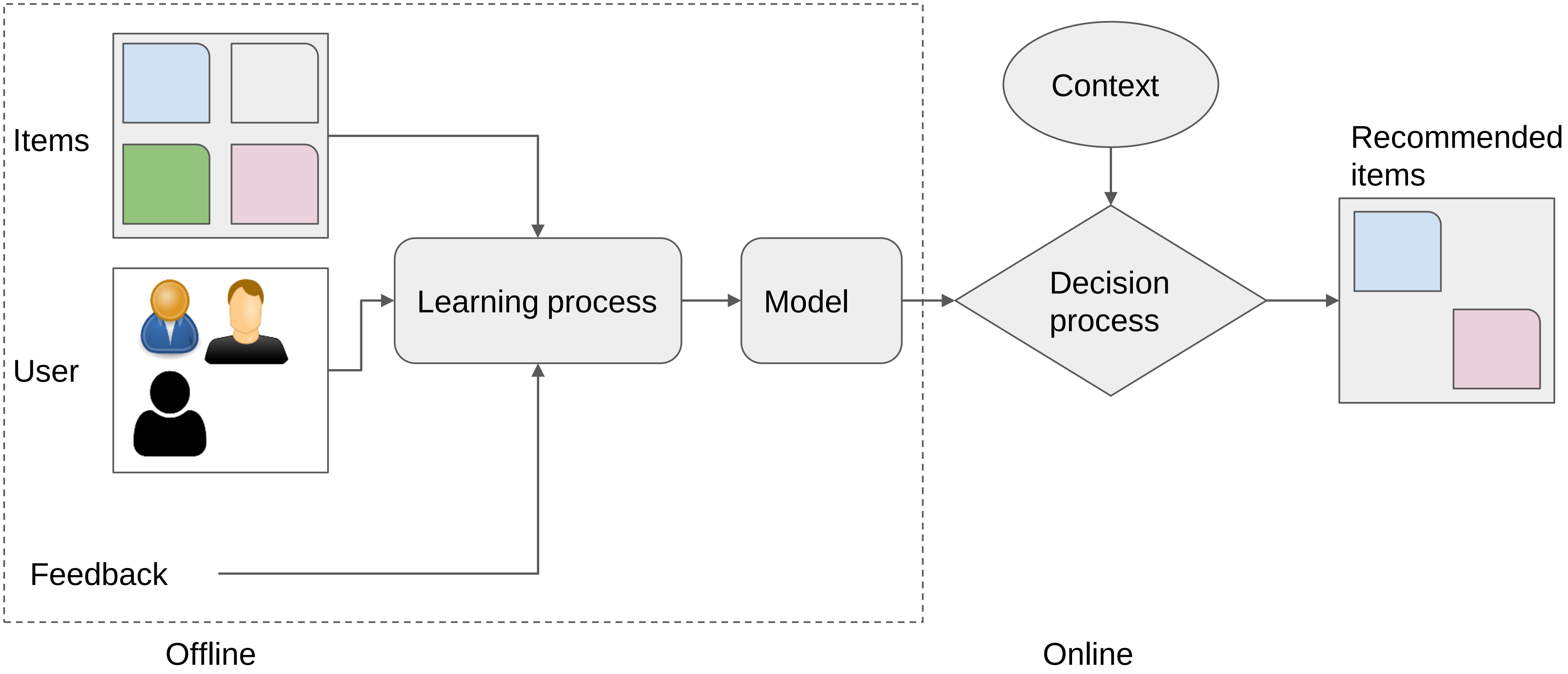
UNIVERSITÉ BISHOP'S UNIVERSITY

# Recommender system

CS417
R. Hedjam

Some slide are sdapted from Andrew Ng. & AI all in one (Stanford Univ)

# Recommendation process



# Input: source of information

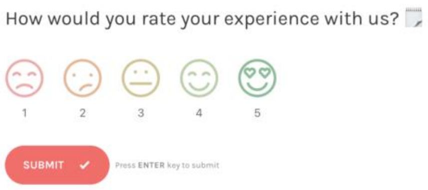
Explicit feedback:

- The users provide explicit signals about their preferences.
  - Explicit numerical rating, e.g., 2 stars, 3 stars, etc.
  - Binary explicit,  thumbs up, thumbs down
- The most popular ratings.

Implicit feedback:

- No explicit feedback of users.
- Clicks/queries/watches to infer preference
- Lack of clicks is implicit lack of preference.
- Time to read a webpage by the users
- Many times an item is sold

Likert scale




# Explicit vs. implicit feedback

Implicit feedback:

If a user did not listen to a track, that does not imply he does not like the track.

if a user listened to track A, 10 times and track B 100 times, then we can infer that he has a higher preference for track B than track A.

| | Implicit feedback | Explicit feedback |
|---|---|---|
| Accuracy | Low | High |
| Abundance | High | Low |
| Context-sensitive | Yes | Yes |
| Expressivity of user preference | Positive | Positive and Negative |
| Measurement reference | Relative | Absolute |

G. Jawaheer et al. Comparison of implicit and explicit feedback from an online music recommendation service

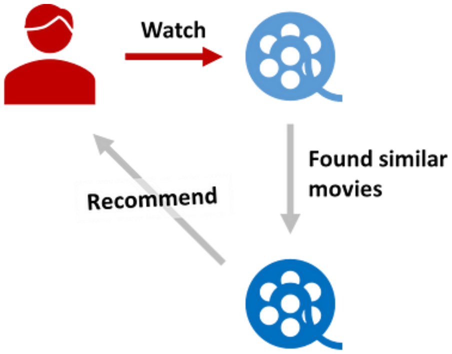
# RS: problem

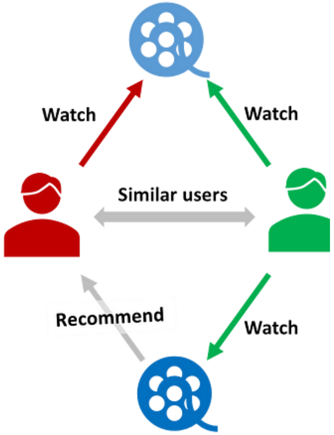
**Example: Predicting movie ratings**

| Movie | Alice (1) | Bob (2) | Carol (3) | Dave (4) |
|---|---|---|---|---|
| Love at last | 5 | 5 | 0 | 0 |
| Romance forever | 5 | ? | ? | 0 |
| Cute puppies of love | ? | 4 | 0 | ? |
| Nonstop car chases | 0 | 0 | 5 | 4 |
| Swords vs. karate | 0 | 0 | 5 | ? |

# RS approaches

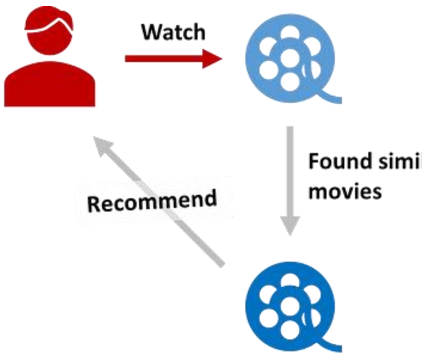
| Type | Definition | Example |
|---|---|---|
| **content-based filtering** | Uses *similarity between items* to recommend items similar to what the user likes. | If user A watches two cute cat videos, then the system can recommend cute animal videos to that user. |
| **collaborative filtering** | Uses *similarities between queries and items simultaneously* to provide recommendations. | If user A is similar to user B, and user B likes video 1, then the system can recommend video 1 to user A (even if user A hasn't seen any videos similar to video 1). |

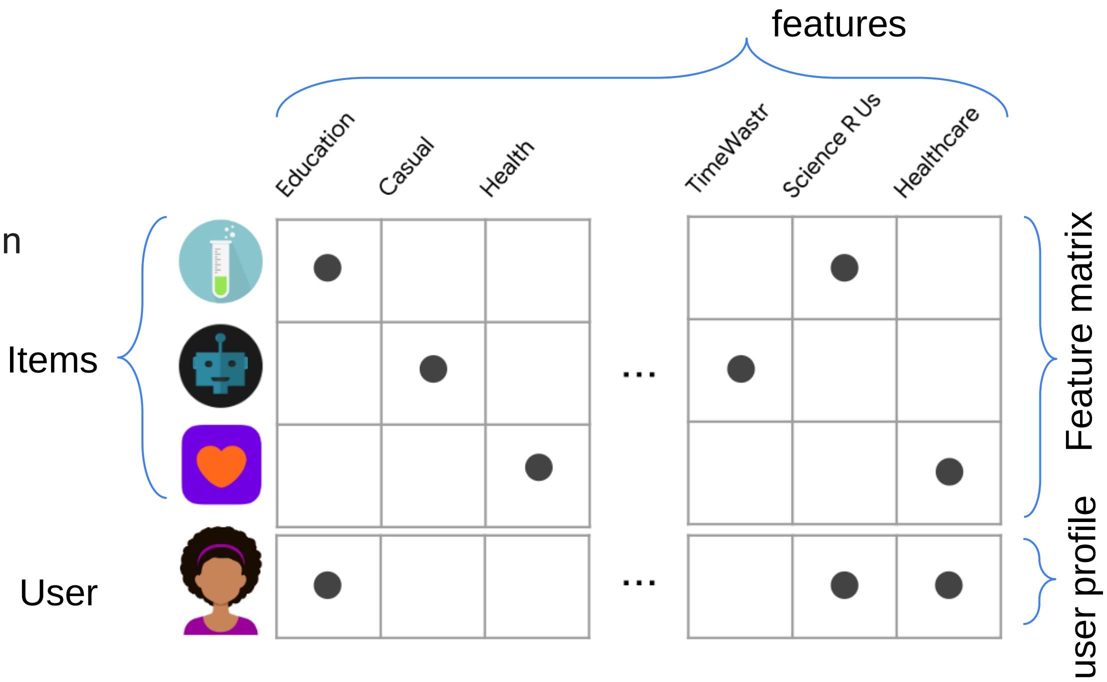


# Content based RS

- Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback.

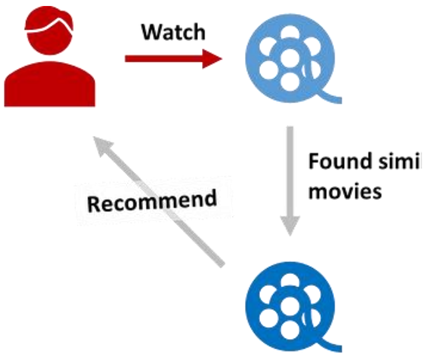


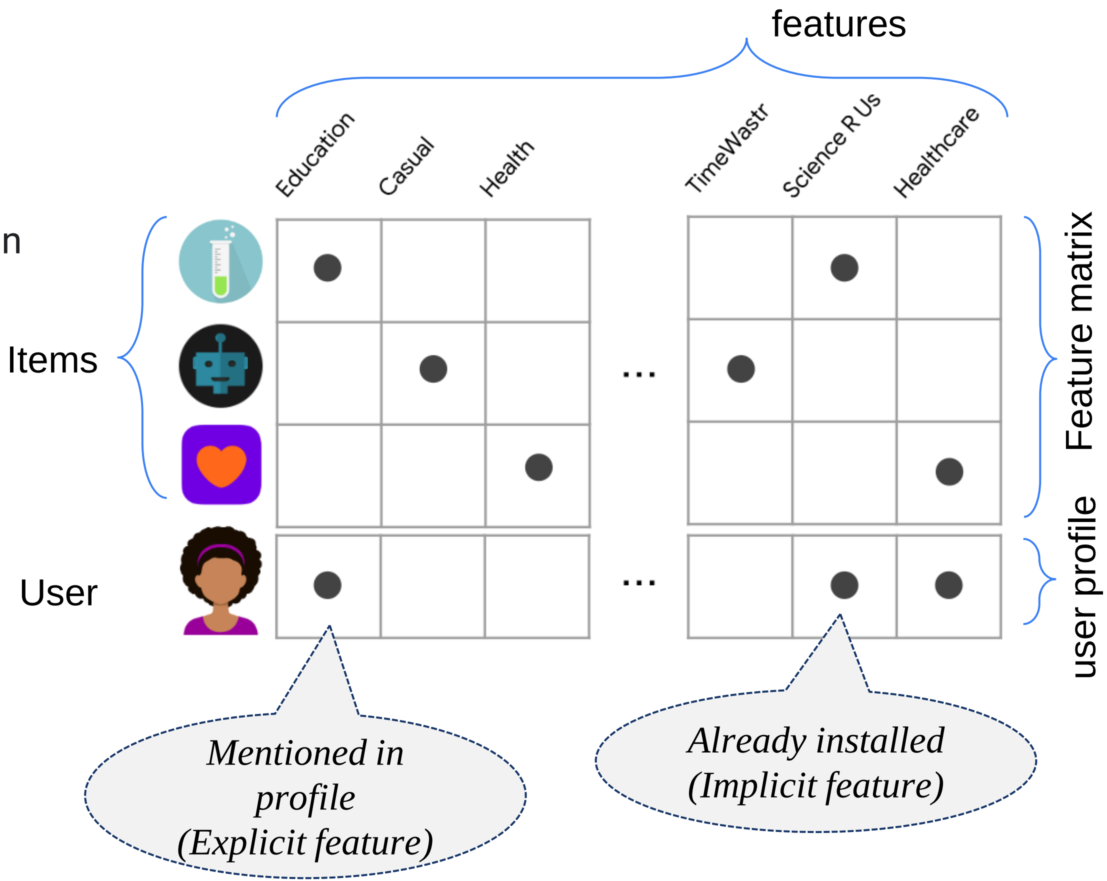




https://developers.google.com/machine-learning/recommendation/content-based/basics

# Content based RS

- Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback.




https://developers.google.com/machine-learning/recommendation/content-based/basics

# Similarity metric

- Compute similarity between items feature vectors and use profile vector.
- Score each candidate item according to this similarity metric.
- Cosine metric

$$s(q, x) = \cos(q, x)$$

- Dot product

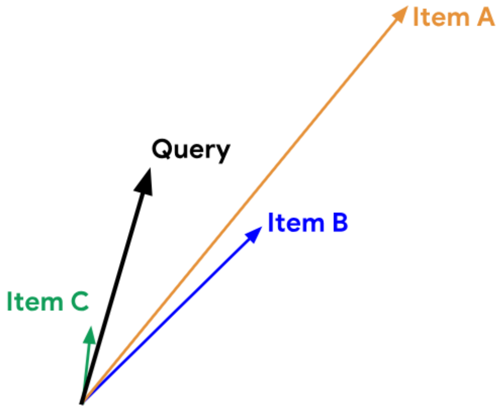
$$s(q, x) = \langle q, x \rangle = \sum_{i=1}^{d} q_i x_i \, .$$

- Euclidean distance

$$s(q, x) = \|q - x\| = \left[ \sum_{i=1}^{d} (q_i - x_i)^2 \right]^{\frac{1}{2}}$$

# Comparing metrics

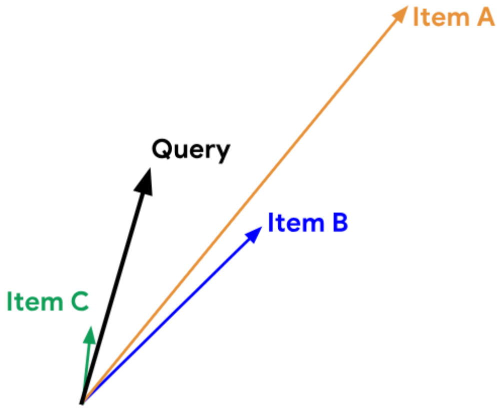
- Determine the item ranking using all three of the similarity measures: cosine, dot product, and Euclidean distance.



# Comparing metrics

- Determine the item ranking using all three of the similarity measures: cosine, dot product, and Euclidean distance.

- Item A has the largest norm, and is ranked higher according to the dot-product. Item C has the smallest angle with the query, and is thus ranked first according to the cosine similarity. Item B is physically closest to the query so Euclidean distance favors it.



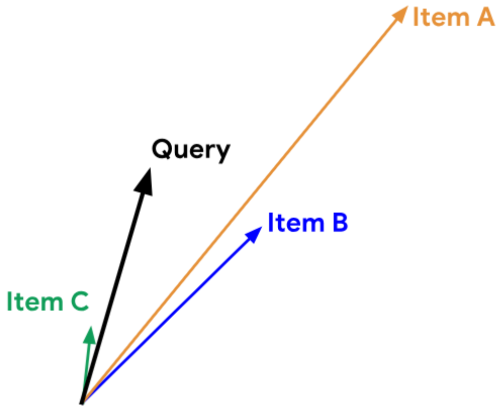

- **Dot-product:**
  Query: Item A -> Item B -> Item C
- **Cosine**
  Querr: Item C -> Item A -> Item B
- **Euclidean**
  Query: Item B -> Item C -> Item A

# Comparing metrics

- Determine the item ranking using all three of the similarity measures: cosine, dot product, and Euclidean distance.

- Item A has the largest norm, and is ranked higher according to the dot-product. Item C has the smallest angle with the query, and is thus ranked first according to the cosine similarity. Item B is physically closest to the query so Euclidean distance favors it.

- **Dot-product:**
  Query: Item A -> Item B -> Item C
- **Cosine**
  Querr: Item C -> Item A -> Item B
- **Euclidean**
  Query: Item B -> Item C -> Item A

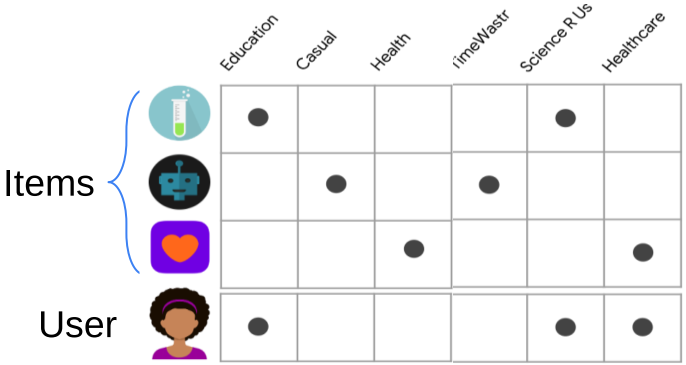



**Which Similarity Measure to Choose? Homework**

# Back to our example, which item will be recommended?

- Assume a binary feature matrix
- Using dot-product.

$$s(q,x) = \langle q, x \rangle = \sum_{i=1}^{d} q_i x_i .$$




- S(user, item1) = s1 = 1x1 + 0x0 + 0x0 + 0x0 + 1x1 + 1*0 = 2.
- S(user, item2) = s2 = 1x0 + 0x1 + 0x0 + 0x1 + 1x0 + 1x1 = 1.
- S(user, item3) = s3 = 1x0 + 0x0 + 0x1 + 0x0 + 1x0 + 1x1 = 1.
- s1>s2 and s1>s3
    - => Item 1 will be recommended.

https://developers.google.com/machine-learning/recommendation/content-based/basics

# Pros and cons of content-based RS

- Pros:
  - The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users.
  - The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.
- Cons:
  - Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
  - The model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

https://developers.google.com/machine-learning/recommendation/content-based/summary

# References


- https://medium.com/mlearning-ai/recommendation-systems-content-based-filtering-e19e3b0a309e
- https://redfield.ai/content-based-recommendation/
- https://builtin.com/data-science/recommender-systems
- https://www.kdnuggets.com/2020/07/building-content-based-book-recommendation-engine.html
- https://www.diva-portal.org/smash/get/diva2:935353/FULLTEXT02.pdf