Machine Learning

Apriori →

Recommendation System

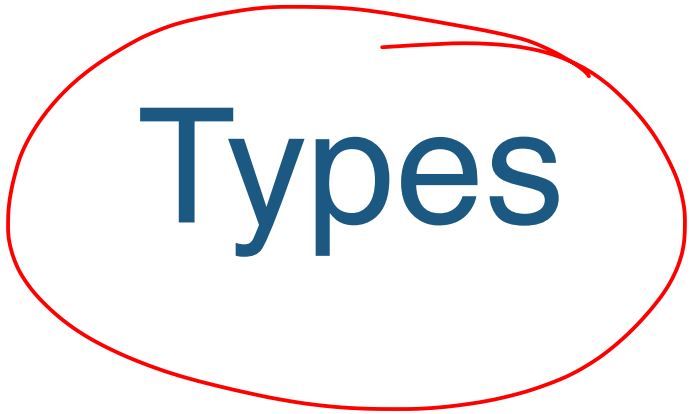
Overview

- Recommender systems are among the most popular applications of data science today
- They are used to predict the "rating" or "preference" that a user would give to an item
- Almost every major tech company has applied them in some or the other form
- Amazon uses it to suggest products to customers, YouTube uses it to decide which video to play next on autoplay, and Facebook uses it to recommend pages to like and people to follow
- Some companies like Netflix, Amazon Prime, Hulu, and Hotstar, the business model and its success revolves around the potency of their recommendations
- Netflix even offered a million dollars in 2009 to anyone who could improve its system by 10%.

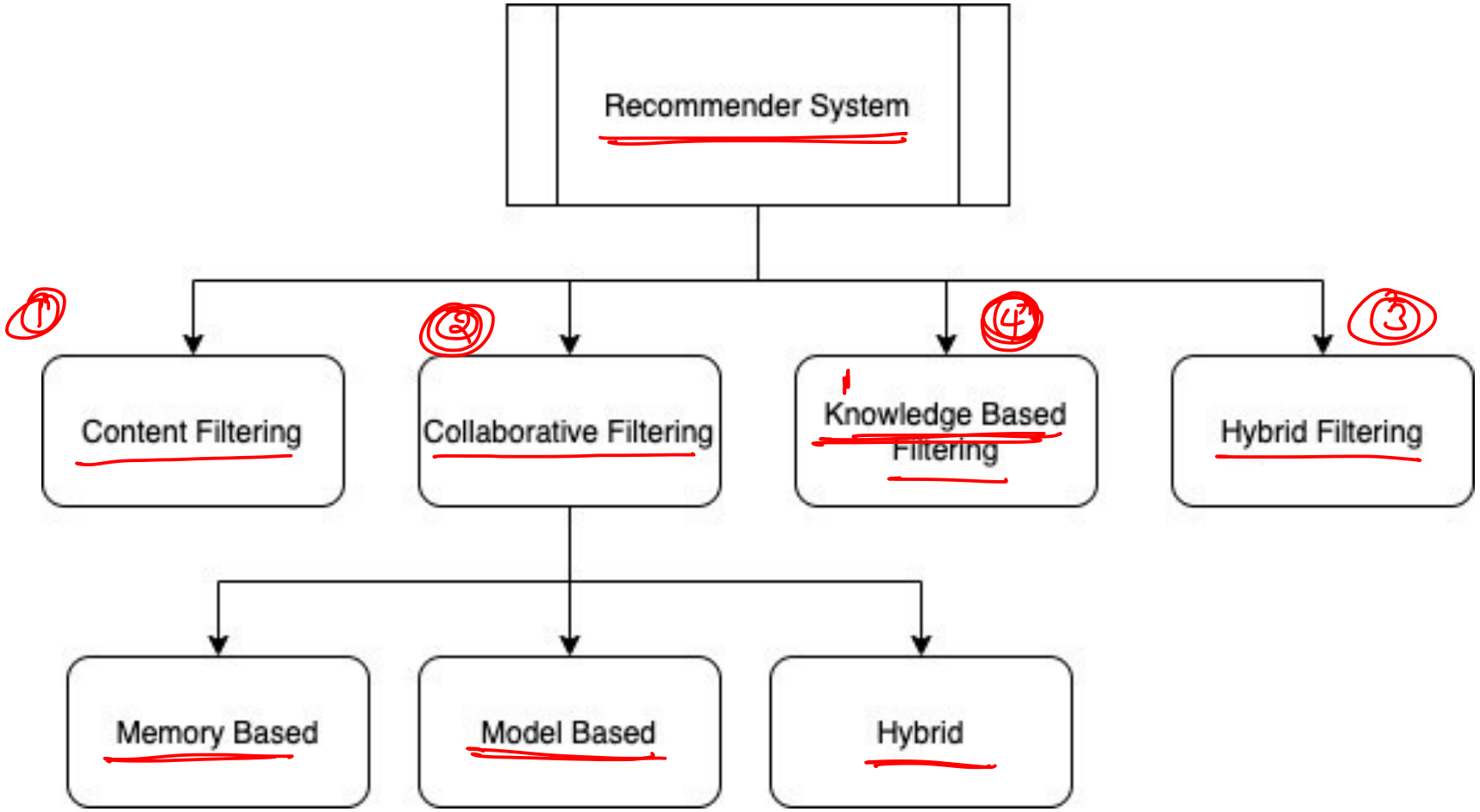
Overview

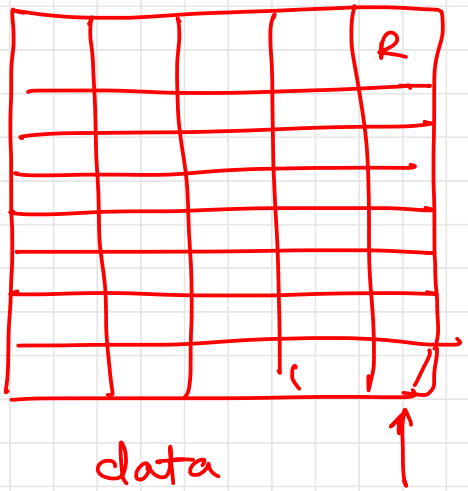
- There are also popular recommender systems for domains like restaurants, movies, and online dating
- Recommender systems have also been developed to explore research articles and experts, collaborators, and financial services
- YouTube uses the recommendation system at a large scale to suggest you videos based on your history
- For example, if you watch a lot of educational videos, it would suggest those types of videos

Types



Types


Recommender System
Content Filtering
Collaborative Filtering
Knowledge Based Filtering
Hybrid Filtering
Memory Based
Model Based
Hybrid



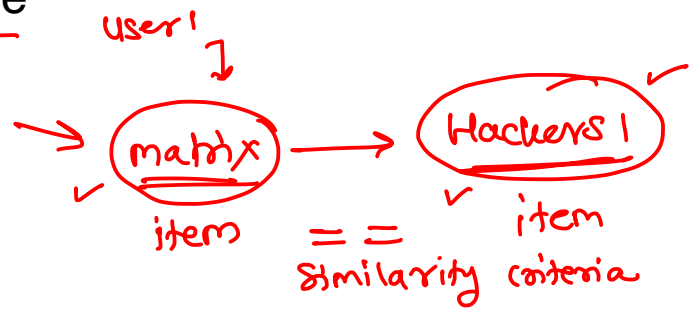
filtering ⇒ filtered data

↓

suggestions

Types

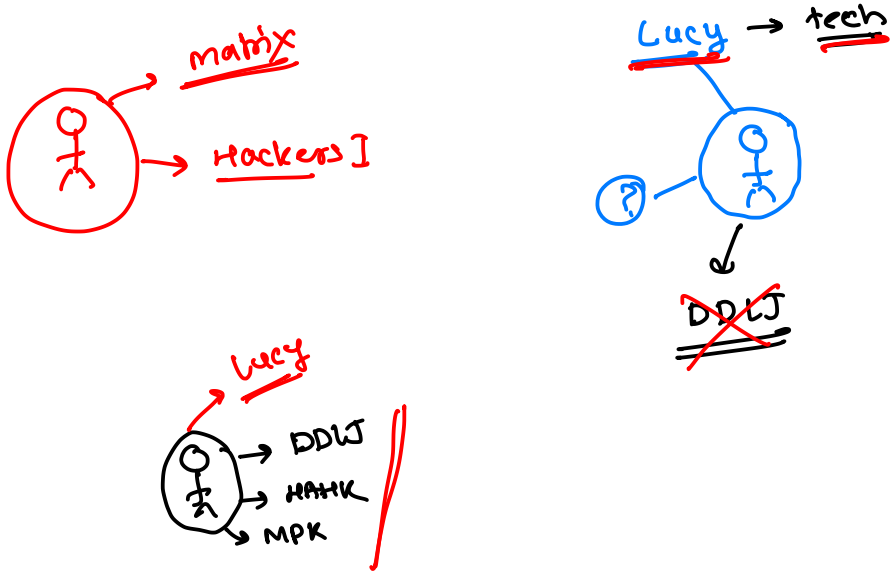
- **Simple recommenders** (knowledge)
  - It offers generalized recommendations to every user, based on movie popularity and/or genre
  - The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience
  - An example could be IMDB Top 250



- **Content-based recommenders**
  - It suggest similar items based on a particular item
  - This system uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations
  - The general idea behind these recommender systems is that if a person likes a particular item, he or she will also like an item that is similar to it
  - And to recommend that, it will make use of the user's past item metadata
  - A good example could be YouTube, where based on your history, it suggests you new videos that you could potentially watch

Types

- **Collaborative filtering engines**
  - These systems are widely used, and they try to predict the rating or preference that a user would give an item-based on past ratings and preferences of other users
  - Collaborative filters do not require item metadata like its content-based counterparts



Steps

Data Collection

- This is the first and most crucial step for building a recommendation engine
- The data can be collected by two means:

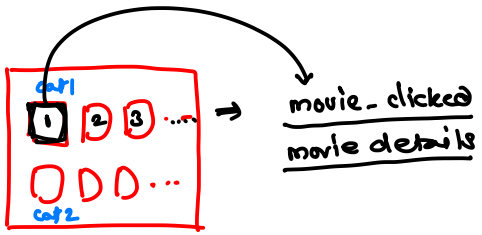
① Explicitly
  - Explicit data is information that is provided intentionally
  - E.g. input from the users such as movie ratings

② Implicitly
  - Implicit data is information that is not provided intentionally but gathered from available data streams
  - E.g. search history, clicks, order history, etc

GUI / api
→ category ⇒
→ language ⇒
→ Genre ⇒
} settings / Preferences
direct data

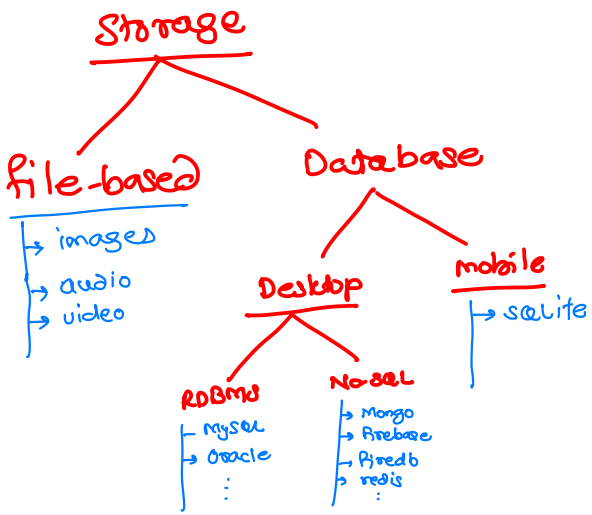
comments

Explicit ⇒ comment ⇒ NLP ⇒ meaning / context / sentiment ⇒ filtering



Data Storage

- The amount of data dictates how good the recommendations of the model can get
- For example, in a movie recommendation system, the more ratings users give to movies, the better the recommendations get for other users
- The type of data plays an important role in deciding the type of storage that has to be used
- This type of storage could include a standard SQL database, a NoSQL database or some kind of object storage

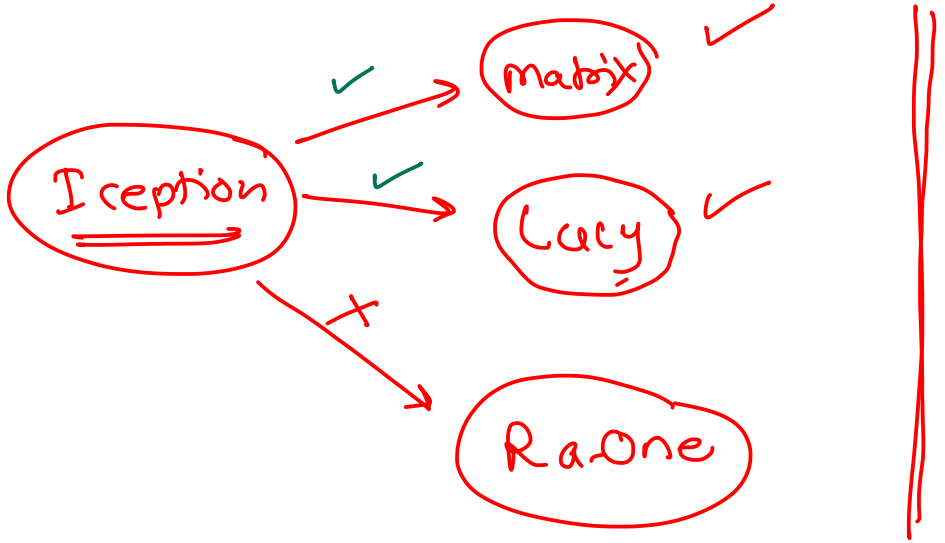


Filtering the data

- After collecting and storing the data, we have to filter it so as to extract the relevant information required to make the final recommendations
- There are various algorithms that help us make the filtering process easier
  - Content based filtering
  - Collaborative filtering

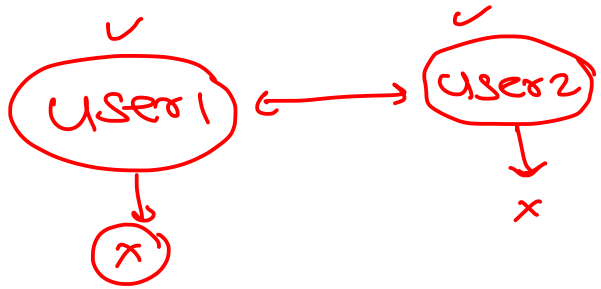
Content based filtering

- This algorithm recommends products which are similar to the ones that a user has liked in the past
- For example, if a person has liked the movie “Inception”, then this algorithm will recommend movies that fall under the same genre. But how does the algorithm understand which genre to pick and recommend movies from?

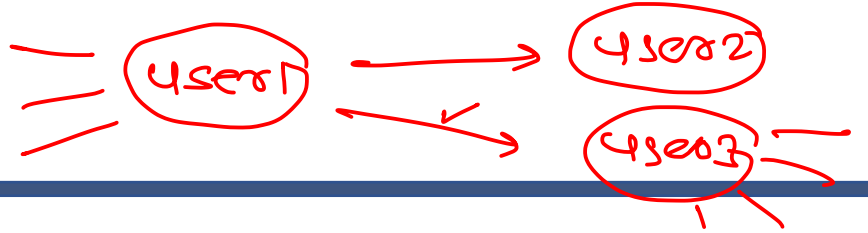


Collaborative filtering

- The collaborative filtering algorithm uses "User Behavior" for recommending items
- This is one of the most commonly used algorithms in the industry as it is not dependent on any additional information
- E.g.
  - If person A likes 3 movies, say Interstellar, Inception and Predestination
  - Person B likes Inception, Predestination and The Prestige
  - Then they have almost similar interests
  - We can say with some certainty that A should like The Prestige and B should like Interstellar
- There are different types of collaborating filtering techniques
  - User-User collaborative filtering
  - Item-Item collaborative filtering



User-User collaborative filtering



- This algorithm first finds the similarity score between users
- Based on this similarity score, it then picks out the most similar users and recommends products which these similar users have liked or bought previously
- This algorithm is quite time consuming as it involves calculating the similarity for each user and then calculating prediction for each similarity score
- One way of handling this problem is to select only a few users (neighbors) instead of all to make predictions, i.e. instead of making predictions for all similarity values, we choose only few similarity values
- There are various ways to select the neighbors:
  - Select a threshold similarity and choose all the users above that value
  - Randomly select the users
  - Arrange the neighbors in descending order of their similarity value and choose top-N users
  - Use clustering for choosing neighbors

Item-Item collaborative filtering

- In this algorithm, we compute the similarity between each pair of items
- This algorithm works similar to user-user collaborative filtering with just a little change – instead of taking the weighted sum of ratings of "user-neighbors", we take the weighted sum of ratings of "item-neighbors"

* Apriori & Eclat