COURSE: APPLICATIONS OF ARTIFICIAL INTELLIGENCE (AAI)

AMP in Business Analytics 2020

INDIAN SCHOOL OF BUSINESS

Vamshi Ambati

Vamshi_Ambati@isb.edu;
vamshi@predera.com

Session 3: Recommender Systems
February 29, 2020

USE CASES OF APPLICATION AREAS

- Natural Language Processing -> IE, IR, MT, NLU, NLG, Dialog -> Chatbot, Customer complaint translation
- Speech Processing -> SR (S2T), SS (T2S), SG (S2S) -> Voice Assistant (Siri), Education, IVR
- Computer Vision -> IU, IG, OR, OD, Video {U,G,R,D} -> OCR, Face Recognition, Self-driving, Entertainment
- Predictive Analytics ->Prediction, Recommender, Forecasting Systems -> E-commerce , Fraud detection
- Robotics -> Locomotion, Mechanics, Sensors, Planning -> Medical Bots, Agriculture, Military
- Agent Systems -> Planning, Reinforcement Learning, Space Optimization -> AlphaGo

THIS COURSE

- Session 1: Machine Translation
- Session 2: Dialog Systems (Chatbot)
- Session 3: Recommender Systems
- Session 4: Computer Vision (Image Recognition)

AGENDA

- Introduction
- Traditional Methods (Ecommerce)
- Machine Learning Approaches to Recommendation
  - Case Study of Netflix
- Summary & Challenges

1. INTRODUCTION

THEY ARE EVERYWHERE

- News recommendation
- Job recommendation
- Movie recommendation
- Restaurant recommendation
- Friend recommendation
- Event recommendation
- Product recommendation
- Content recommendation

WHY RECOMMENDER SYSTEMS?
CUSTOMER RATIONALE

80% of the total information in the world today has been created in the past 2 years

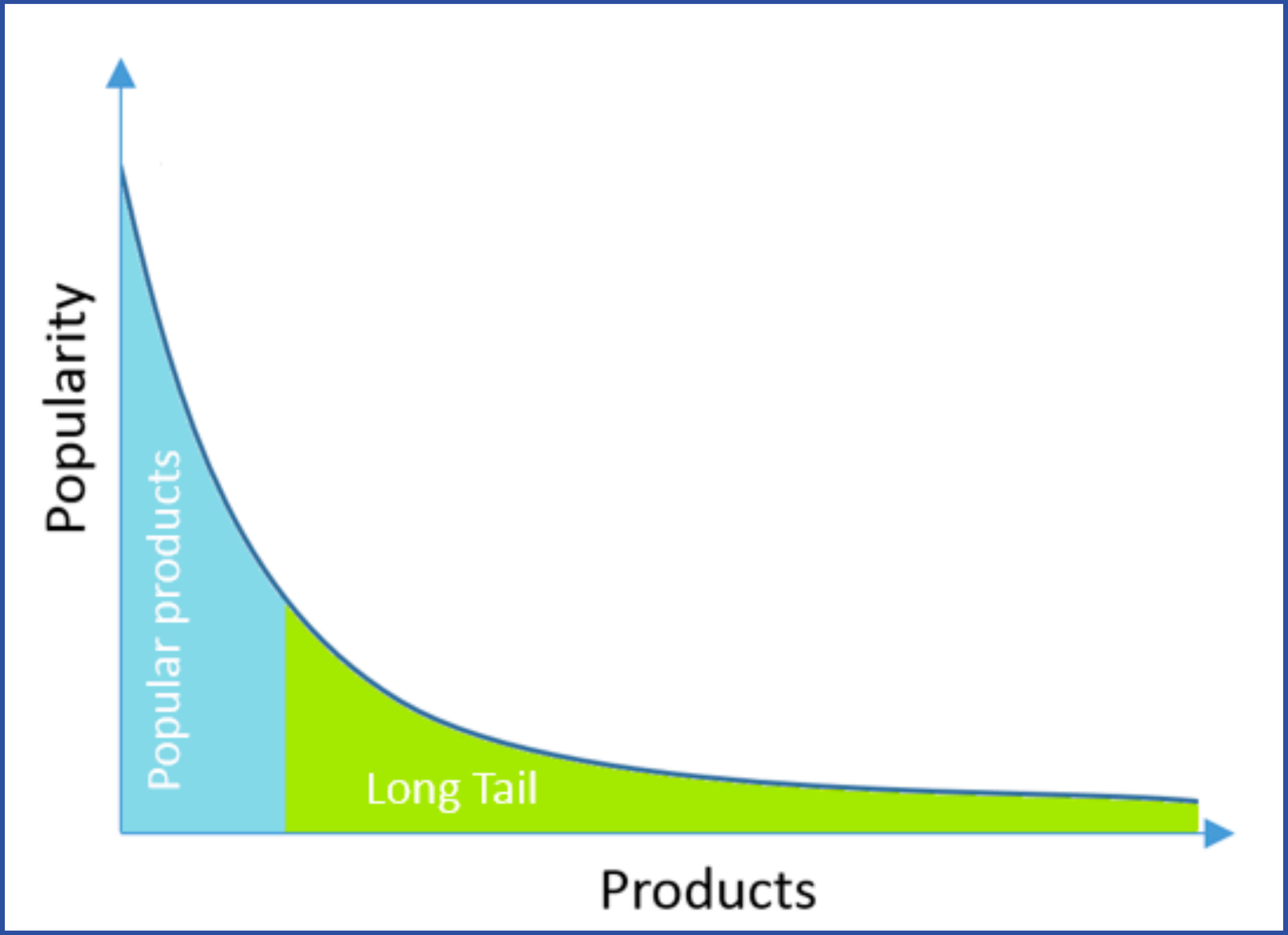
300 hours of video get added to Youtube every minute ; 5 billion video watched everyday

100 million+ photos are shared on Instagram everyday

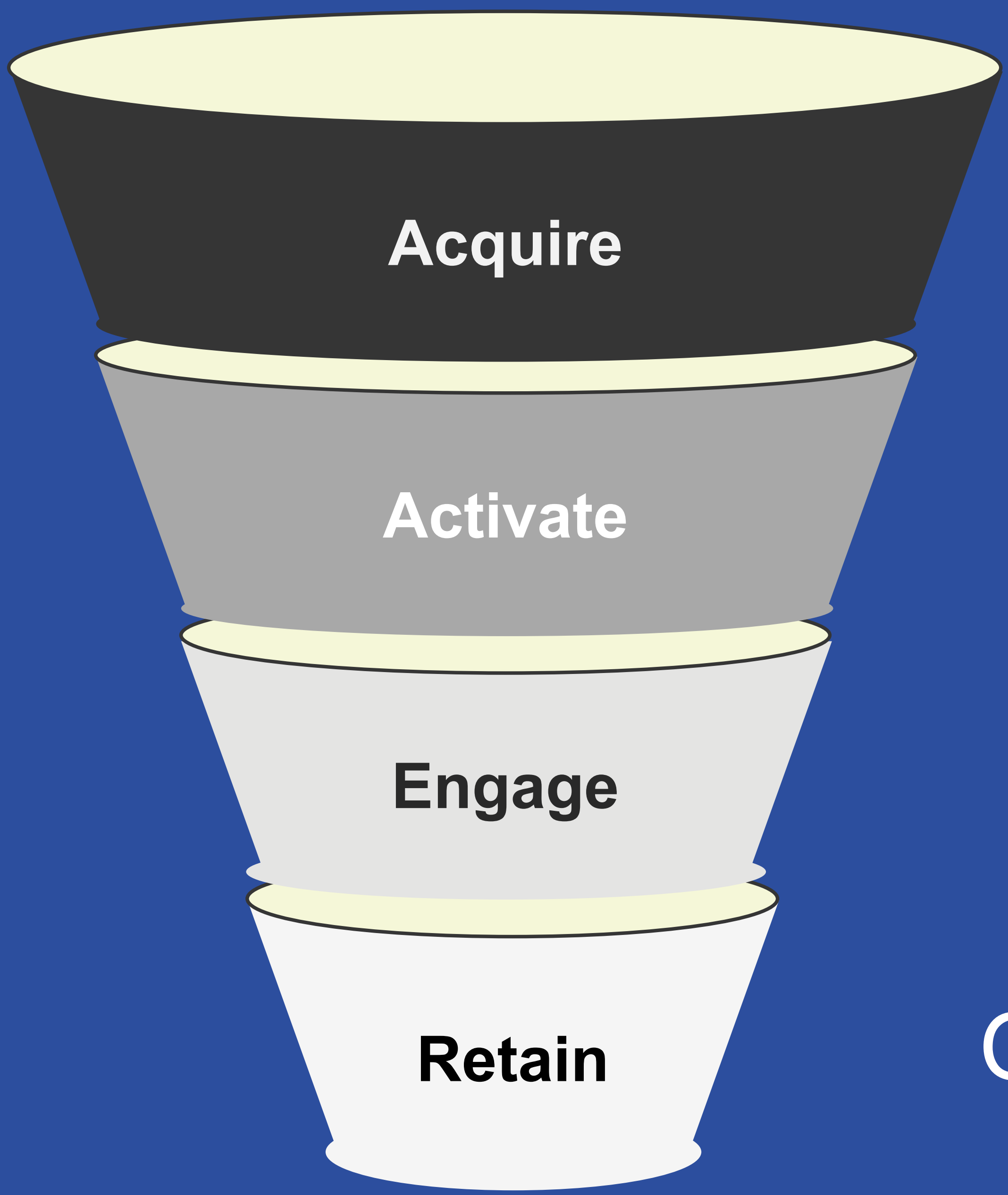
115M+ users consume 17M hours of content on Netflix

WHY RECOMMENDER SYSTEMS?
BUSINESS RATIONALE: LONG TAIL PROBLEM

Power's law applies to everyone



WHY RECOMMENDER SYSTEMS?
BUSINESS RATIONALE : THE ENGAGEMENT PROBLEM



Marketing / Advertising

Personalized on boarding

Recommender systems

Offers, Loyalty programs

DEFINITION OF A RECOMMENDER SYSTEM

- Users : U
- Items : I
- Utility Matrix : M
- Goal is to fill the utility matrix with maximum accuracy and minimum human effort

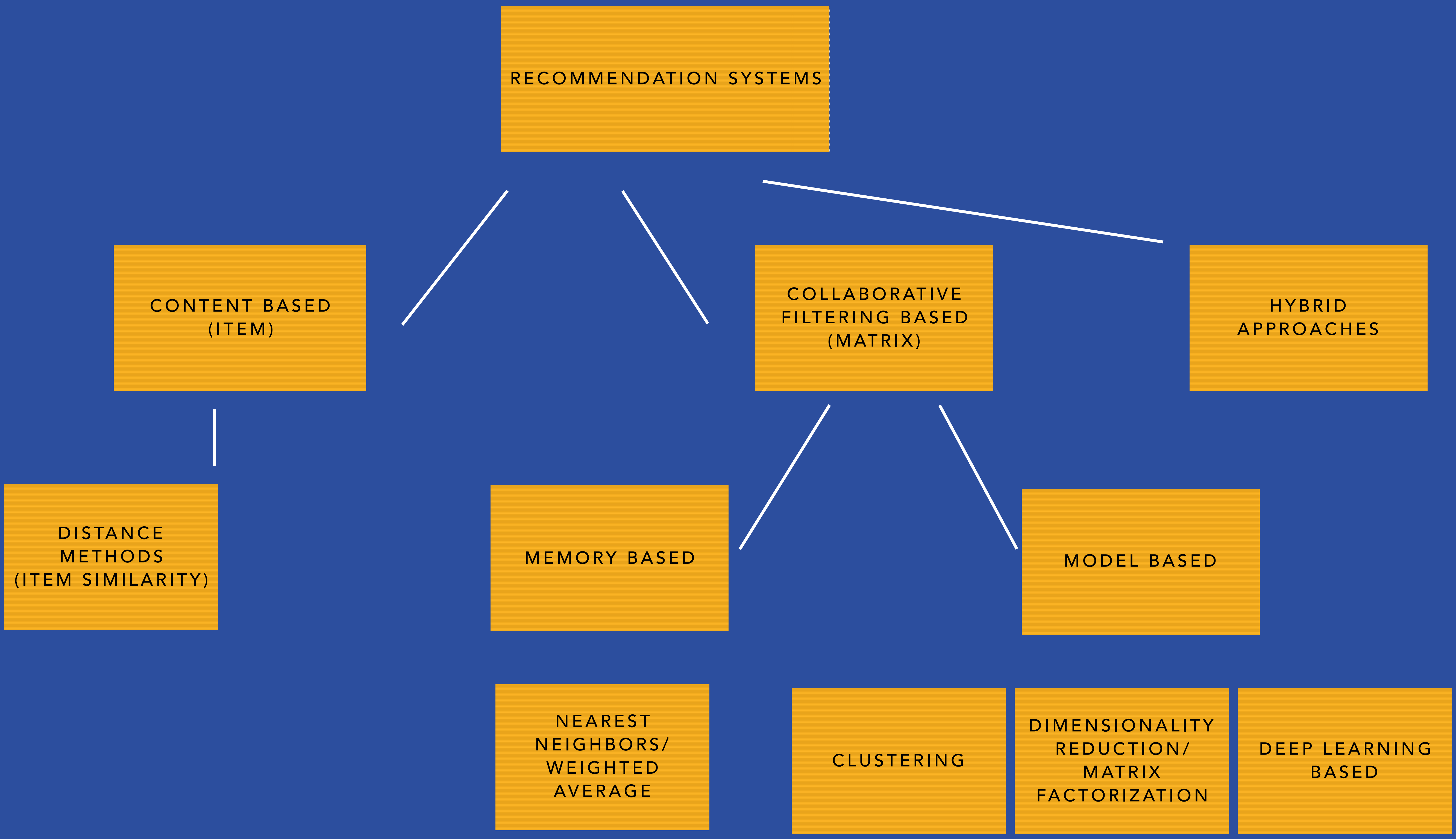
BENEFITS OF RECOMMENDATION

- Amazon
  - 35% of purchases for Amazon come from the product recommendation ("Frequently bought", "You may also like")
  - Creates serendipity which helps merchants and customers
- Netflix
  - More than 80 per cent of the TV shows and movies people watch on Netflix are discovered through their recommendation system
  - Significant data moat and continues to invest in IP

PERSONALIZATION VS RECOMMENDATION

- You visit a restaurant and the waiter
  - suggests vegetarian dishes because he remembers the last time you visited the you ordered vegetarian (PERSONALIZATION engine)
  - recommends tikka panner because several others that tried the dish liked it (RECOMMENDER System)

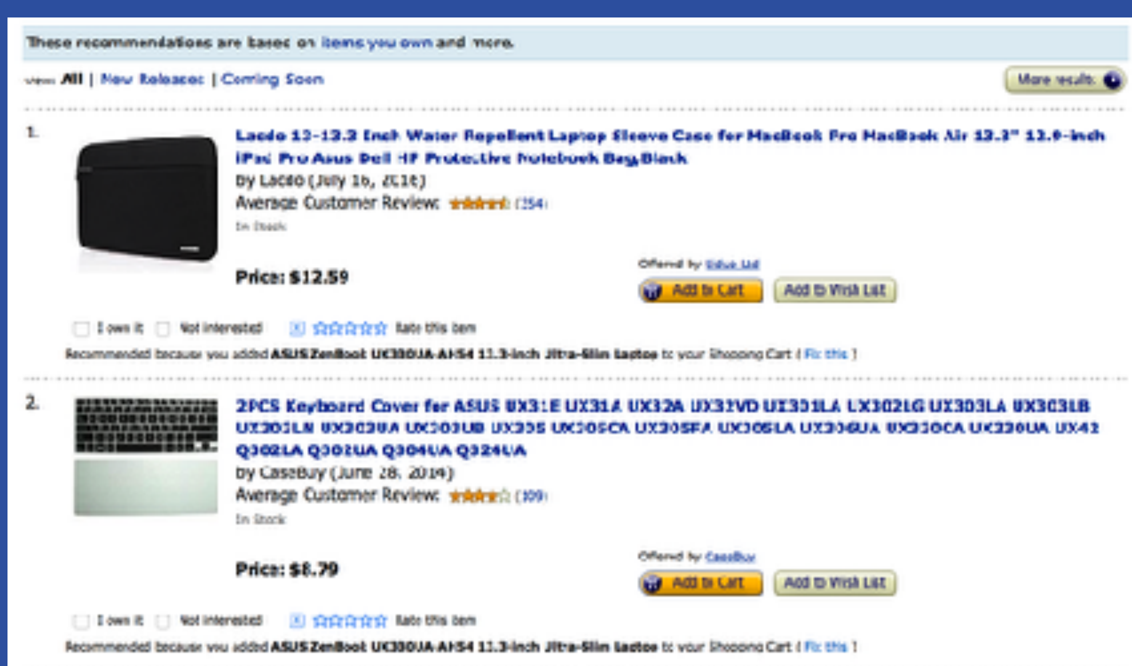
RECOMMENDER SYSTEMS

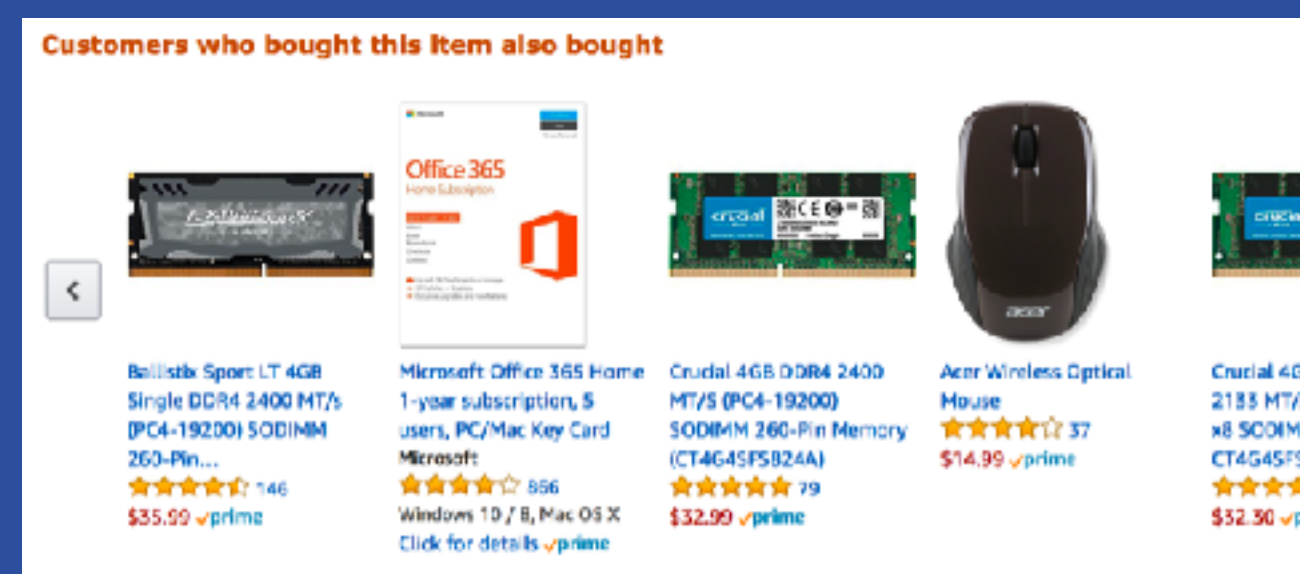


2. DATA MINING APPROACHES (ECOMMERCE)

ECOMMERCE AND RETAIL DATA

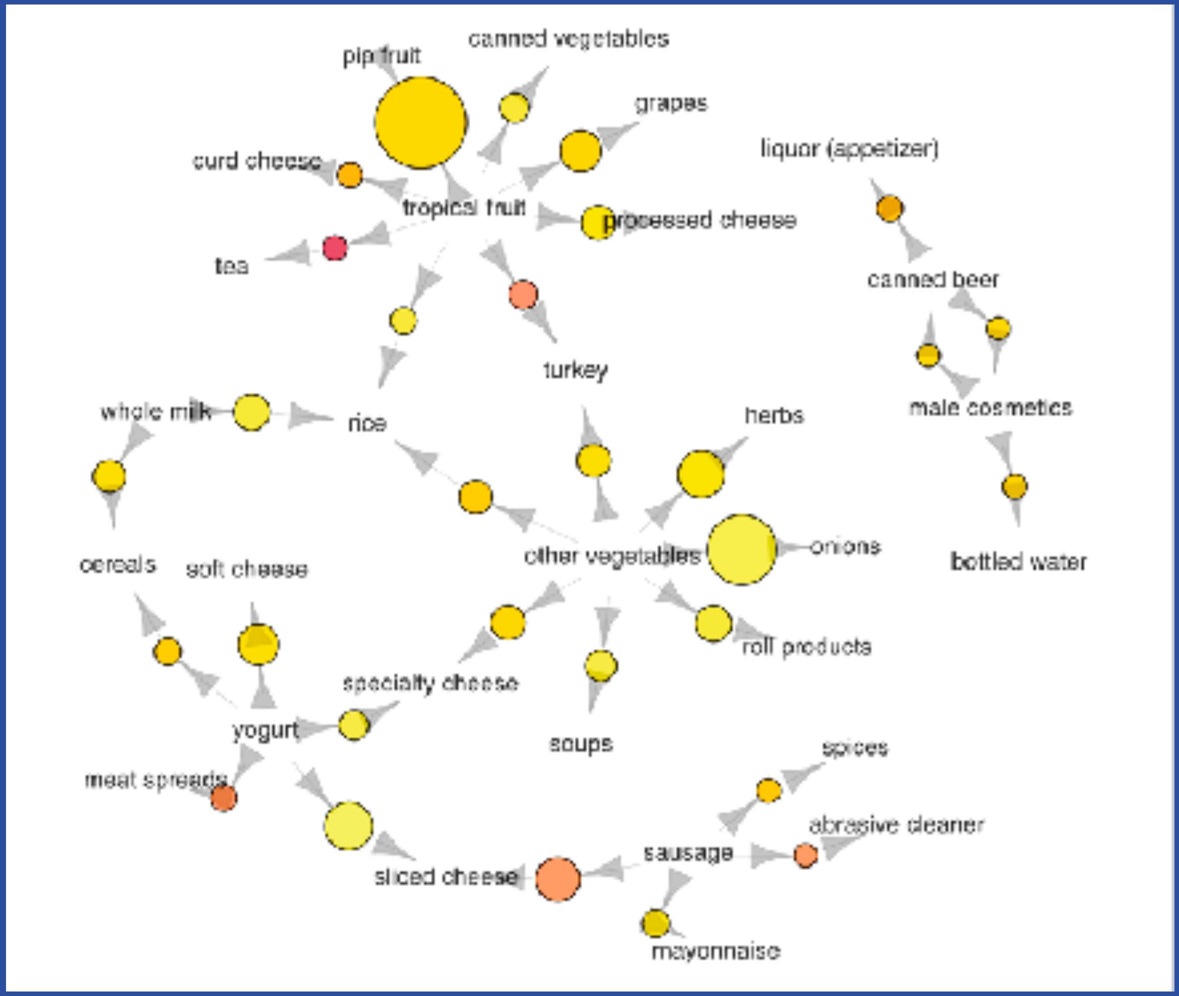
3 billion+ products are sold on Amazon in a year; 600 million+ listings





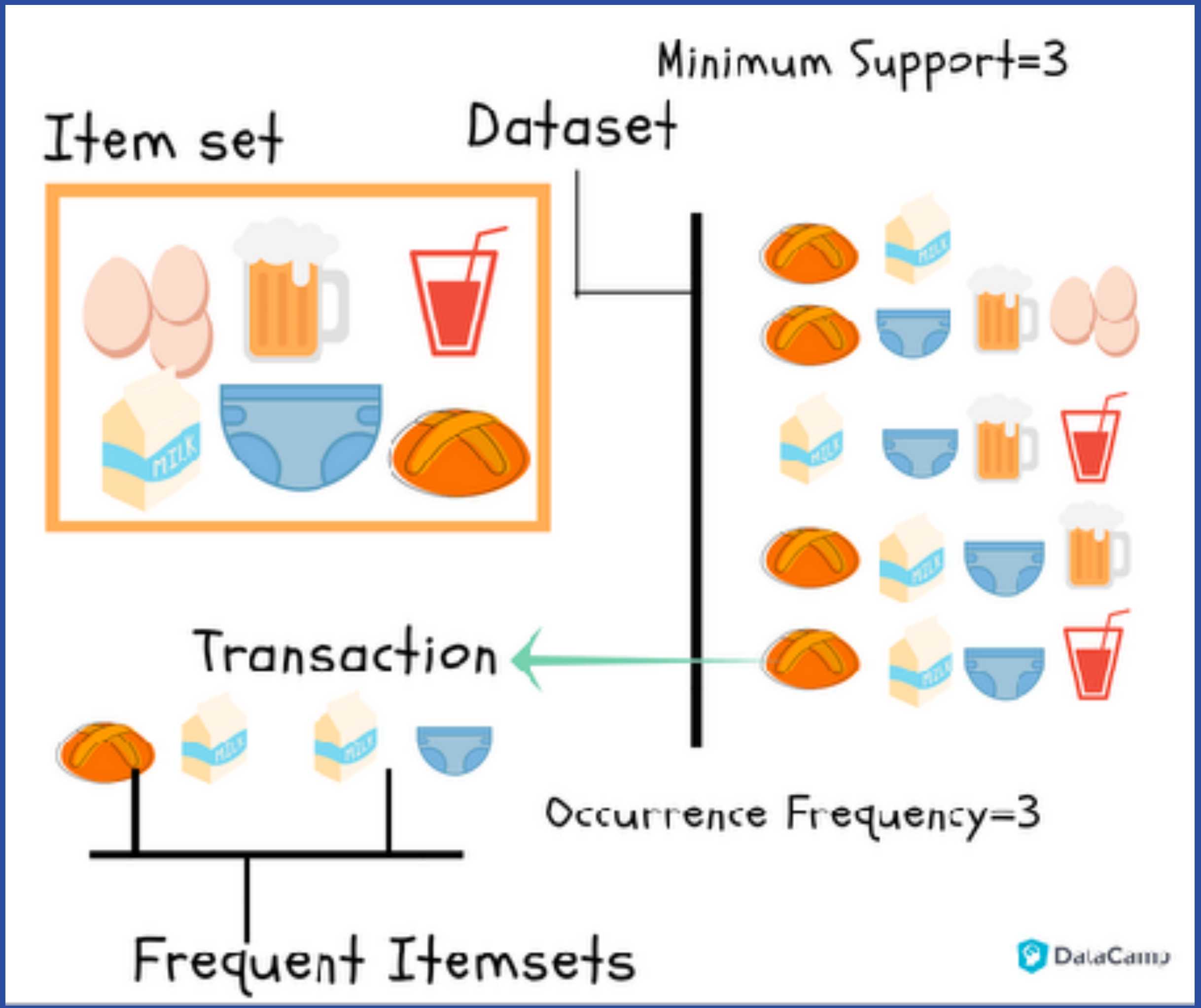
ECOMMERCE AND RETAIL

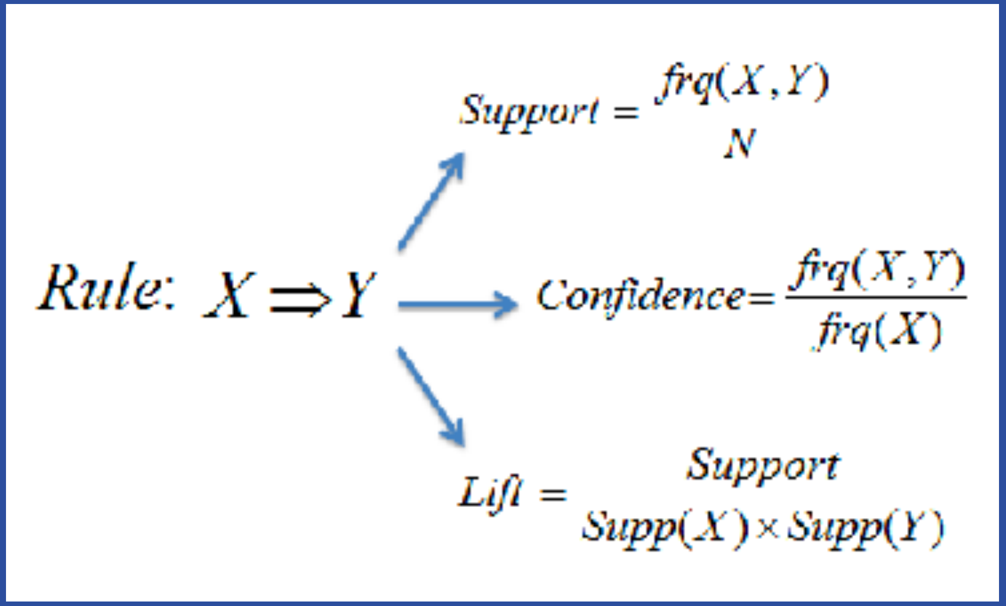
- Market Basket analysis | Shopping cart analysis
  - You are what you buy (customer segmentation)
  - You may also need this (Recommendation)
  - Your needs change over time (Loyalty, personalization)



AFFINITY ANALYSIS TECHNIQUES

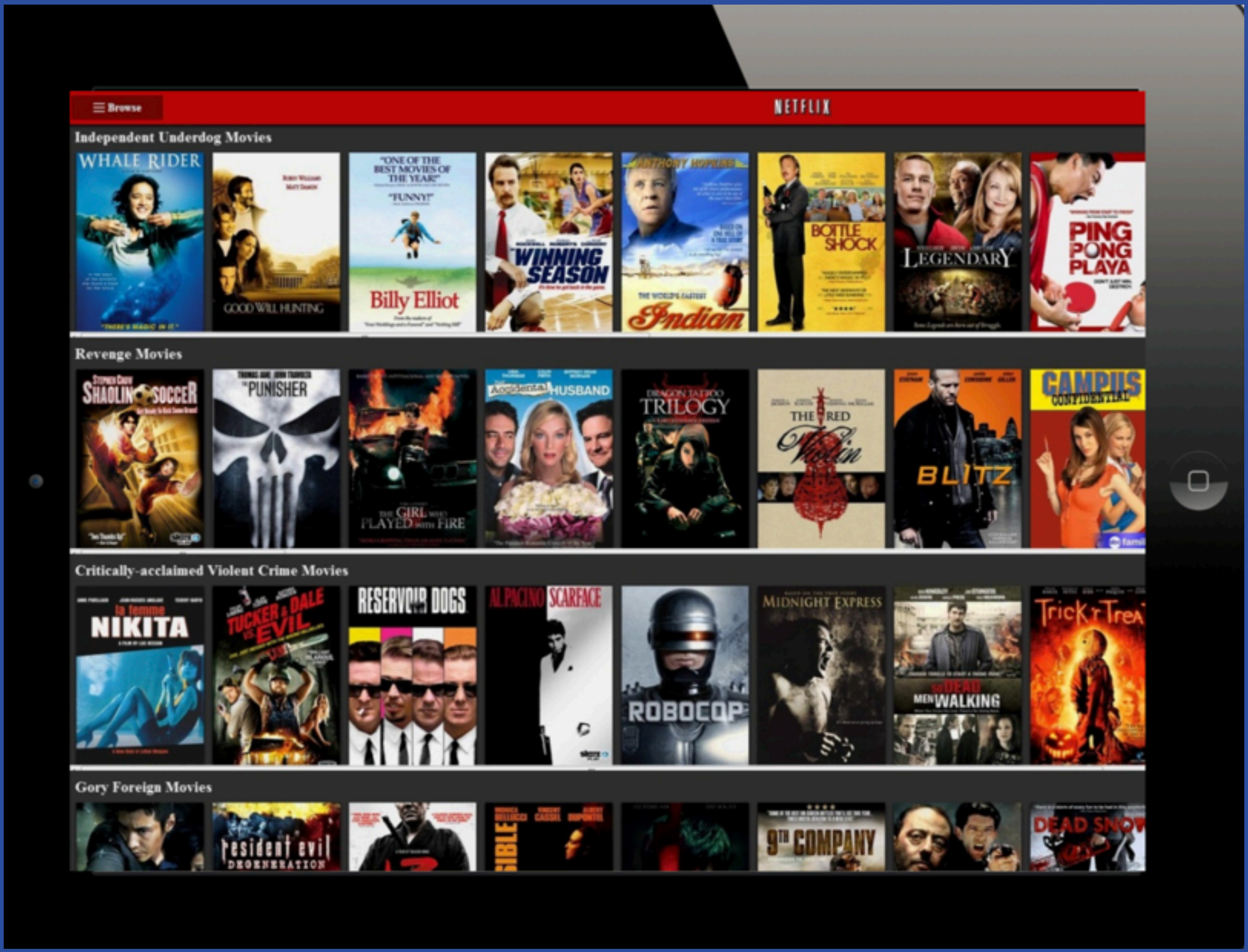
- Algorithms
  - Item-set mining | Association rules : Apriori algorithm | Eclat
- Challenges
  - Data sparsity
  - Computational complexity
- Pros
  - Easy to implement
  - Fast to compute
  - Scales well for really large sets (ecommerce)





3. MACHINE LEARNING APPROACHES
CASE STUDY OF NETFLIX

NETFLIX RECOMMENDATION



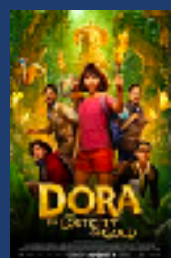
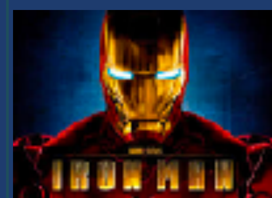
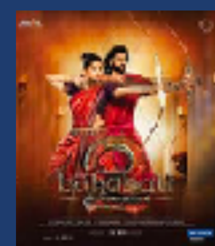
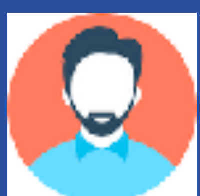
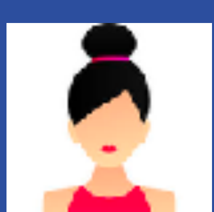
NETFLIX CHALLENGE (2006-2009)

- Goal: predicting the rating of the user
- Accuracy in predicting rating is compared as RMSE

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

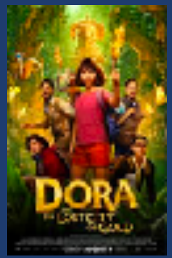
- Achieve 10% improvement in accuracy
- 500K users ratings data for 17K movies
- Price money of $1M
- Winning models that were computationally less-expensive were finally used as an ensemble

NETFLIX DATA

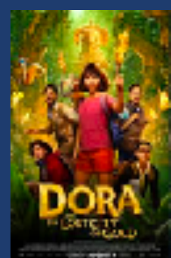
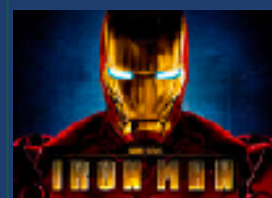
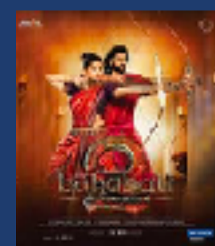
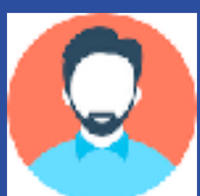
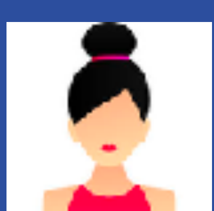
				
				1
	1	1	1	
			1	
	1			1

Watched it?

NETFLIX DATA

				
				4
	1	3	2	
			3	
	1			2

NETFLIX DATA

				
	?	?	?	4
	1	3	2	?
	?	?	3	?
	1	?	?	2

How do we fill the missing entries?

IDEAS

- **Items** that are similar to each other may get a similar rating from the same **User or User group**

CONTENT BASED: ITEM PROFILES

- **Item** needs to be represented first to compute similarity
  - Document (Metadata + TFIDF)
  - Video (Metadata + Description+ Video analysis)
  - Shopping item (Metadata + Descriptions)

CONTENT BASED: ITEM PROFILES

- Compute distance based similarity between items
  - Cosine
  - Eucledian
  - IR techniques (search)
- Calculate rating
  - Weighted sum

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2}$$

ITEM PROFILES: MOVIE

ITEM ID	LANGUAGE	MOVIE NAME	…….	ACTORS
1	HINDI	BAHUBALI	….	PRABHAS
2	ENGLISH	FORESTGUMP	…	TOM HANKS
3	ENGLISH	DORA	…	…
4	ENGLISH	IRONMAN	…	…

Data is the key

- Movie Metadata
- Combine external data (scrape from IMDB?)
- Combine reviews
- Combine social media data

CONTENT BASED

- Pros
  - Does not require ratings to make recommendation
  - Works well for news articles (no ratings; articles that change everyday)
  - Transparency of recommendations
- Cons
  - Staleness; limited novelty problem
  - Works for items with rich content

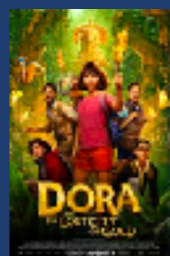
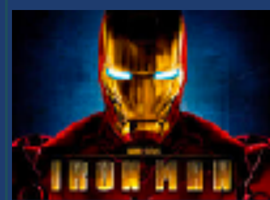
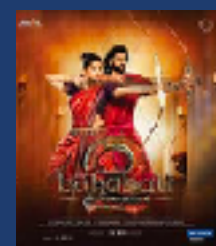
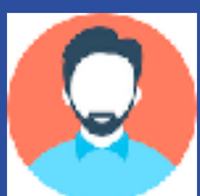
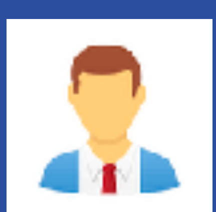
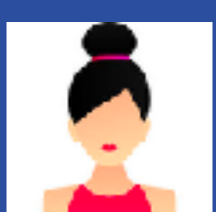
IDEAS

- **Item** that are similar to each other may get a similar rating from the same **User or User group**
- **User** that is similar to another user may provide similar ratings to the **Item**

A USER PROFILE?

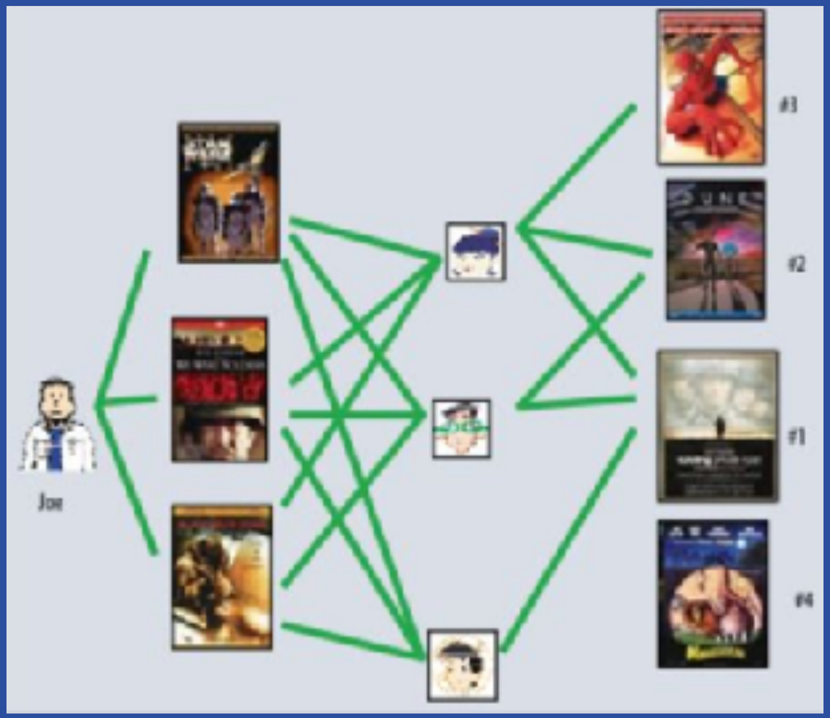
- **User** needs to be represented first to compute similarity
  - Demographics
  - Viewing patterns
  - Movie ratings
  - ….
- Challenges
  - Lack of data
  - Privacy

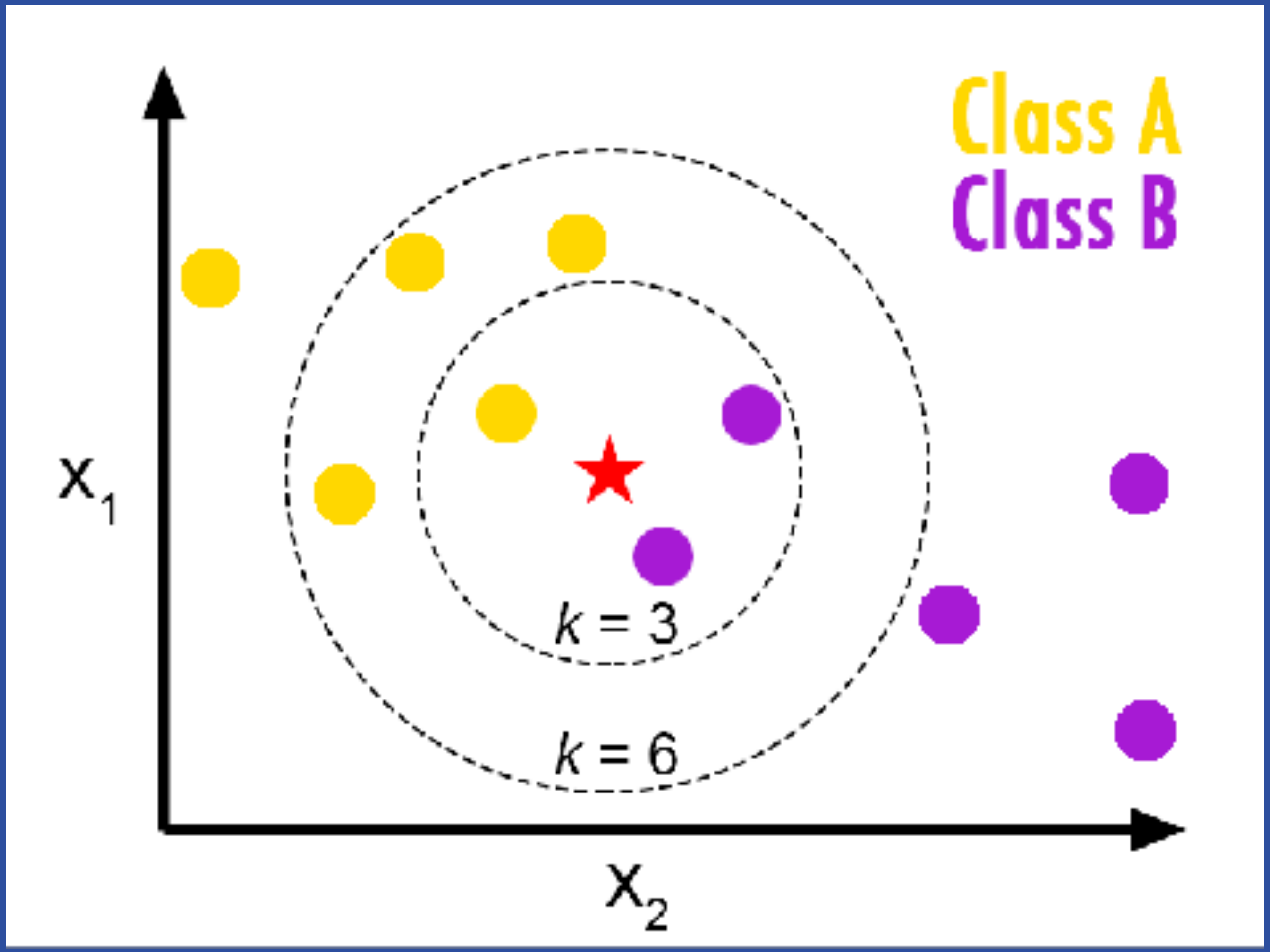
USER PROFILE

				
				4
	1	3	2	
			3	
	1			2

U1 : {0,0,0,4}
U2 : {1,3,2,0}
U3 : {0,0,3,0}
U4 : {1,0,0,2}

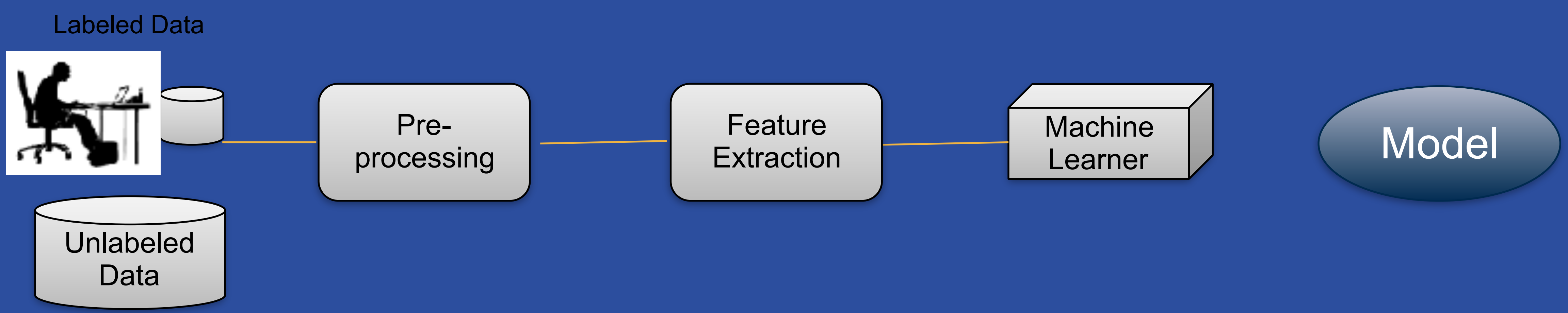
MEMORY METHOD: USER BASED CF



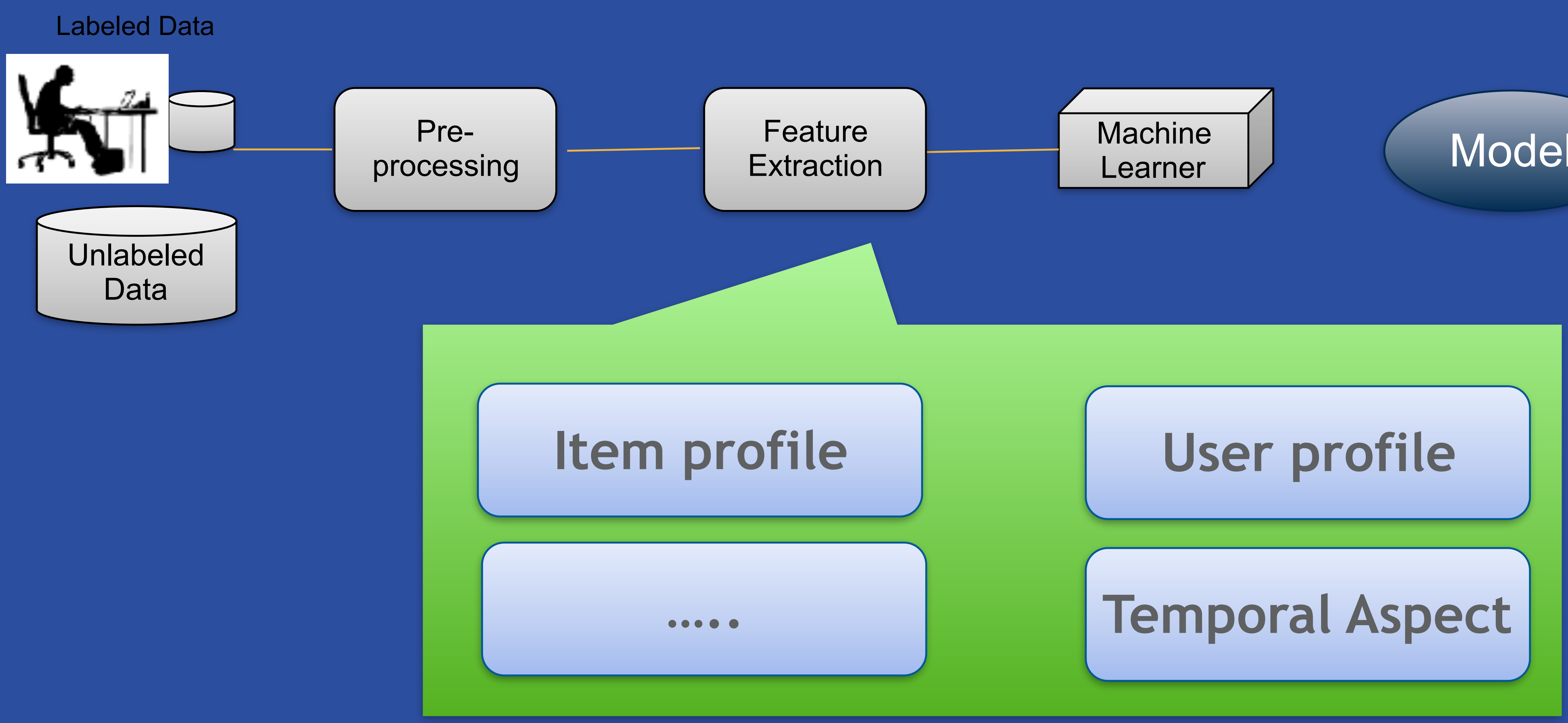


k-Nearest Neighbors Algorithm

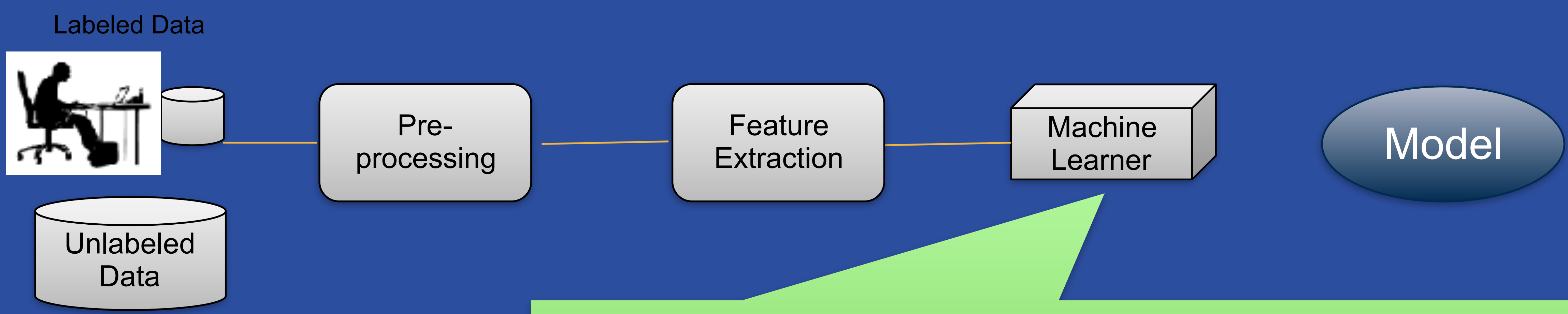
HYBRID APPROACH: RECOMMENDATION AS RATING CLASSIFICATION



HYBRID APPROACH: RECOMMENDATION AS RATING CLASSIFICATION



HYBRID APPROACH: RECOMMENDATION AS RATING CLASSIFICATION



Logistic Regression

$$\ln\left(\frac{P}{1-P}\right) = a + bX$$

$$\frac{P}{1-P} = e^{a+bX}$$

$$P = \frac{e^{a-bX}}{1+e^{a+bX}}$$

Maximum Entropy

$$Pr(c_i|t) = \frac{1}{Z(t)} exp\left(\sum_{j=1}^{n} \lambda_j f_{ij}(c_i, t)\right)$$

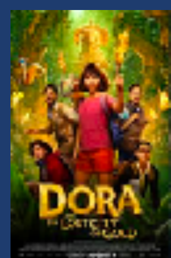
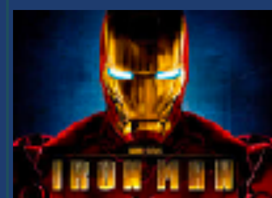
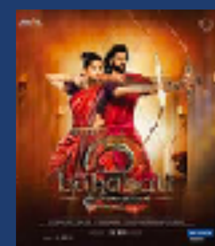
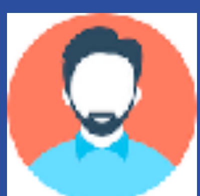
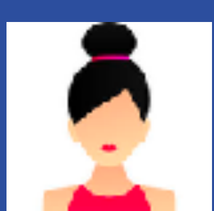
CHALLENGES

- Data Sparsity hinders similarity computation
  - Few rating pairs: <user, item> -> rating {1.. 5}
- Cold start problem
  - Need lots of ratings from each user before providing any meaningful recommendation
- Only feasible when the set of users is small (Netflix has 100M+ users today
- Or when the set of Items is small (Millions of movies vs. Genres of movies)

IDEAS

- **Item** that are similar to each other may get a similar rating from the same **User or User group**
- **User** that is similar to another user may provide similar ratings to the same **Item**
- **User** and **Item** interactions are latent and need to be modeled ("users who liked this also liked")

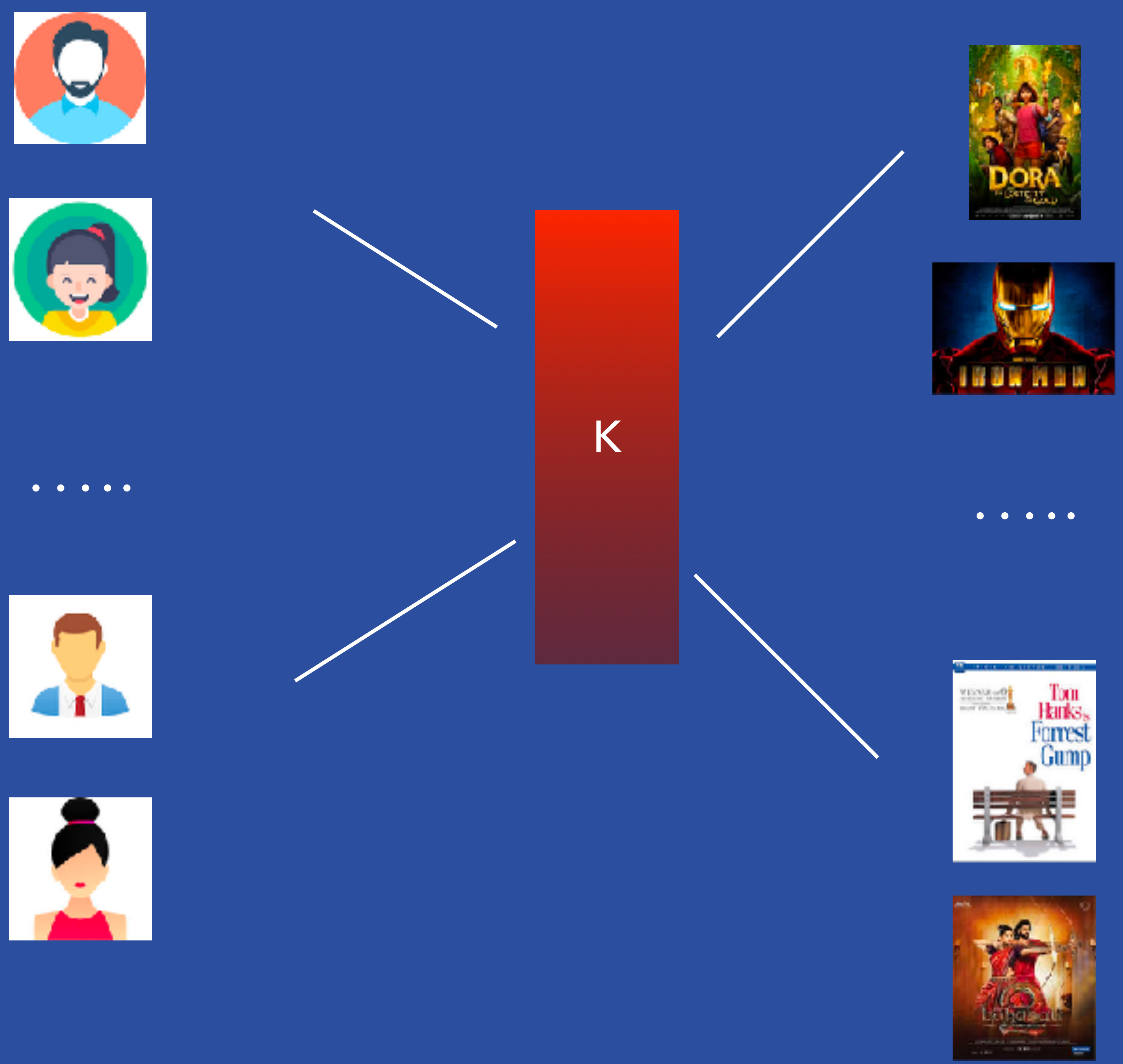
RECOMMENDATION USING COLLABORATIVE FILTERING MATRIX

				
	?	?	?	4
	1	?	2	?
	?	?	3	?
	1	?	?	2

Problem: Data Sparsity is a real problem for most domains

Solution: Reduce dimensionality of the data

LATENT DIMENSIONS FOR CF MATRIX



Movies map into a set of hidden dimension/topics

Users have preferences for each of those dimensions/topics

SINGULAR VALUE DECOMPOSITION: TRAINING

- X matrix is factored/de-composed into U , S, V matrices
- such that S is the singular values of matrix X ; with rank 'r'

$$
\underset{m \times n}{\overset{\hat{X}}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} \approx \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}
$$

SINGULAR VALUE DECOMPOSITION: PREDICTION

- X_hat matrix is re-generated from U , S, V matrices from previous training step

$$\underset{m \times n}{\overset{\hat{X}}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} \approx \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

- X_hat is filled with average ratings initially and upon computation of SVD, the corresponding cells have predicted ratings

OTHER MATRIX FACTORIZATION METHODS

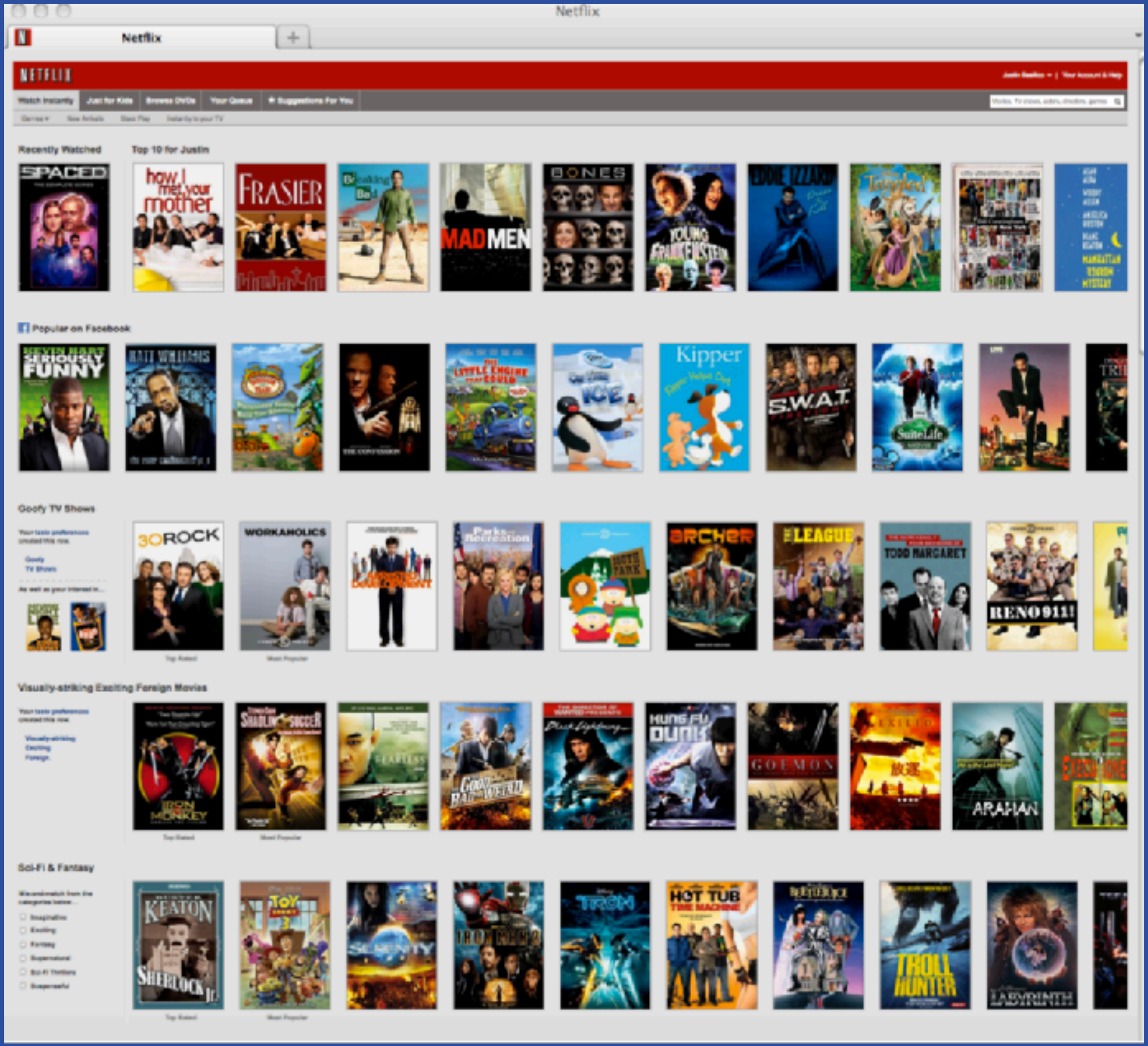
- SVD++ (incremental and iterative variant of SVD)
- Non-negative Matrix Factorization (for sparse matrices)
- Latent Semantic Analysis/ LDA/ PCA
- Hybrid recommenders combine regression + dimensionality reduction techniques

4. CHALLENGES IN PRODUCTIZING RECOMMENDATION ENGINES

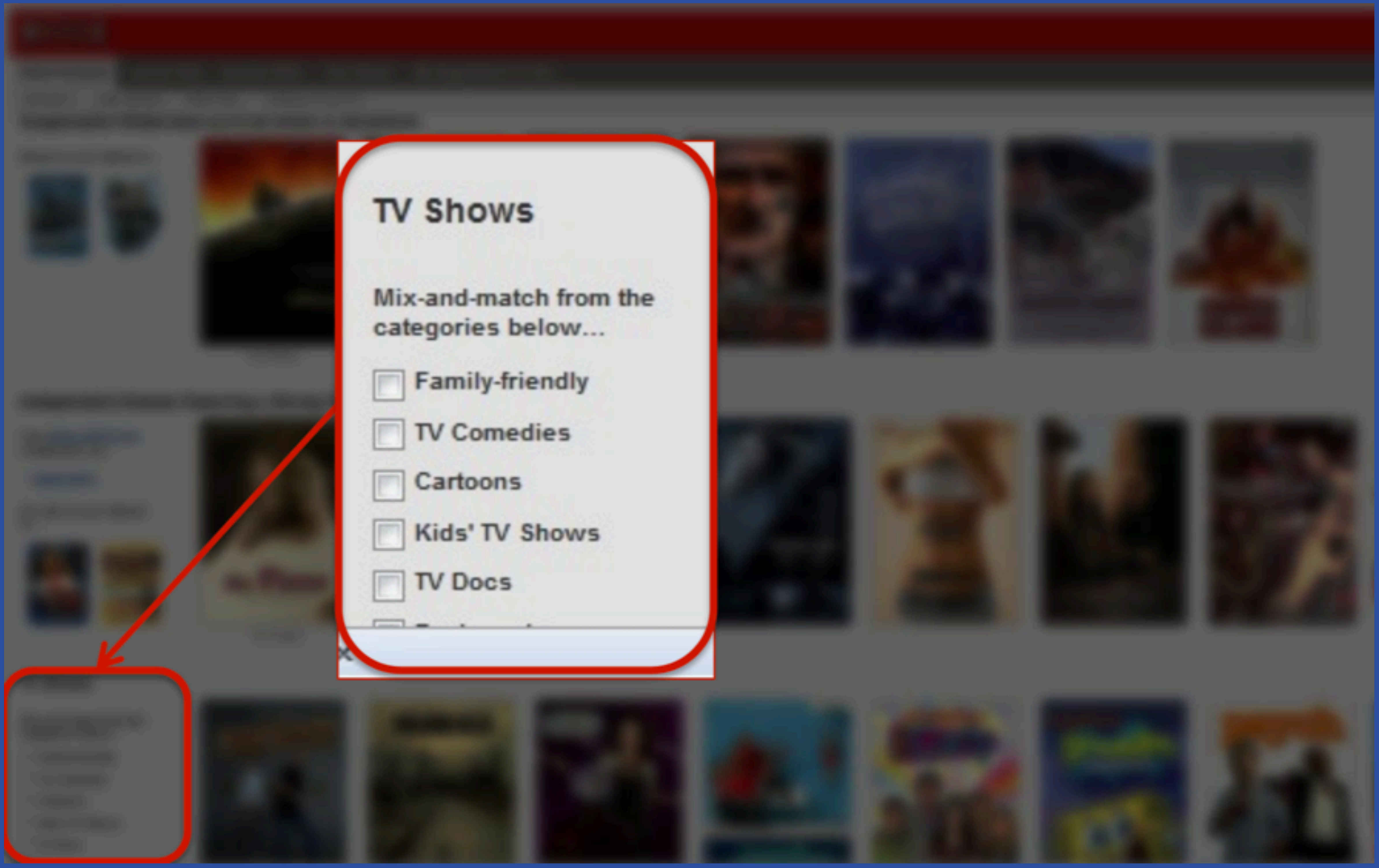
EXPLICIT VS. IMPLICIT FEEDBACK

- What if I do not rate a movie? - Implicit feedback
  - User viewership data
  - Does watching for an hour mean I like it?
  - Does skipping in between mean I don’t like it?
  - Does resuming a movie mean I like it?
  - Does replaying a movie mean I love it?
- Other data
  - Social data
  - Diversity
  - Freshness
  - Popularity

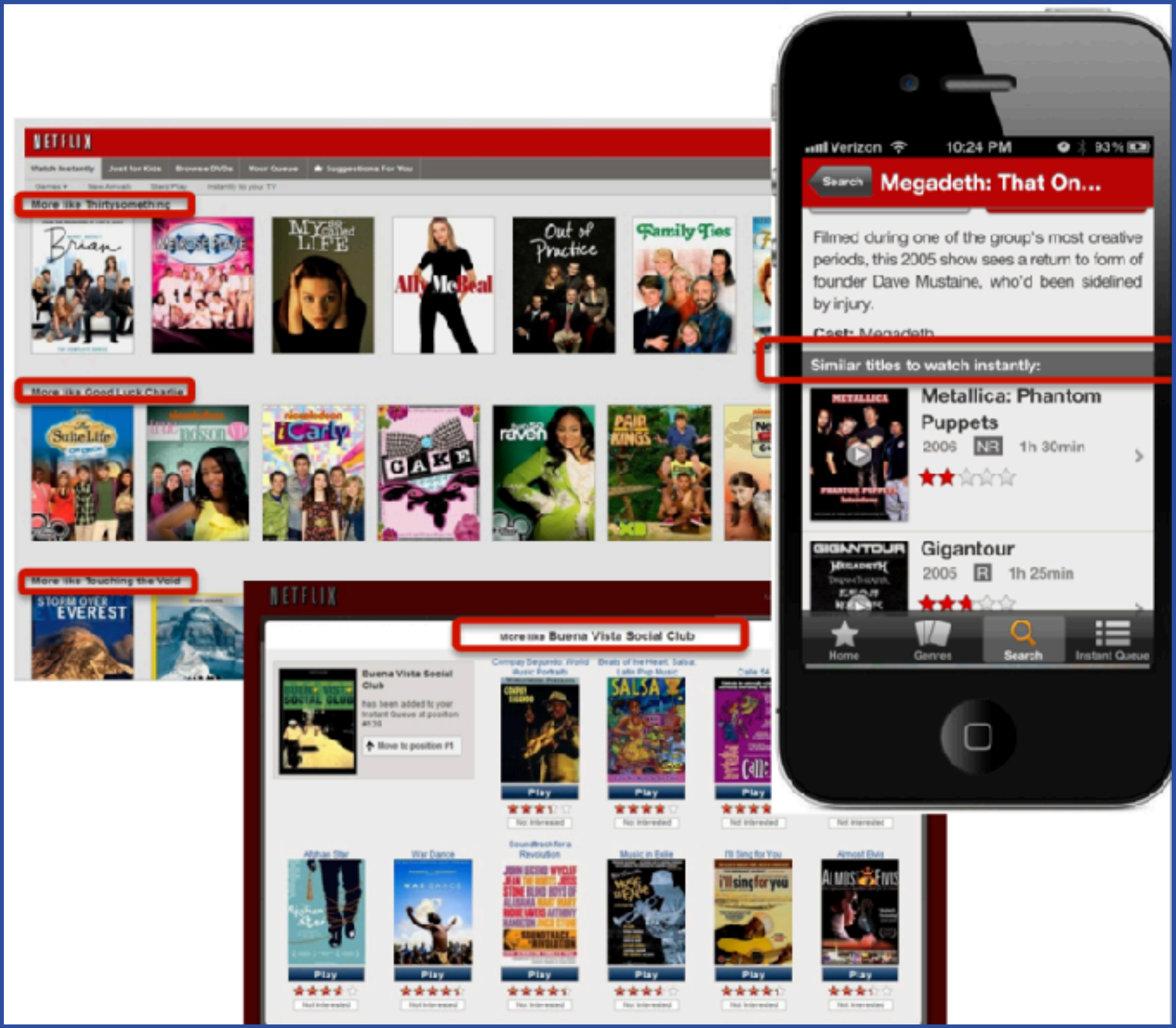
PERSONALIZATION



COMBINING USER META INPUT VS. RATINGS

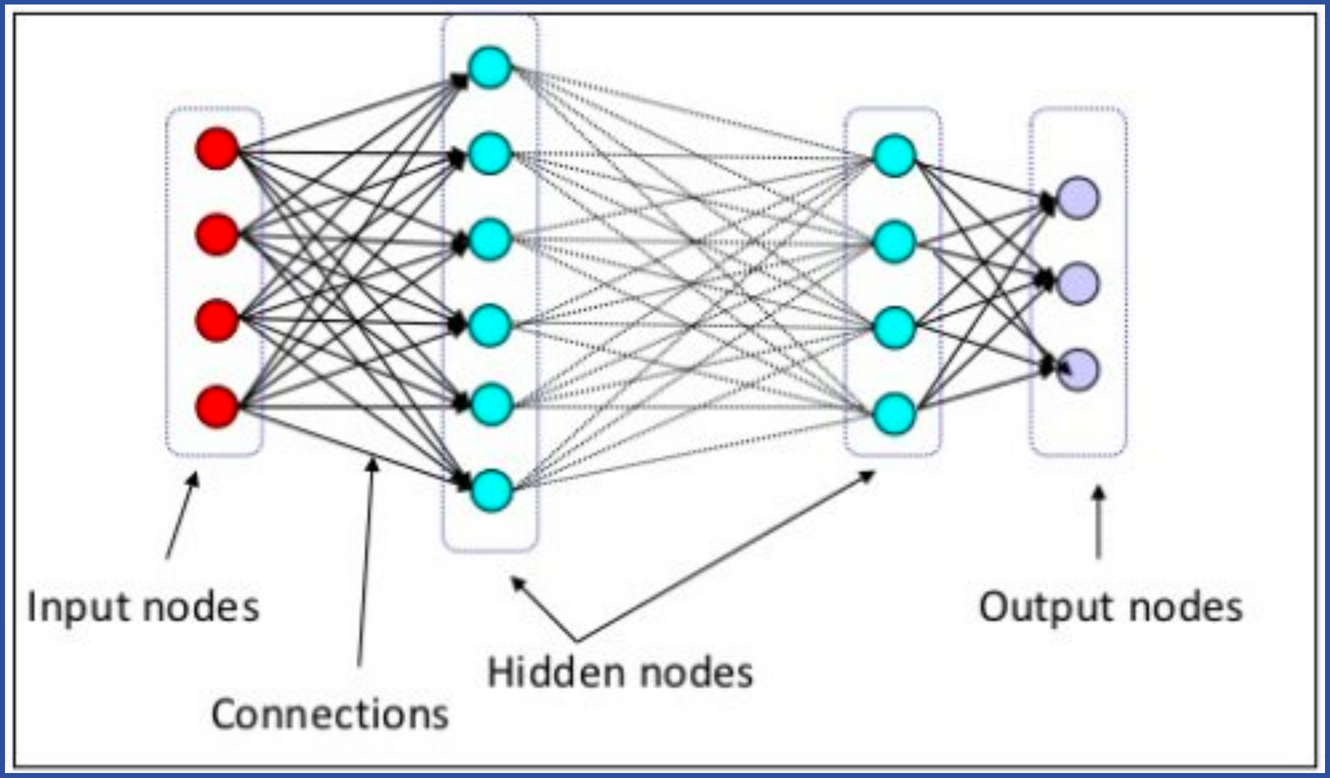


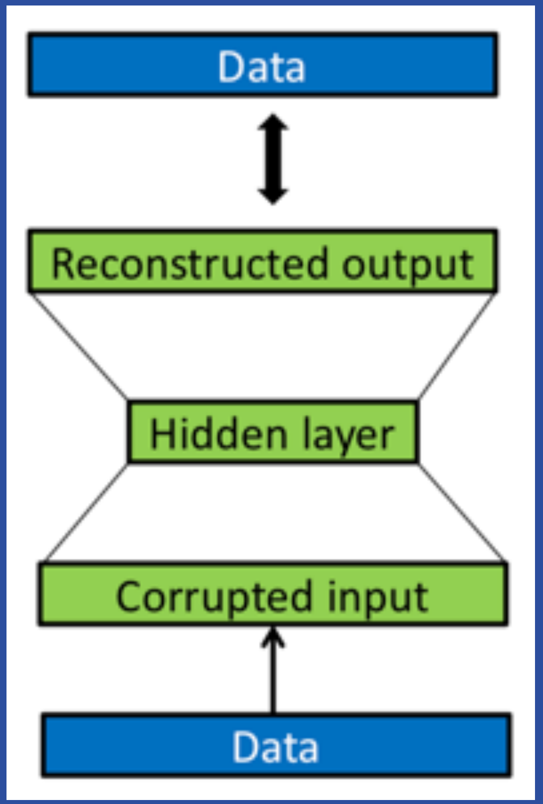
EXPLAINING RECOMMENDATIONS



DEEP LEARNING BASED METHODS

- ML algorithms that are modeled after the way human brains work - network of neurons (1970s - now)
- Prod2vec - Similar to word2vec models but using "item vectors" instead of words
- Restricted Boltzman Machines ; Deep Boltzman Machines ; Autoencoders

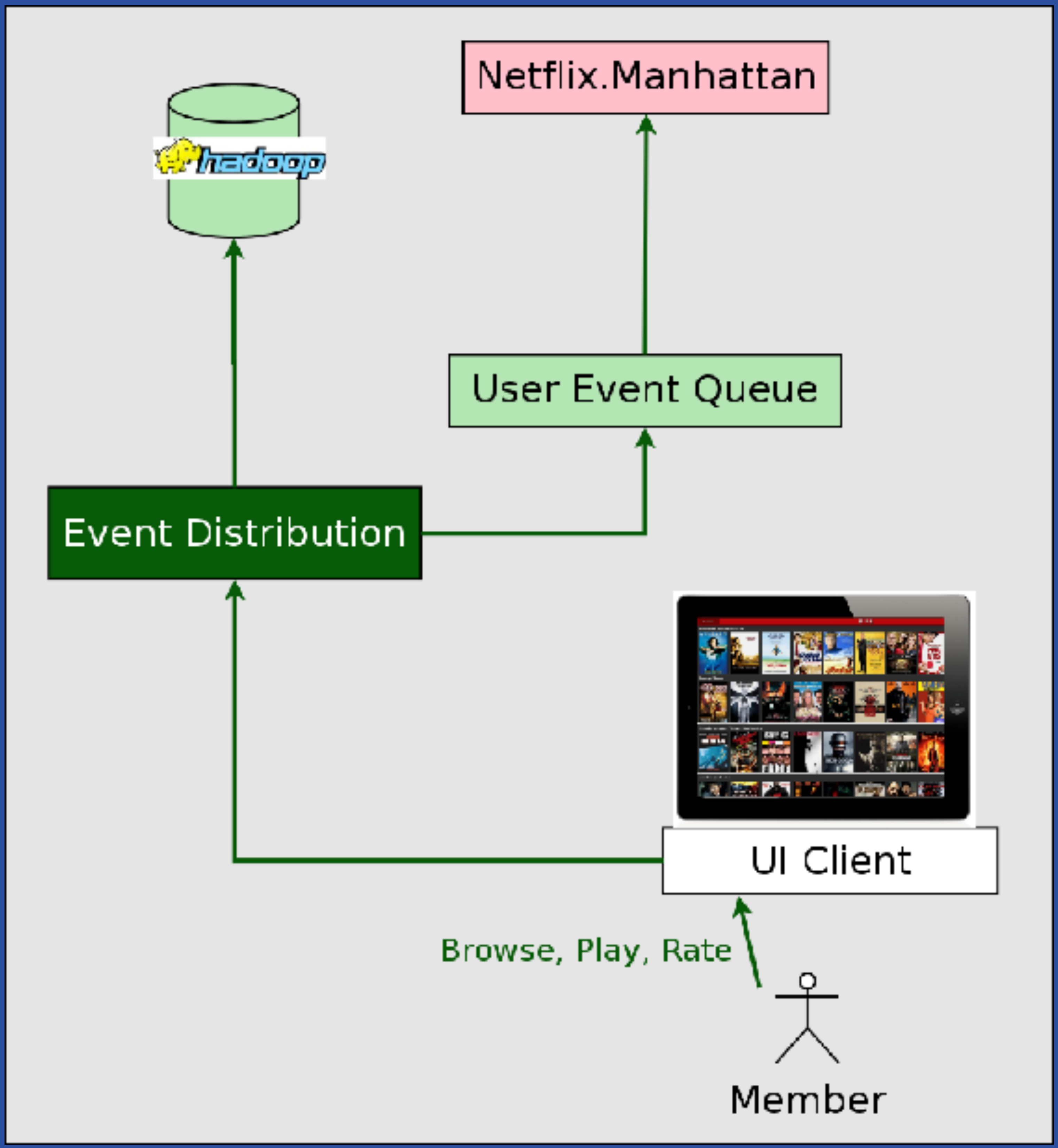




HYBRID TECHNIQUES FOR PRACTICAL SYSTEMS

- Phase 1: Content based approaches
  - Do not depend upon ratings
- Phase 2: Item-based approach
  - Use ratings as they come in
- Phase 3: Ensembles
- Phase 4: Deep learning approaches

STATIC VS. DYNAMIC DATA



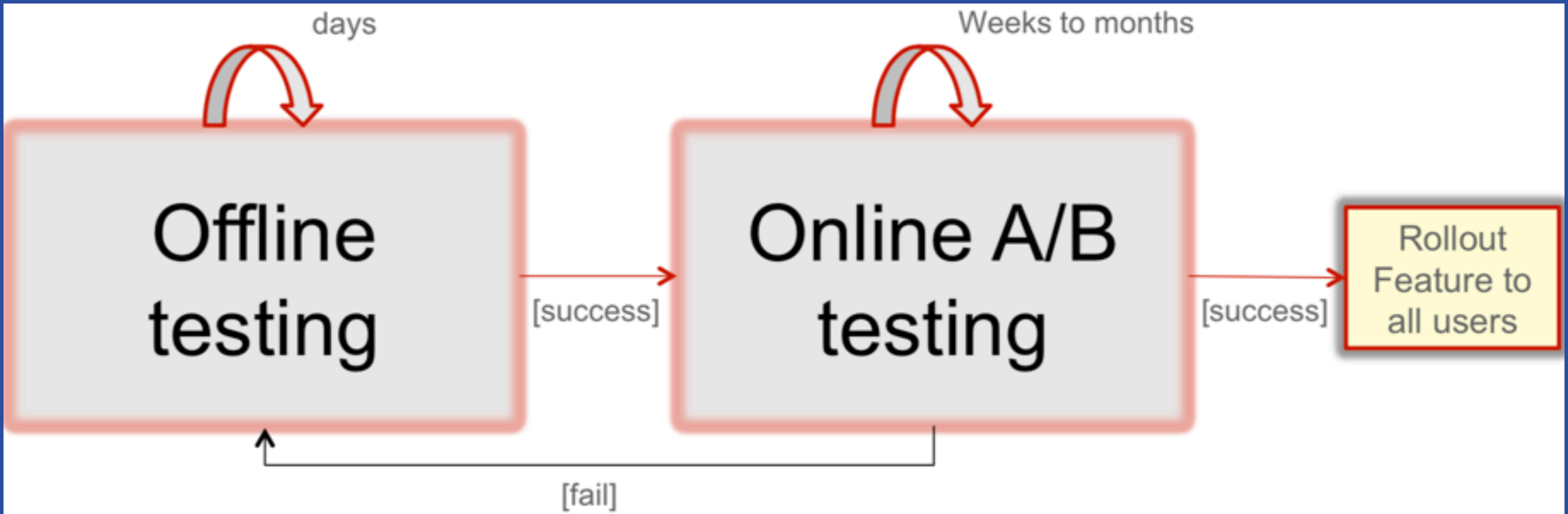
- Viewing patterns are dynamic and needs real-time data
- Movie data is static and updates less frequently

EVALUATION OF RECOMMENDER SYSTEMS

- How do you evaluate for performance of serendipity
- Classification / Regression evaluation - what is your rating on a scale of 1-5?
- Ranking evaluation - Do you like A over B ?
- Task based evaluation
  - Did you watch that movie?
  - Did you make the recommended purchase
- Short term vs. Long term metrics
  - Engagement scores vs Life time value

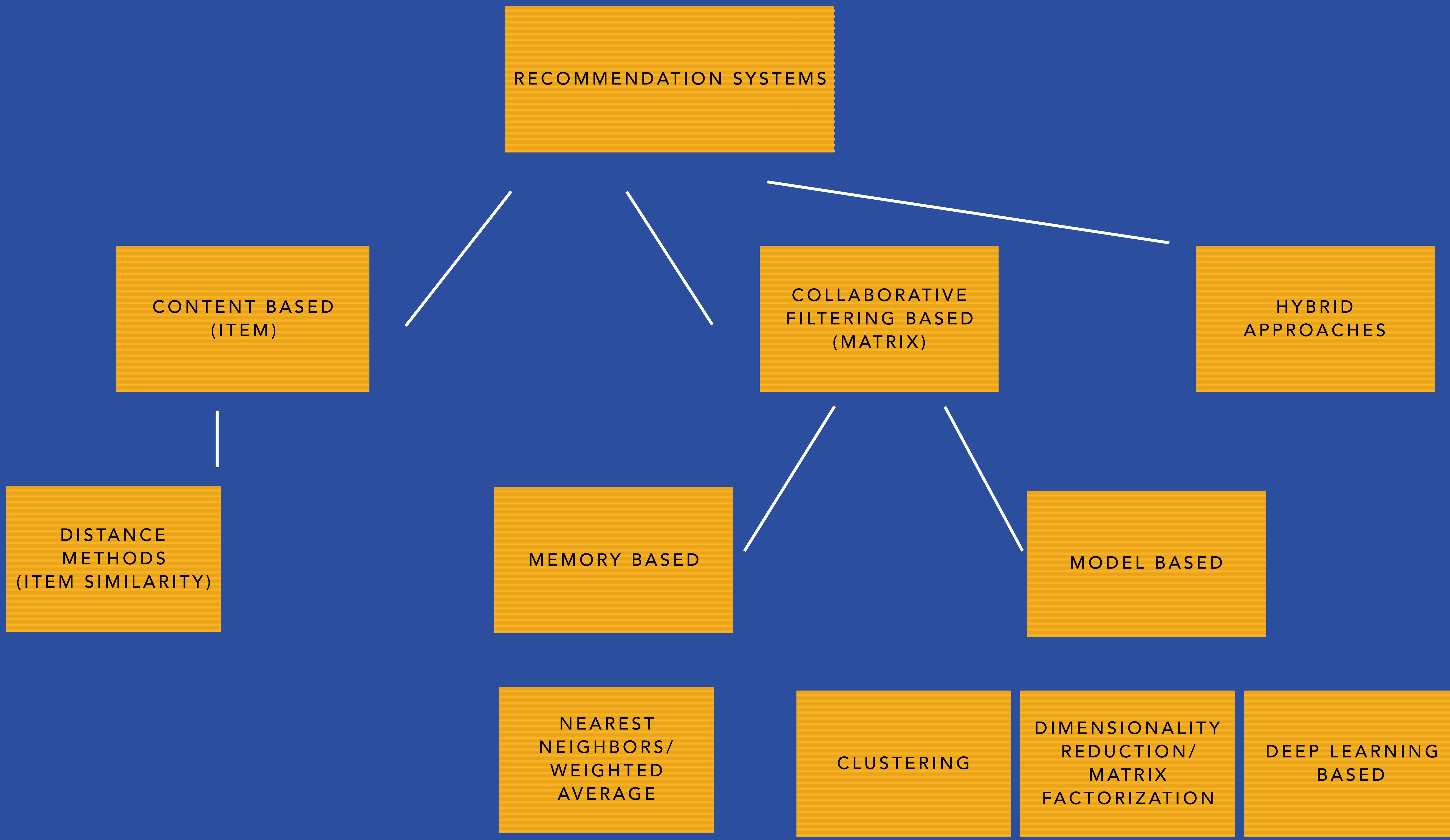
ROLLING OUT NEW ALGORITHMS

- Offline testing
- A/B Testing
- Phased rollouts



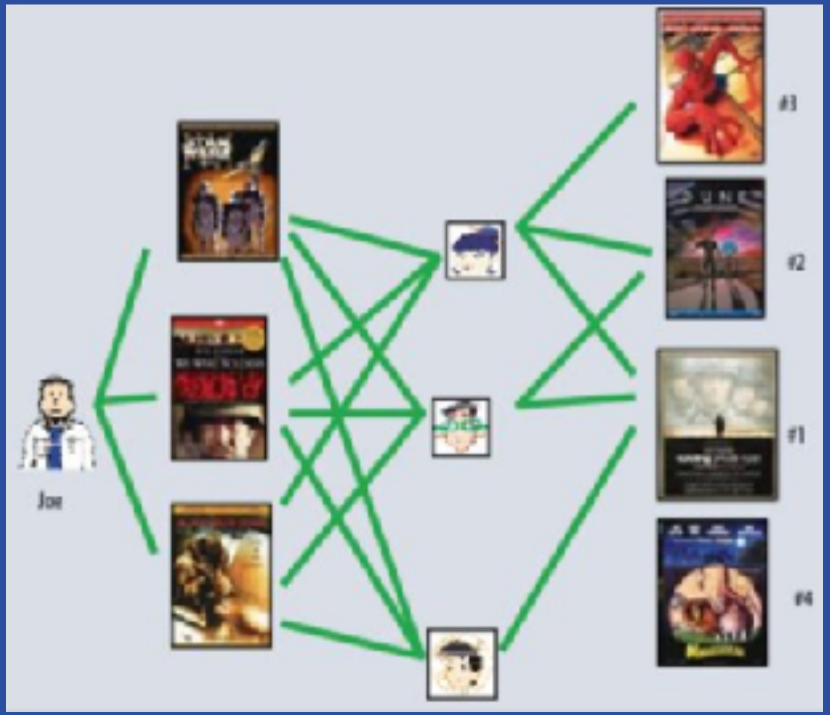
SUMMARY

RECOMMENDER SYSTEMS

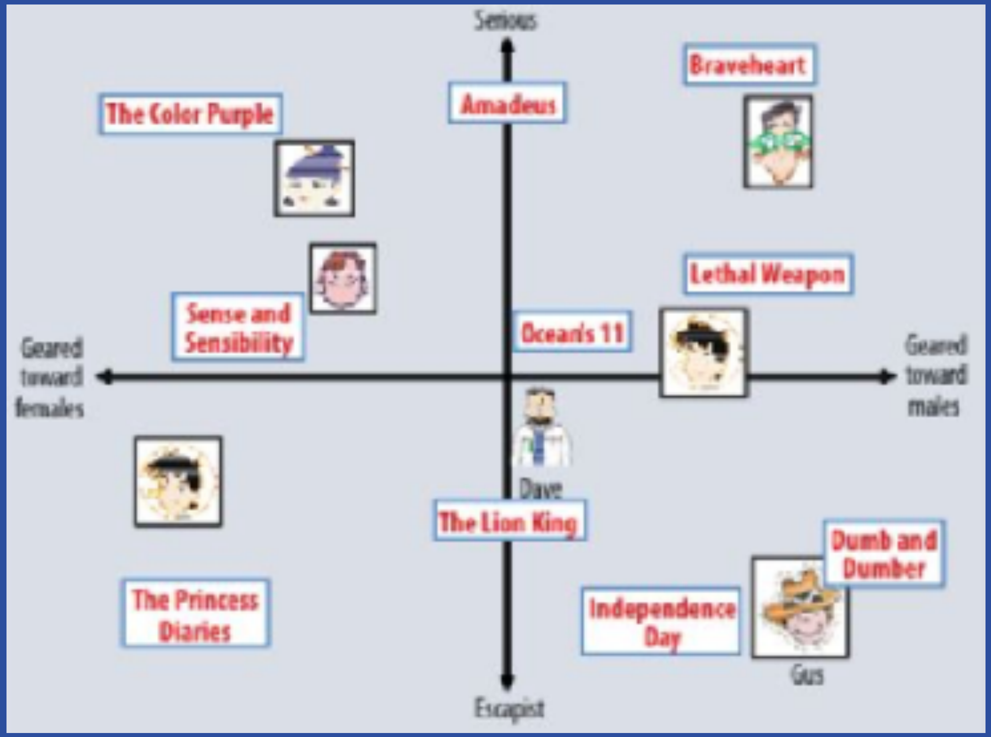


MEMORY VS MODEL BASED APPROACHES

- Memory Based Approach
  - Huge memory required
  - Does not scale for large item inventories
  - Adaptable to changes (new movie updated etc) as no offline training



- Model Based Approach
  - Conducive to low memory scenarios
  - Scales well for large user or item sets
  - Not adaptable to changes (needs to be retrained offline)



FROM HERE

- Open source libraries
  - Python Surprise - http://surpriselib.com/
  - Amazon DSSTNE
  - Tensorflow - https://cloud.google.com/solutions/machine-learning/recommendation-system-tensorflow-overview
- Datasets
  - Netflix Challenge dataset
  - MovieLens dataset
  - Amazon dataset