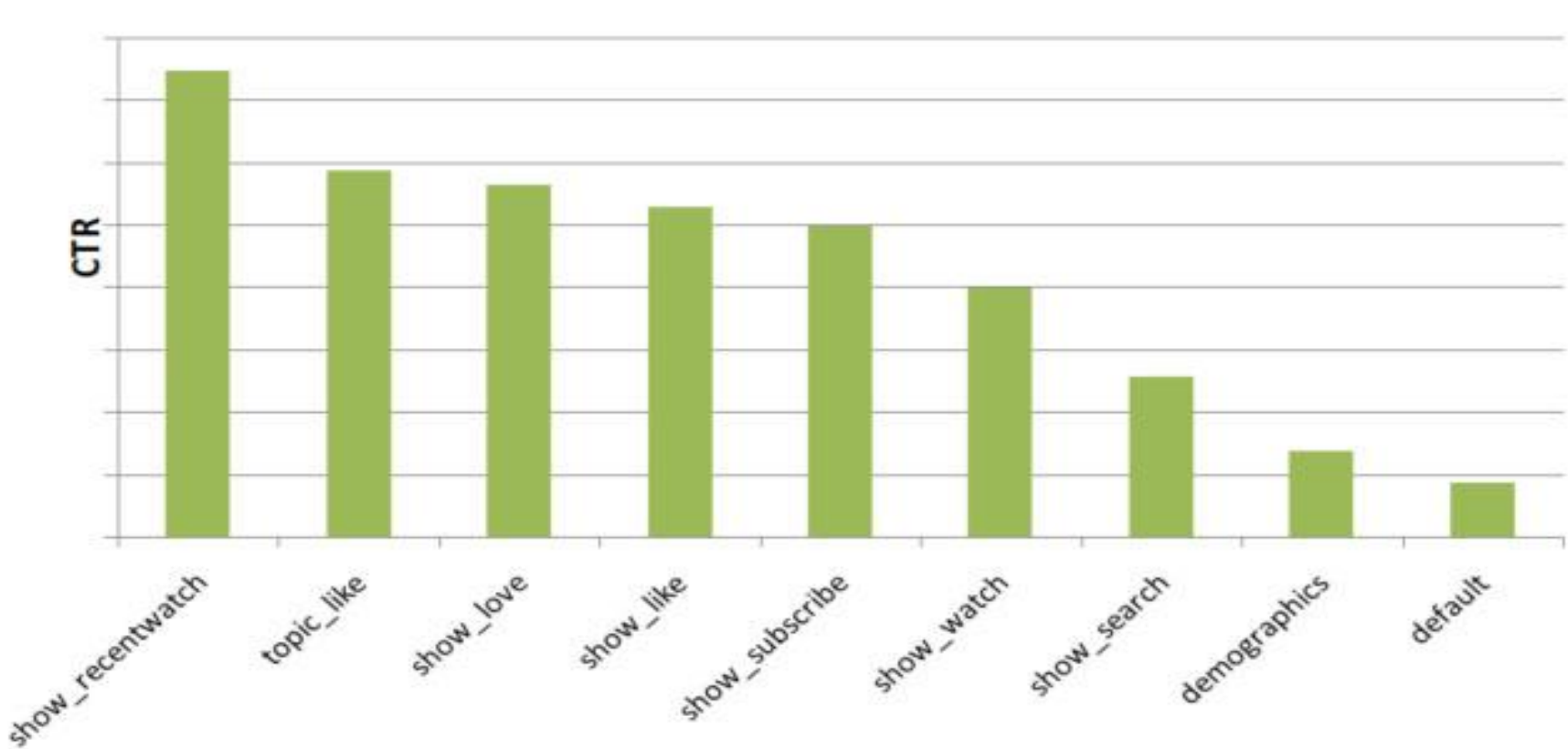

http://tech.hulu.com/blog/2011/09/19/recommendation-system/

# Hulu: Evidence that Recommendations work




http://tech.hulu.com/blog/2011/09/19/recommendation-system/

# Hulu: A/B Testing (Click-Through Rate)




http://tech.hulu.com/blog/2011/09/19/recommendation-system/

# Hulu: Similarity Metric

N(*i*): Set of users who watched show *i*.

s(*i*, *j*): Similarity between show *i* and show *j*

$$s(i,j) = \frac{|N(i) \cap N(j)|}{\sqrt{|N(i)||N(j)|}}$$

*NOTE: Every show will be rated as very similar to popular shows.*

http://tech.hulu.com/blog/2011/09/19/recommendation-system/

# Hulu: Item-based Collaborative Filtering

"ItemCF is the basis of all our algorithms"

N(*u*): Set of items user *u* has preferred previously.

$$p(u, i) = \sum_{j \in N(u)} r(u, j)s(i, j)$$

**p(*u*, *i*): User *u*'s preference on item *i*.**

r(*u, j*): Preference weight (rating) of user *u* on show *j*

s(*i*, *j*): Similarity between show *i* and show *j*

http://tech.hulu.com/blog/2011/09/19/recommendation-system/

# QUESTIONS?