Recommender Systems

Data Science Dojo

datasciencedojo
data science for everyone

Overview

- Introduction
  - Collaborative vs Content-based
- How do they work?
  - Data structure
  - Ranking by similarity
  - Predicting
  - Evaluation
- Advantages/Disadvantages
- Example using Azure ML

INTRODUCTION

data science for everyone

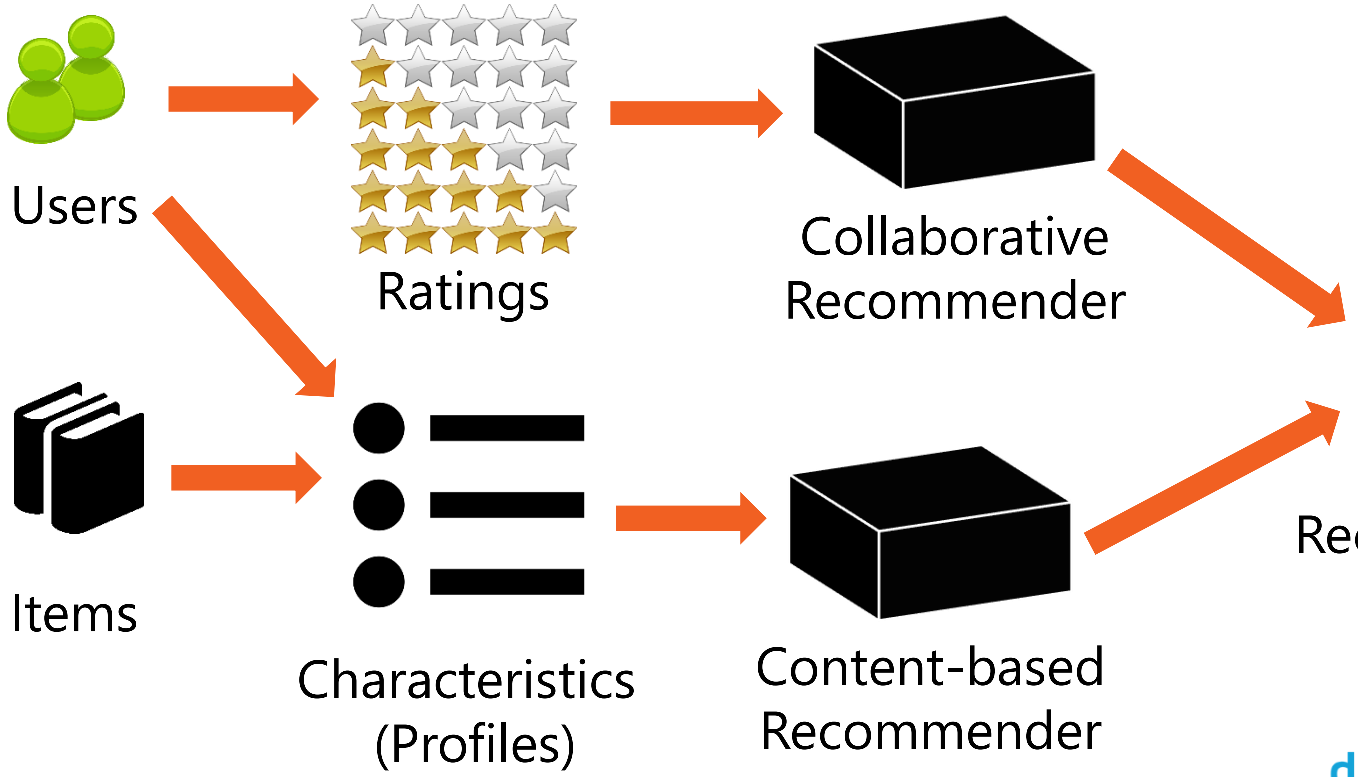
Recommendation Systems

- What are Recommendation Systems?
  - Automated systems to filter and recommend products based on users' interest and taste.
  - Designed to solve the information overload problem

Why recommendation systems?

- For Customers
  - Narrow down the set of choices
  - Discover new, interesting things
  - Save time
- For Business
  - Increase the number of items sold
  - Sell more diverse items
  - Better understand what the user wants

Two Types of Recommenders



Item	Score
i_1	0.6
i_2	0.8
i_3	0.3

Recommendations

Two Types of Recommenders

Collaborative

- 'Give me items that **people like me** enjoy'
- Wisdom of the crowds
- Widely applicable

Content-Based

- 'Give me items similar to **items I like**'
- Content analysis based
- Related to Information Retrieval

Two Types of Recommenders

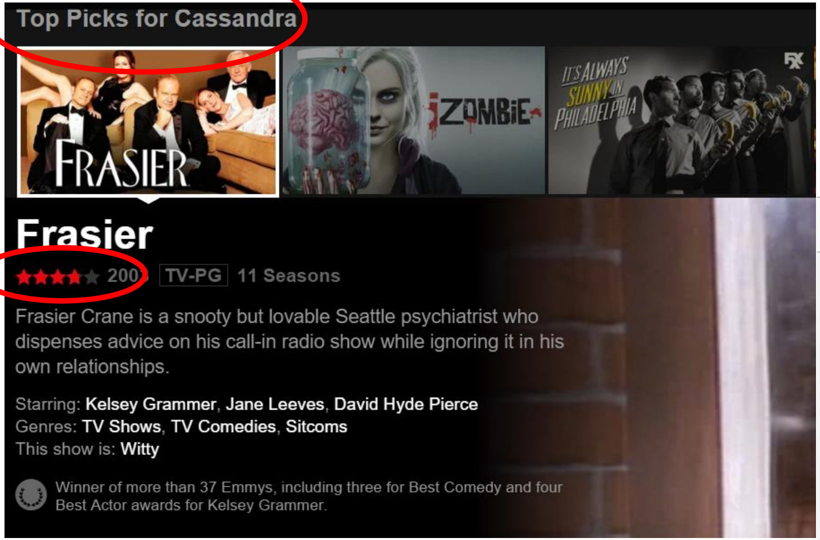
Collaborative

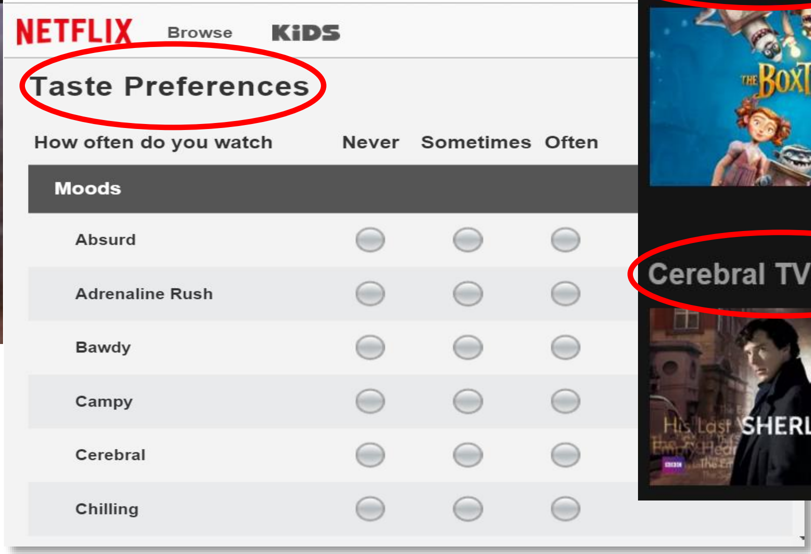
- Users, Items, & Ratings
- Use Ratings of similar Users to recommend unseen Items

Content-Based

- User & Item profiles
- Use overlap of User and Item characteristics to recommend unseen items

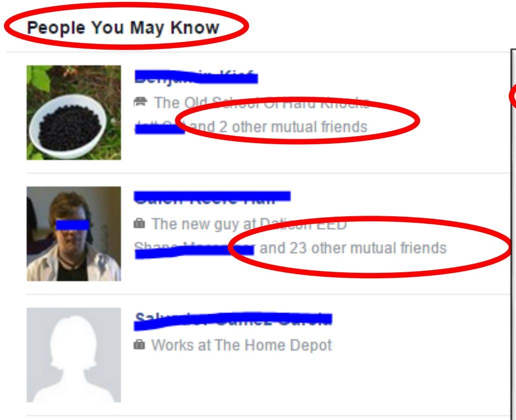
Example: Netflix

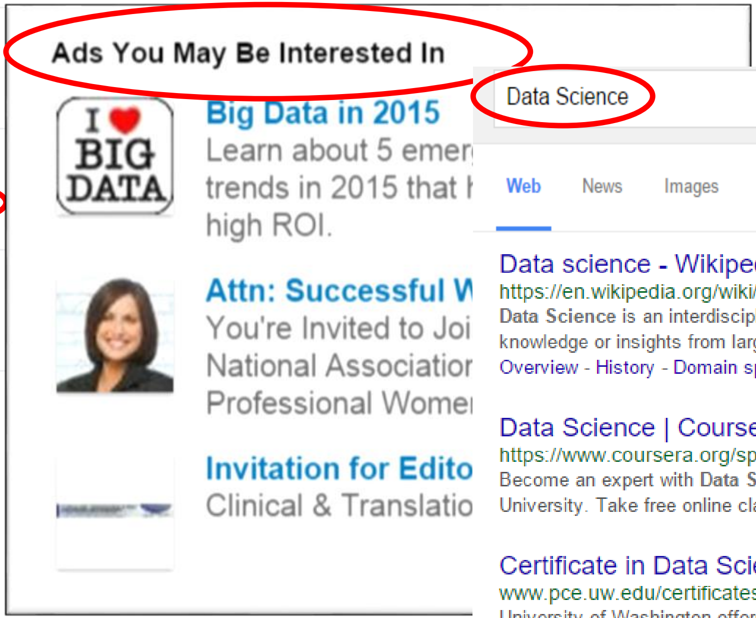


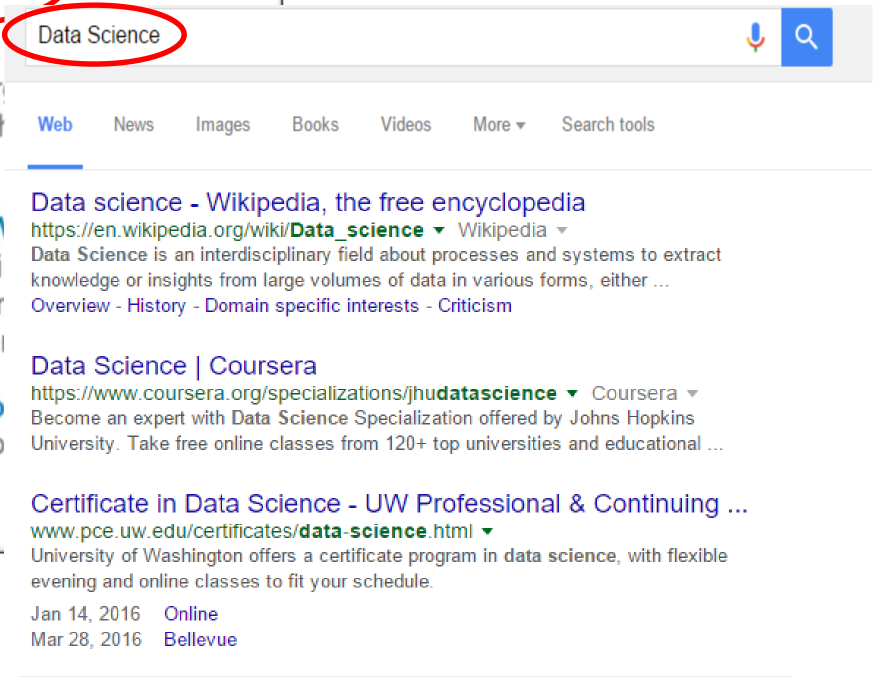


datasciencedojo
data science for everyone

Example: Social Media & Search

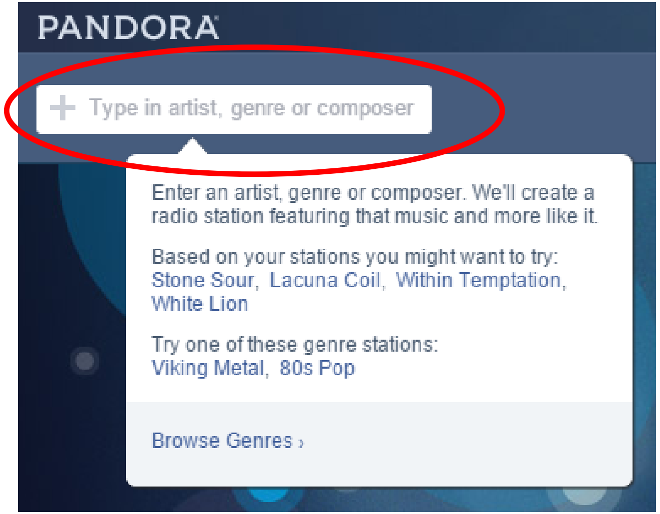


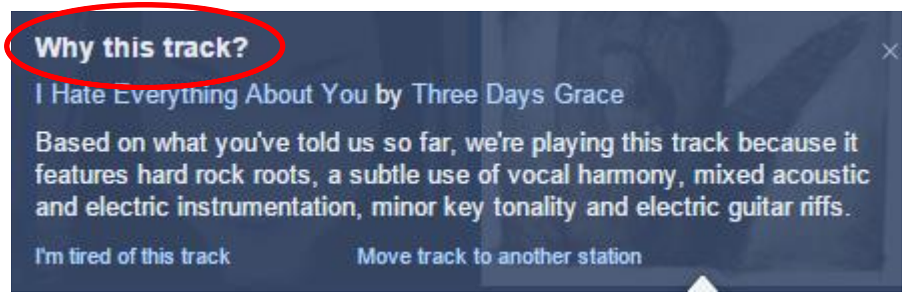


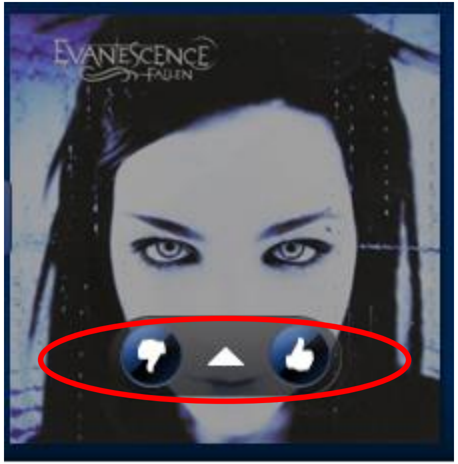


datasciencedojo
data science for everyone

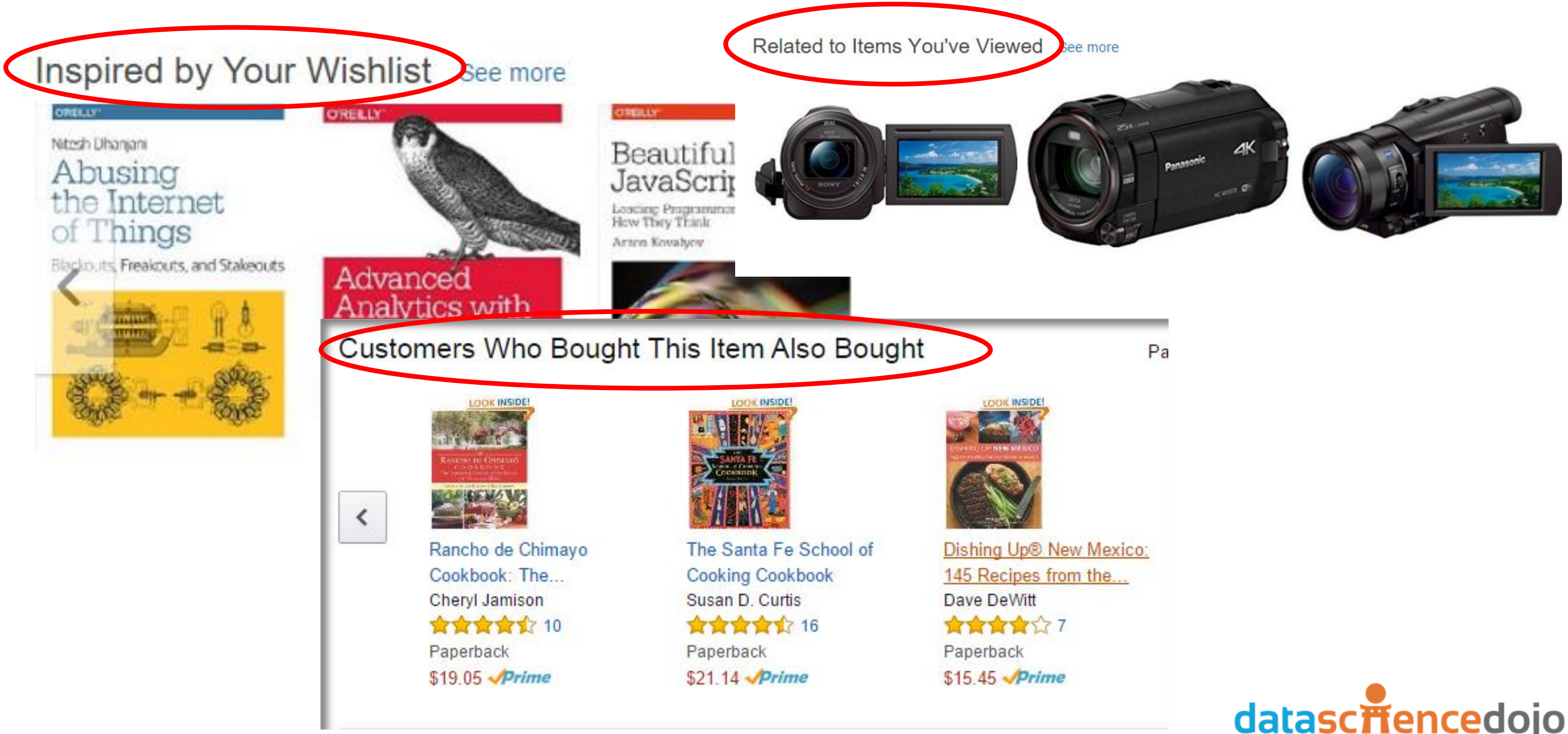
Example: Pandora







Example: Amazon



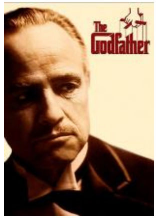
datasciencedojo
data science for everyone

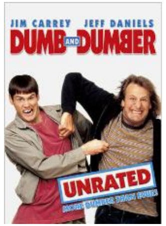
DATA STRUCTURE

Data Structure

- What kind of data?
  - Collaborative
    - Ratings of Items by Users
  - Content-based
    - Characteristic profiles of Users and Items

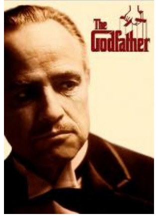
Data Structure - Collaborative

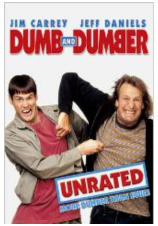




Alice	5	3	4	4	?
Bob	3	1	2	3	3
Chris	4	3	4	3	5
Donna	3	3	1	5	4
Evi	1	5	5	2	1

Data Structure – Content-based





Item/User	Drama?	Comedy?	Adventure?	Romance?
The Godfather	5	1	2	1
Titanic	4	3	2	5
Lord of the Rings	4	2	5	1
Dumb & Dumber	1	5	2	2
Spirited Away	5	3	5	2
Alice	5	4	1	4
Bob	3	1	1	1
Chris	4	2	5	2

Content-based: User Profiles

- **User Provided**
  - Ask for preferences
  - Needs accounts
  - Often low completion rates
- **Automated Generation**
  - Cookies follow behavior
  - No user persistence (often)
  - Loss in translation

Content-based: Item Profiles

- **Expert Labeling**
  - Assign keywords based on content
  - May be provided by creators/distributors
  - Crowd sourcing?
- **Automated Indexing**
  - Used for text documents
  - Based on word content of document set
  - No expert knowledge involved

SIMILARITY

Similarity Measurements

- Given two vectors $\vec{x}$ and $\vec{y}$ with n components each
  - Ratings of User x and User y
  - Ratings for Item x and Item y
  - Profiles of User x and Item y
- How similar are the Users/Items?

Similarity Measurements

- Pearson's Correlation

$$sim(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Cosine Similarity

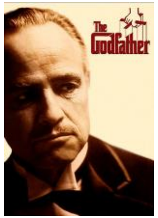
$$sim(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}| * |\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

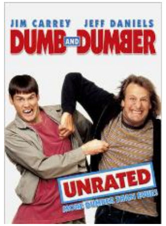
Collaborative: User-Based

- Goal: Predict User u's rating on a movie m they haven't seen
  - Find the N most similar Users to u who have seen m
  - Use their ratings to predict u's rating

Collaborative: User-based

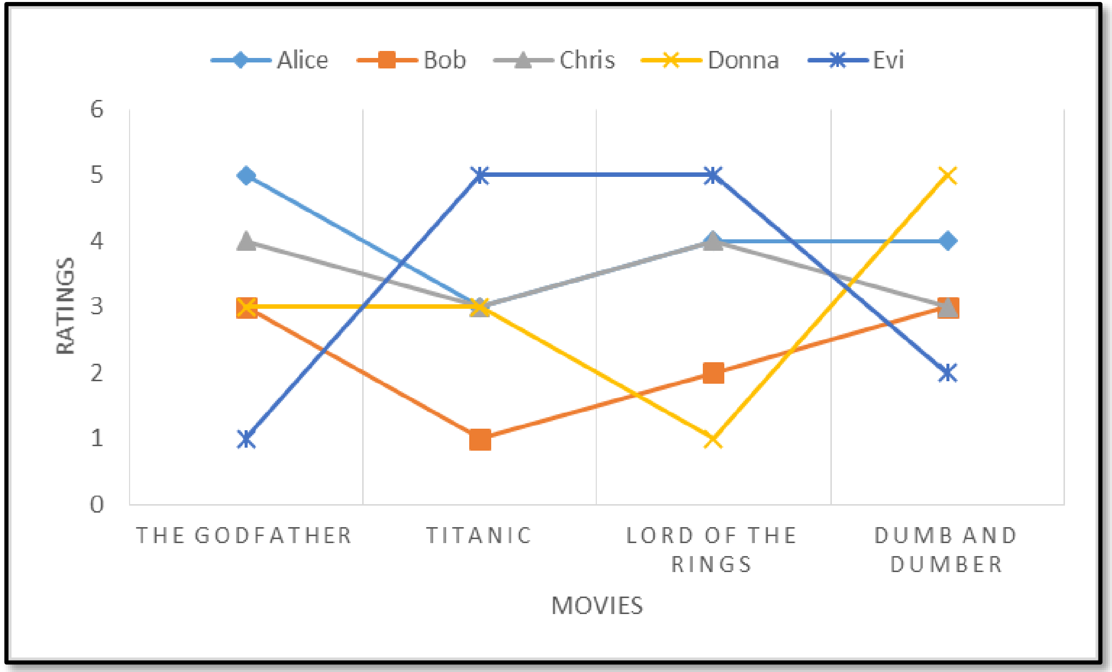
Which metric should we use?





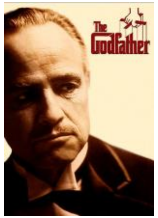
Alice	5	3	4	4	?	
Bob	3	1	2	3	3	sim = ?
Chris	4	3	4	3	5	sim = ?
Donna	3	3	1	5	4	sim = ?
Evi	1	5	5	2	1	sim = ?

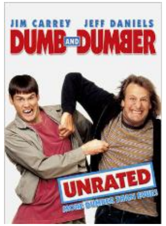
Collaborative: User-based



Collaborative: User-based

Pearson's correlation corrects for varied baselines





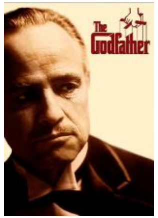
Alice	5	3	4	4	?	
Bob	3	1	2	3	3	sim=0.85
Chris	4	3	4	3	5	sim=0.90
Donna	3	3	1	5	4	sim=0.70
Evi	1	5	5	2	1	sim=0.79

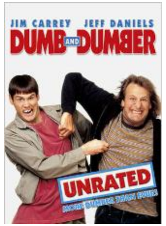
Collaborative: Item-based

- Alternate approach:
  - Use the similarity between items (and not users) to make predictions
  - Look for movies that are similar to movie m
  - Take **Alice**'s ratings for these items to predict the rating for movie m

Collaborative: Item-based

Which metric should we use?





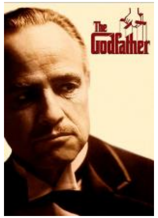
Alice	5	3	4	4	?
Bob	3	1	2	3	3
Chris	4	3	4	3	5
Donna	3	3	1	5	4
Evi	1	5	5	2	1

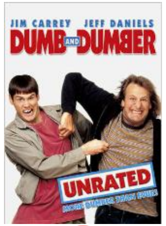
sim = ? sim = ? sim = ? sim = ?

datasciencedojo
data science for everyone

Collaborative: Item-based

Cosine similarity allows for objective good/bad





Alice	5	3	4	4	?
Bob	3	1	2	3	3
Chris	4	3	4	3	5
Donna	3	3	1	5	4
Evi	1	5	5	2	1

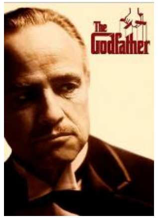
sim=0.99 sim=0.74 sim=0.72 sim=0.93

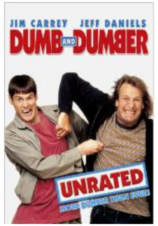
datasciencedojo
data science for everyone

Content-based: Similarity

- Goal: Return a recommendation list of items for each user
  - Find similarity of each User to each Item
  - Order Items by similarity

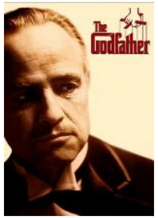
Content-based: Similarity

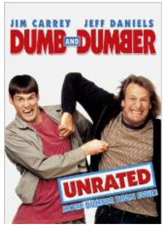




Item/User	Drama?	Comedy?	Adventure?	Romance?
The Godfather	5	1	2	1
Titanic	4	3	2	5
Lord of the Rings	4	2	5	1
Dumb & Dumber	1	5	2	2
Spirited Away	5	3	5	2
Alice	5	4	1	4
Bob	3	1	1	1
Chris	4	2	5	2

Content-based: Similarity





Alice	0.83	0.96	0.72	0.79	0.83
Bob	0.99	0.86	0.85	0.59	0.91
Chris	0.87	0.82	0.99	0.69	0.99

- Cosine similarity doesn't erase baselines
- Predict order, not exact rating

PREDICTIONS

Collaborative: Predictions

- Use "Aggregation Function"
- Choose N nearest neighbors to User u
- Combine each neighbor j's rating on Item i ($r_{j,i}$)
- Simple
  - $r_{u,i} = \frac{1}{N}\sum_{j=1}^{N} r_{j,i}$
- Weighted & Centered
  - $r_{u,i} = \overline{\boldsymbol{r}_u} + \alpha\sum_{j=1}^{N} sim(j,u)(r_{j,i} - \overline{r_j})$

Content-based: Predictions

- Simple
  - Rank in order of similarity
- Information retrieval techniques
  - Well studied, wide diversity of models
  - Classification algorithms

EVALUATION

Evaluating Recommendation

- **Mean Absolute Error (*MAE*)** computes the deviation between predicted ratings and actual ratings

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

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

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

Metrics

- Order matters, not exact rating value
- Graded Relevance
  - Have humans assign scores to possible results
  - Ideal results will be ordered by relevance, high to low
- Discounted cumulative gain (DCG)
  - Logarithmic reduction factor

$$DCG_N = rel_1 + \sum_{i=2}^{N} \frac{rel_i}{\log_2 i}$$

Where:

- N is the length of the recommendation list
- rel_i returns the relevance of recommendation at position i

Metrics

- **Ideal discounted cumulative gain (IDCG)**
  - DCG value when items are ordered perfectly

$$IDCG_N = rel_1 + \sum_{i=2}^{N} \frac{rel_i}{\log_2 i}$$

- **Normalized discounted cumulative gain (nDCG)**

$$nDCG_{pos} = \frac{DCG_{pos}}{IDCG_{pos}}$$

  - Normalized to the interval [0..1]

ADVANTAGES/DISADVANTAGES

Advantages

Collaborative

- Wide applicability
- Serendipity
- Simple

Content-based

- No community needed
- Transparency
- Good cold start

Disadvantages

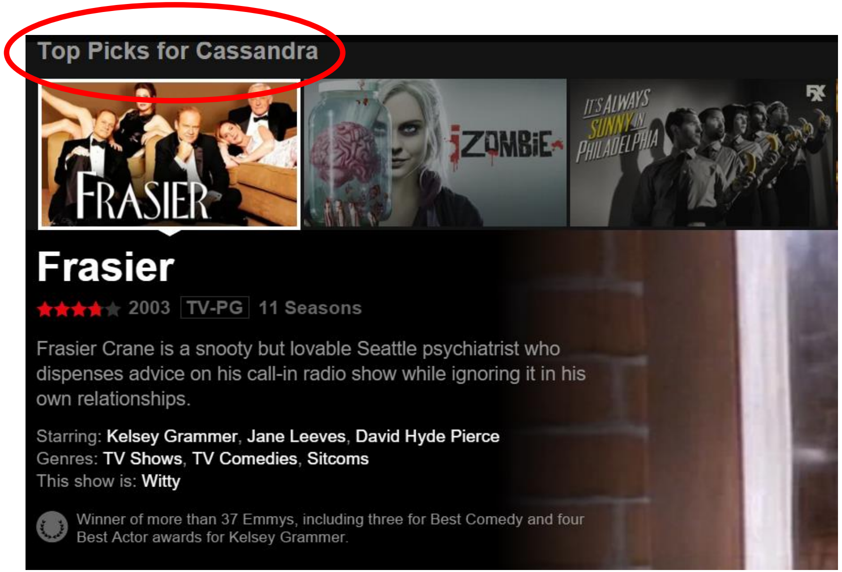
Collaborative

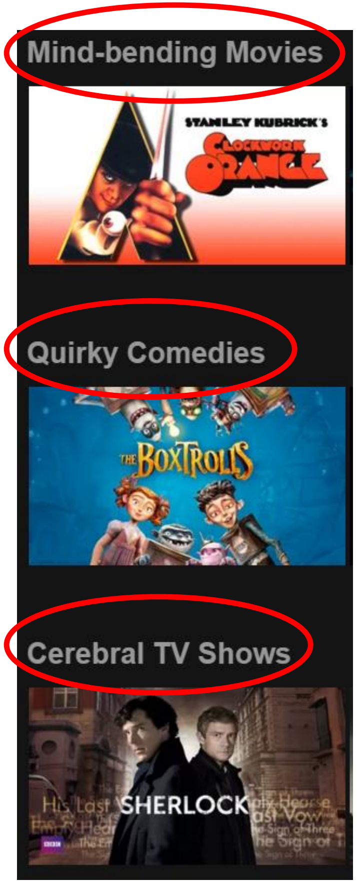
- Poor cold start
- Grey Sheep
  - Shared accounts
- Shilling
- Poor scaling

Content-based

- Limited profiles
  - New users
  - Cost of expert labeling
- Over-specialization
  - Lack of diversity

Back to Netflix





datasciencedojo
data science for everyone

QUESTIONS

EXAMPLE USING AZURE ML