Berkeley Engineering | BerkeleyHaas

PROFESSIONAL CERTIFICATE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Office Hour #19 with Matilde D’Amelio

August 4, 2022 at 9 pm UTC

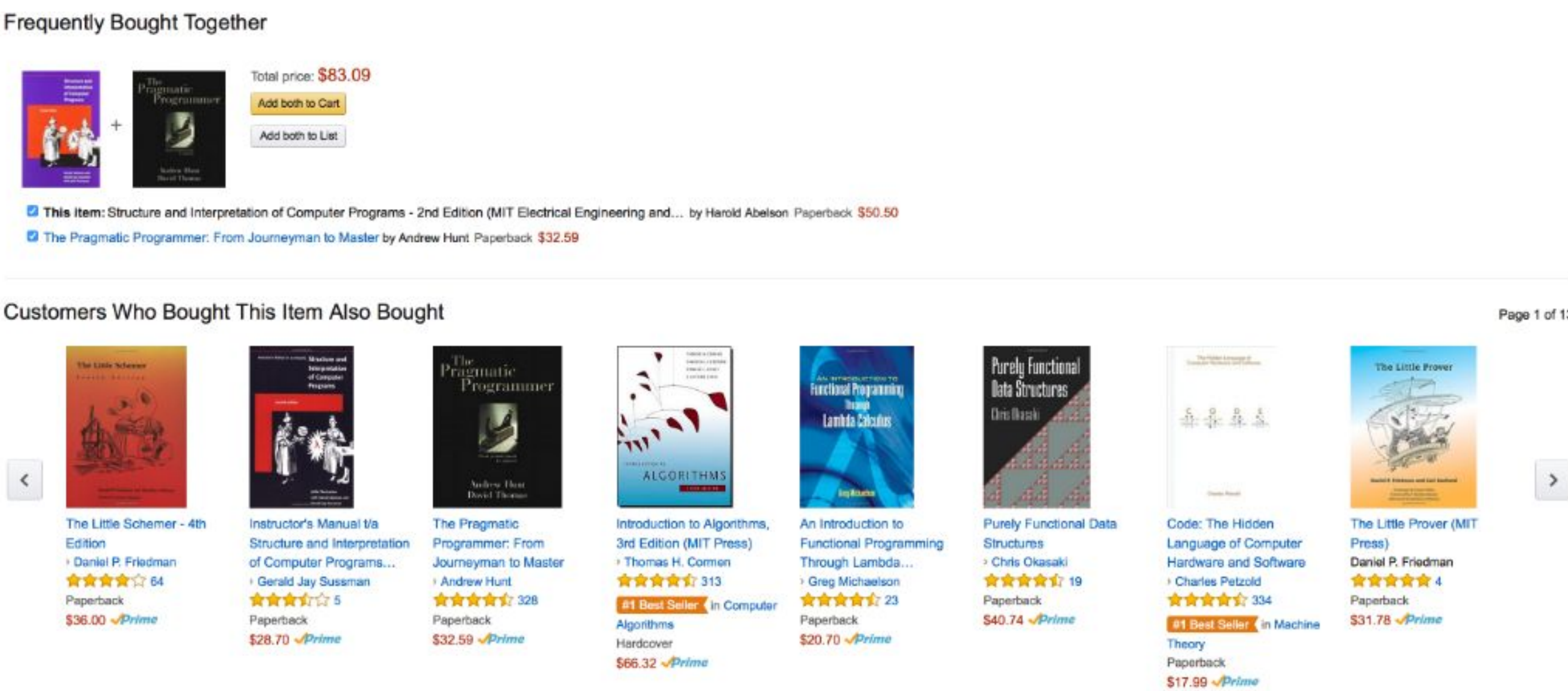
Recommendation Systems

Recommender Systems, they are simple algorithms which aim to provide the most relevant and accurate items to the user by filtering useful stuff from of a huge pool of information base. Recommendation engines discovers data patterns in the data set by learning consumers choices and produces the outcomes that co-relates to their needs and interests

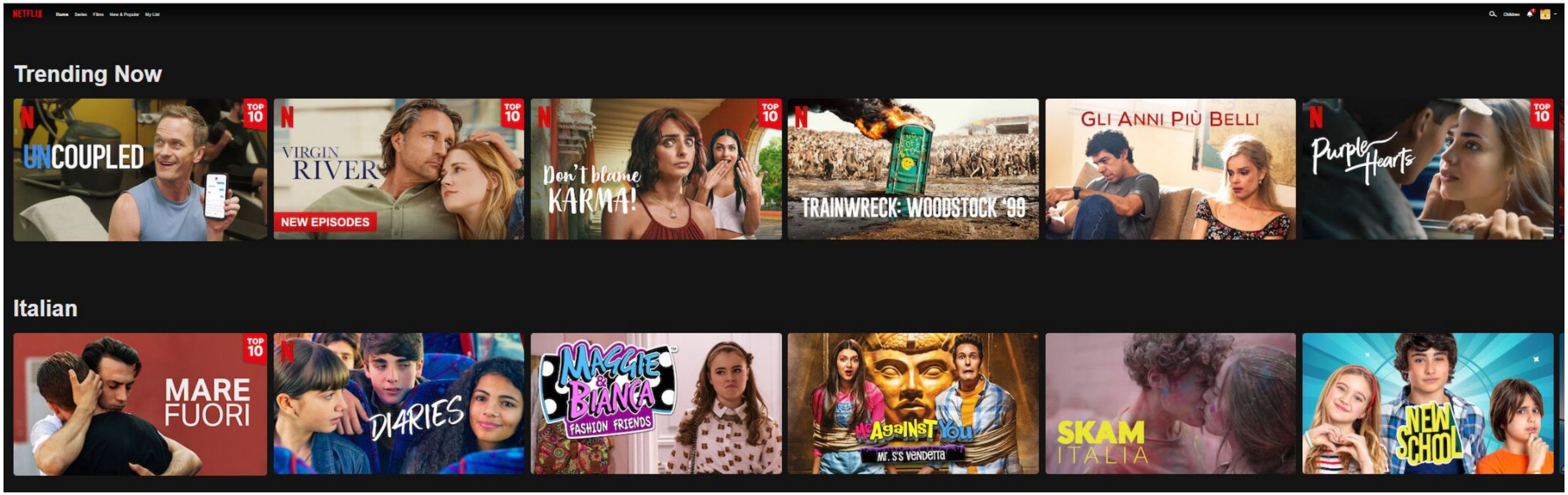
What can be examples of Recommendation Systems?

Berkeley Engineering | BerkeleyHaas

Recommendation Systems- Amazon

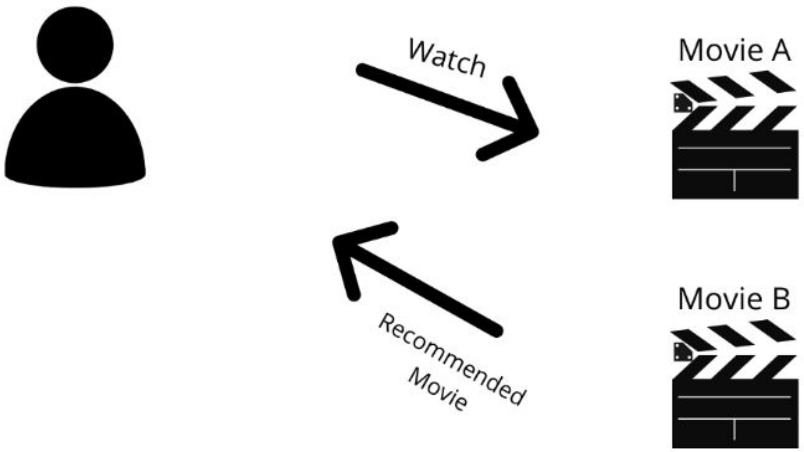


Recommendation Systems- Netflix



Content-Based Filtering

Content-based filtering is a type of recommender system that attempts to guess what a user may like based on that user's activity. Content-based filtering makes recommendations by using **keywords and attributes assigned to objects in a database (e.g., items in an online marketplace) and matching them to a user profile**. The user profile is created based on data derived from a user's actions, such as purchases, ratings (likes and dislikes), downloads, items searched for on a website and/or placed in a cart, and clicks on product links. For example, suppose you're recommending accessories to a user that just purchased a smartphone from your website and has previously bought smartphone accessories. Aside from keywords such as the smartphone manufacturer, make, and model, the user profile indicates prior purchases include phone holders with sleeves for credit cards. Based on this information, the recommender system may suggest similar phone holders for the new phone with attributes such as an RFID blocking fabric layer to help prevent unauthorized credit card scanning. In this example, the user would expect recommendations for similar phone holders, but the RFID blocking feature may be something they didn't expect yet appreciate.



Content-Based Filtering

Assigning attributes: Content-based filtering relies on assigning attributes to database objects so the algorithm knows something about each object. These attributes depend primarily on the products, services, or content you're recommending. Assigning attributes can be a monumental undertaking. Many companies resort to using **subject-matter expert teams to assign attributes to each item manually.** For example, Netflix has hired screenwriters to rate shows on aspects ranging from shooting locations and actors to plotlines, tone, and emotional effects. The resulting tags, used by the recommender, are algorithmically combined to group films together that share similar aspects.

Building a user profile: User profiles are another element crucial to content-based recommender systems. **Profiles include the database objects the user has interacted with**—purchased, browsed, read, watched, or listened to—as well as their assigned attributes.Attributes appearing across multiple objects are weighted more heavily than those that show up less often. This helps establish a **degree of importance** because not all of an object's attributes are equal to the user. **User feedback** is also critical when weighting items, which is why websites that provide recommendations are continually asking you to rate products, services, or content. Based on **attribute weightings and histories**, the recommender system produces a unique model of each user's preferences. The model consists of attributes the user is liable to like or dislike based on past activities, weighted by importance. User models are compared against all database objects, which are then assigned scores based on their similarity to the user profile.

Example: Let's say you've listened to Taylor Swift's "The Last Time," Shakira's "Can't Remember to Forget You," and "Me, Myself and I" by Beyoncé. A recommender system might recognize that you like female pop artists and breakup songs. You could expect to receive recommendations for more breakup songs by these and other female pop artists, such as Miley Cyrus's "Slide Away."The recommender system may also suggest different types of songs by Miley Cyrus because you appear to like female pop artists. Still, since you didn't choose to listen to this artist or songs unassociated with breakups before, these selections would receive a lower assigned score.

Collaborative Filtering

Collaborative filtering models are based on assumption that people like things similar to other things they like, and things that are liked by other people with similar taste.

Collaborative filtering models are two types:

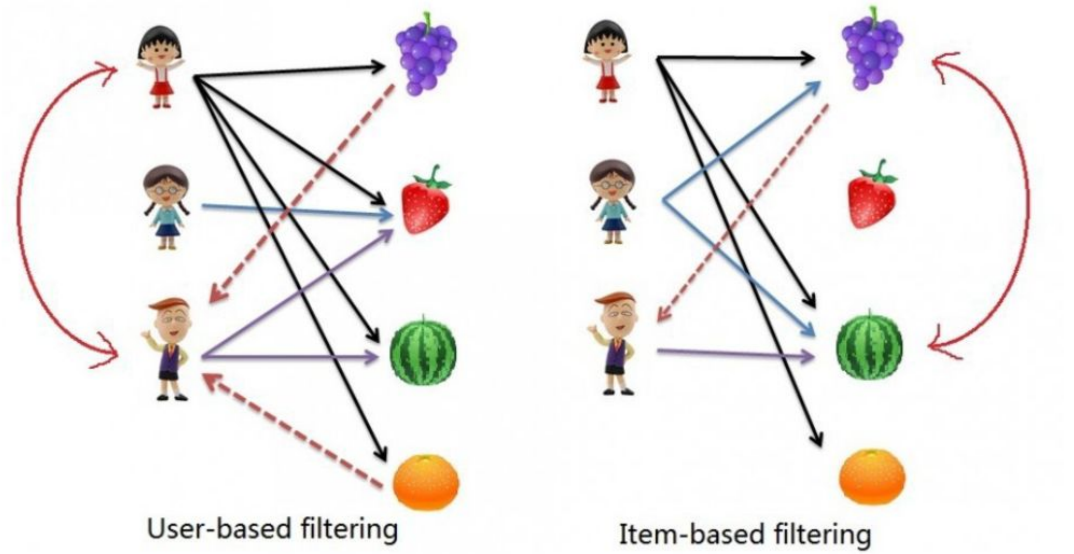
- **Nearest neighbor**
- **Matrix factorization**

Nearest Neighbor

These type of recommendation systems are recommending based on nearest neighbors, nearest neighbor approach used to find out either similar users or similar products

User-based collaborative filtering:
Find the users who have similar taste of products as the current user , similarity is based on purchasing behavior of the user, so based on the neighbor purchasing behavior we can recommend items to the current user.

Item-based collaborative filtering :
Recommend Items that are similar to the item user bought,similarity is based on co-occurrences of purchases. Item A and B were purchased by both users X and Y then both are similar.



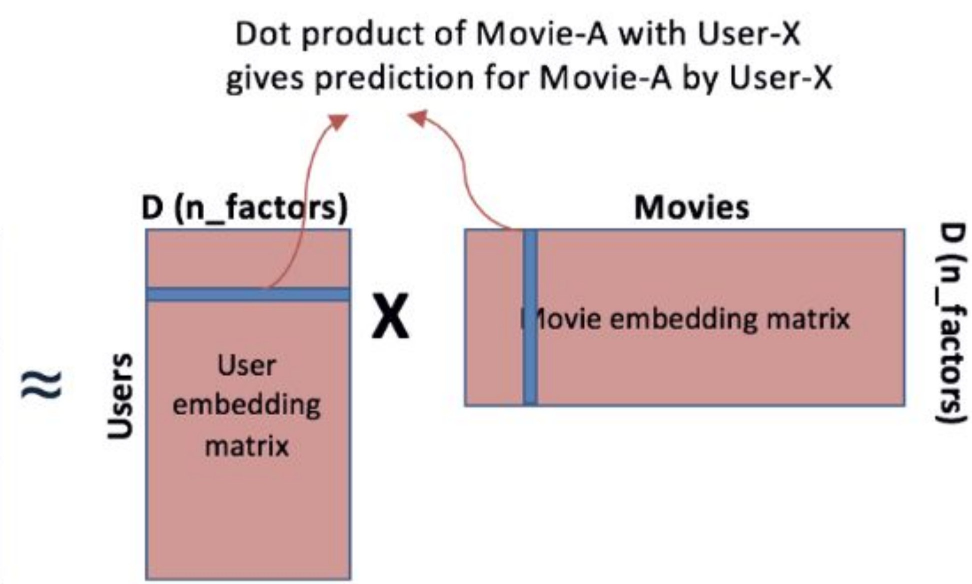
Matrix Factorization

When a user gives feedback to a certain movie they saw (say they can rate from one to five), this collection of feedback can be represented in a form of a matrix. Where each row represents each users, while each column represents different movies. Obviously the matrix will be sparse since not everyone is going to watch every movies, (we all have different taste when it comes to movies).

The idea behind such models is that attitudes or preferences of a user can be determined by a small number of hidden factors. We can call these factors as **Embeddings.**

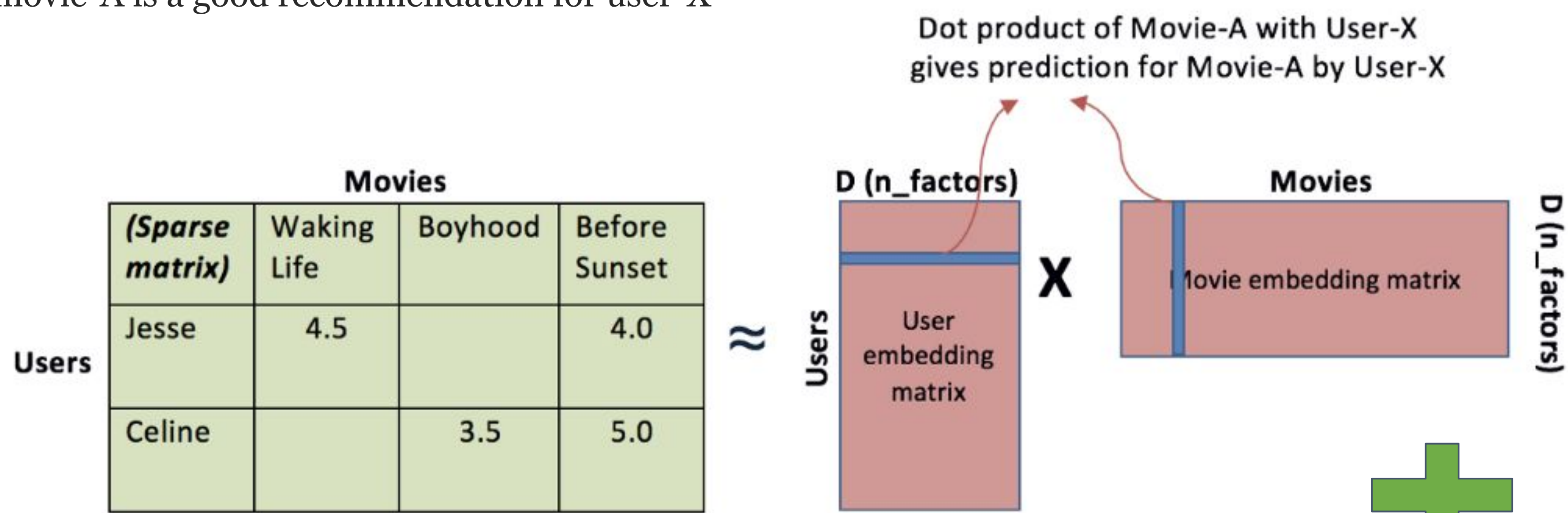
Movies

	(Sparse matrix)	Waking Life	Boyhood	Before Sunset
Users	Jesse	4.5		4.0
	Celine		3.5	5.0

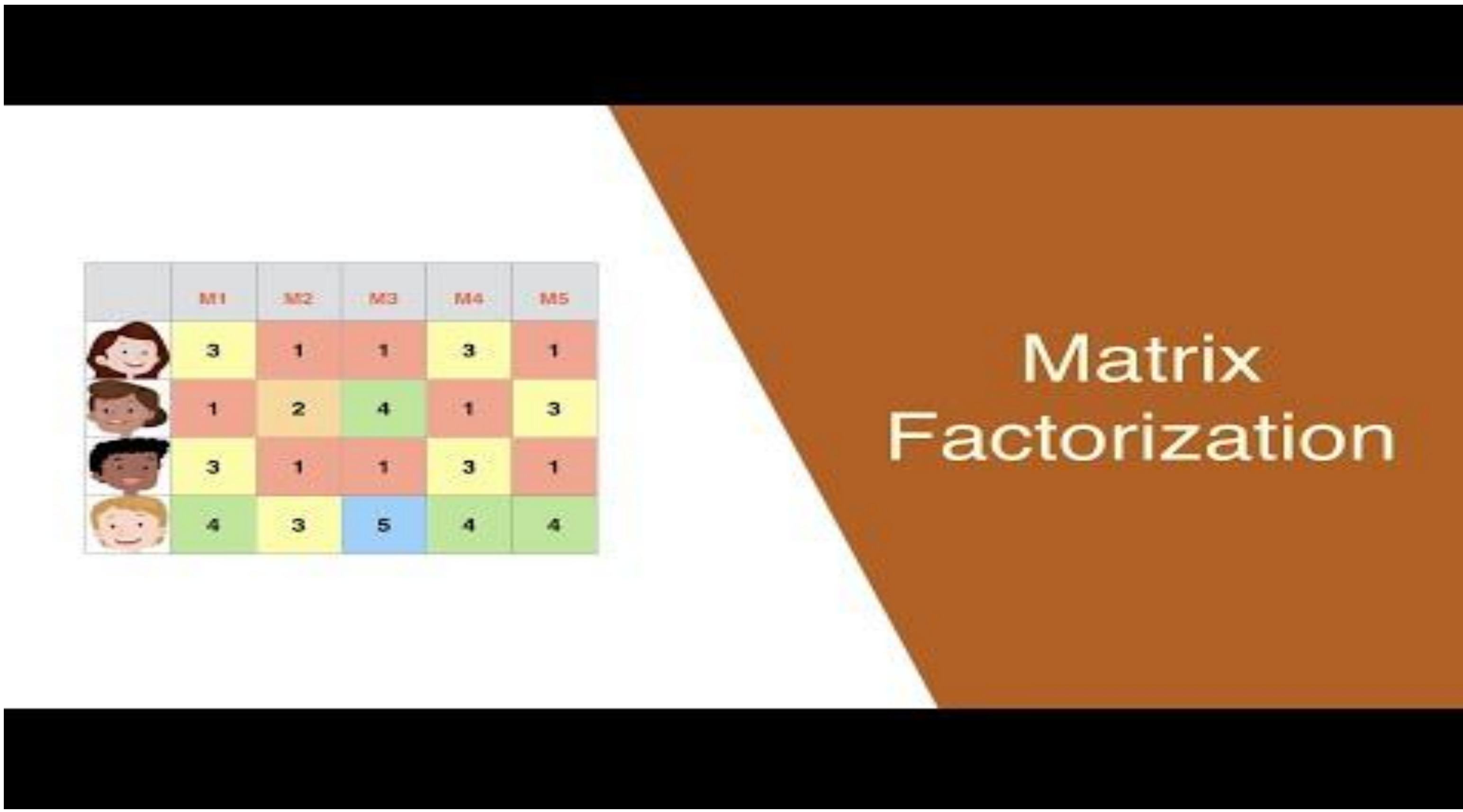


Matrix Factorization

For e.g. say we have 5 dimensional **(i.e. D or n_factors = 5** in above figure**)** embeddings for both items and users (# 5 chosen randomly). Then for user-X & movie-A, we can say the those 5 numbers **might** represent 5 different characteristics about the movie, like ***(i)*** *how much movie-A is sci-fi intense* ***(ii)*** *how recent is the movie* ***(iii)*** *how much special effects are in movie A* ***(iv)*** *how dialogue driven is the movie* ***(v)*** *how CGI driven is the movie. Likewise, 5 numbers in user embedding matrix might represent,* ***(i)*** *how much does user-X like sci-fi movie* ***(ii)*** *how much does user-X like recent movies …and so on.*
In below figure, a higher number from dot product of user-X and movie-A matrix means that movie-A is a good recommendation for user-X



How Netflix Recommend Movies



Hybrid Systems

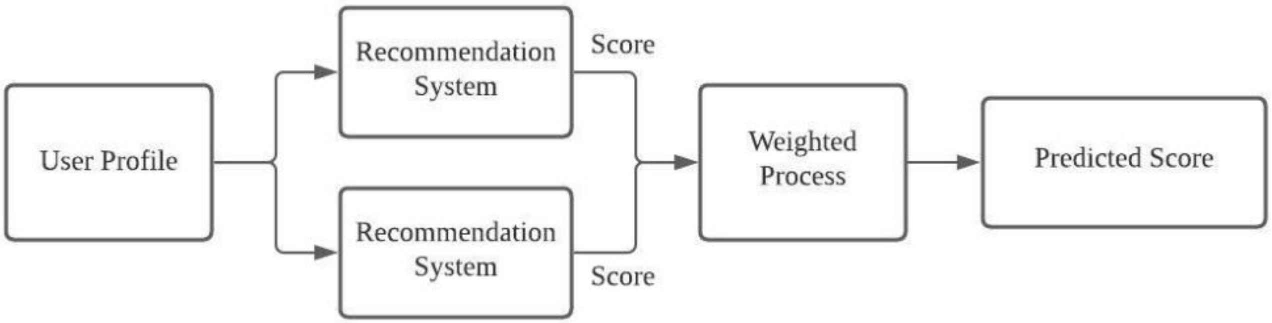
Hybrid Recommendation systems are combining collaborative and content-based recommendation can be more effective. Hybrid approaches can be implemented by making content-based and collaborative-based predictions separately and then combining them.

Types:

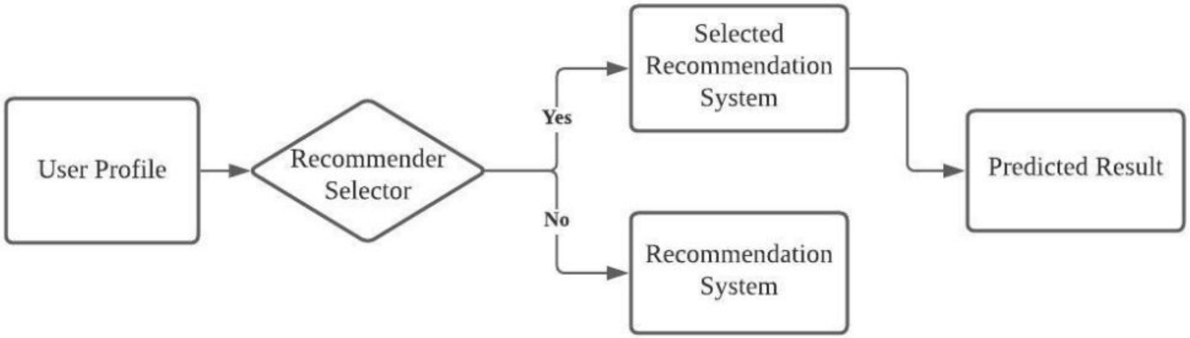
1. Weighted
2. Switching
3. Mixed
4. Feature Combination
5. Cascade
6. Feature Augmentation
7. Meta-Level

Hybrid Systems

Weighted: The weighted recommendation system will take the outputs from each of the models and combine the result in static weightings, which the weight does not change across the train and test set.

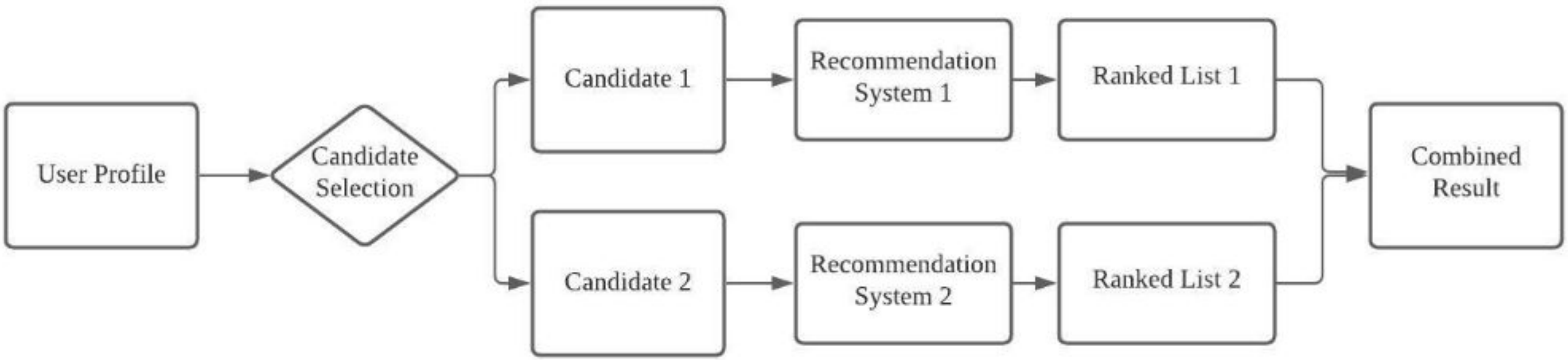


Switching: The switching hybrid selects a single recommendation system based on the situation. The model is used to be built for the item-level sensitive dataset, we should set the recommender selector criteria based on the user profile or other features



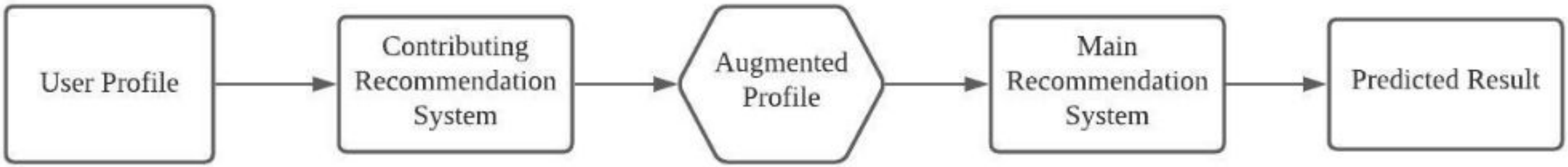
Hybrid Systems

Mixed: first takes the user profile and features to generate different set of candidate datasets. The recommendation system inputs different set of candidate to the recommendation model accordingly, and combine the prediction to produce the result recommendation

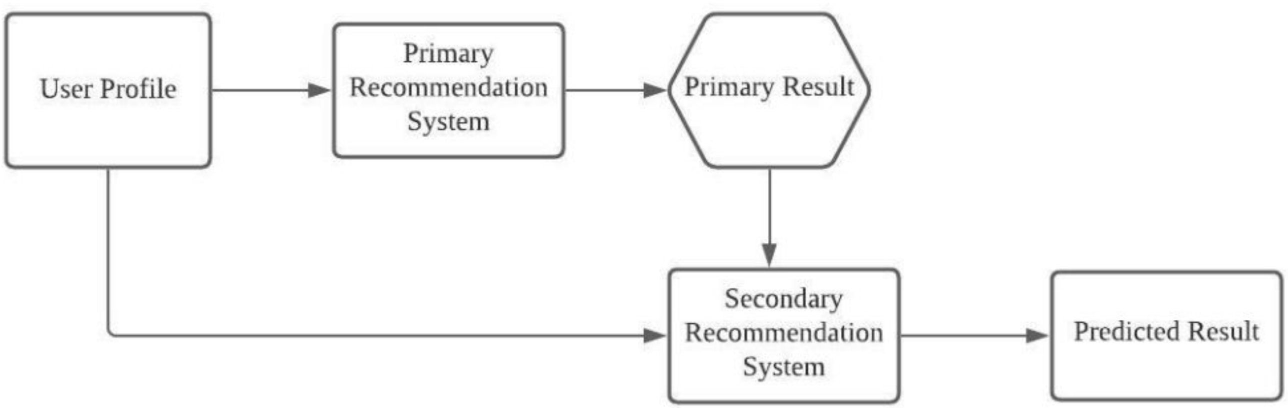


Hybrid Systems

Feature Augmentation: to improve the performance of the core system without changing the main recommendation model. For example, by using the association rule, we are able to enhance the user profile dataset. E.g., with the augmented dataset, the performance of content-based recommendation model will be improved

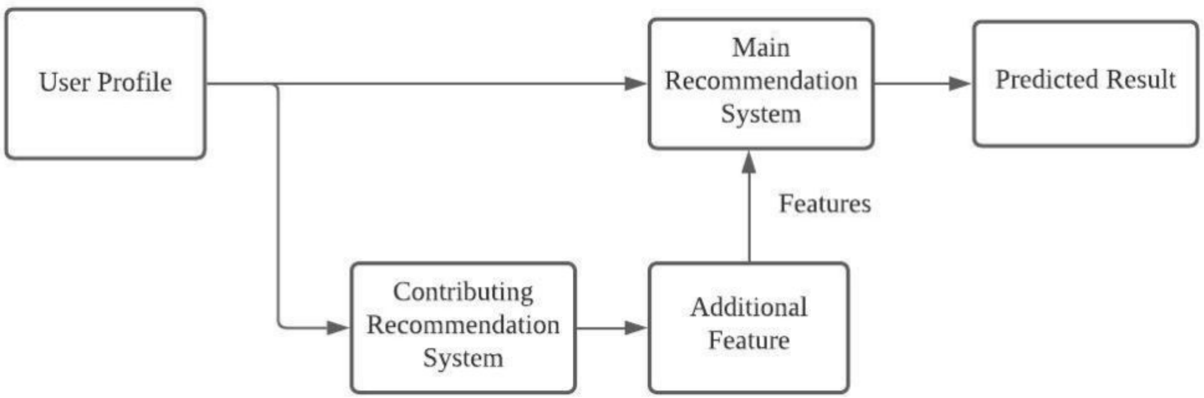


Cascade: defines a strict hierarchical structure recommendation system, such that the main recommendation system produce the primary result, and we use the secondary model to resolve some minor issues of the primary result, like breaking tie in the scoring.



Hybrid Systems

Feature Combination: We add a virtual contributing recommendation model to the system, which works as feature engineering toward the original user profile dataset.



Meta-Level: is similar to the feature augmentation hybrid, such that the contributing model is providing augmented dataset to the main recommendation model. Different from the feature augmentation hybrid, meta-level replaces the original dataset with a learned model from the contributing model as the input to the main recommendation model.

Berkeley Engineering | BerkeleyHaas

QUESTIONS?