Go Back to Recommendation Systems

Course Content

The Basics of Recommendation Systems

The term **Recommendation Systems** itself is self-explanatory. A Recommendation System (also known as a Recommender System or Recommendation Engine) is an automated system that provides preferable suggestions/recommendations of products to users.

For example: When you watch a movie of some particular genre on Netflix or watch some video on YouTube, they start to give you suggestions/recommendations for similar types of content that you may like.

Another example could be Amazon's recommendation system for online shoppers, after you view some product, it starts suggesting similar products for you to buy. After you listen to songs on the app, the recommendation systems of music apps like Spotify or Apple Music start to suggest other similar songs you may like to listen to.

Recommendation Systems are one of the most profitable applications of machine learning in the world today. The Netflix Recommendation Engine, for example, is reportedly worth $1 billion per year, and **around 80% of viewer activity is driven by personalized recommendations.** Nearly every major successful product today utilizes recommendation systems in some way, and these recommendation systems are all similar in one important way of using the customer data (and other kinds of data) to try and provide good recommendations of products to customers.

Recommendation Systems essentially use the customer data (what you watch, what you buy, what music you listen to, etc.) to find patterns in that data, and based on these patterns, **provide personalized suggestions to improve customer experience.**

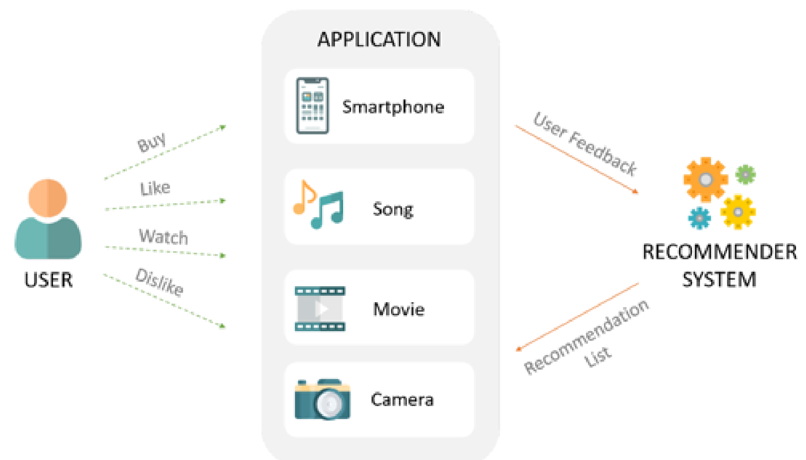

Image source

Data representation for Recommendation Systems

The customer data used to build Recommendation Systems is represented as a matrix of users and items. This matrix can vary according to the problem at hand.

For example, the figure below represents the matrix where rows are users and columns are movies, and the values in the matrix represent the rating given by a particular user to a particular movie.

		Item0	Item1	Item2	Item3	Item4	Item5	Item6
		0	1	2	3	4	5	6
User0	0	?	3	5	4	2	5	2
User1	1	5	5	?	5	5	3	5
User2	2	2	3	4	5	2	5	2
User3	3	2	2	3	2	?	2	3
User4	4	4	?	5	3	2	5	4
User5	5	5	2	5	3	2	4	4

? - Unknown Ratings

Image Source

This type of matrix is called a **user-item interaction matrix**. The ‘?’ in the matrix represents missing data, i.e., there is no interaction between that user-item pair, or in other words, that user has not rated that movie.

In reality, most of the data points in such a matrix would be ‘?’. There would be a large number of users and movies, and the actual number of user-item interactions which took place would usually pale in comparison to the total number of possible user-item interactions because **only a few users rate a few movies.**

In other words, this represents a **sparse matrix**.

Like every machine learning algorithm, a recommendation system makes predictions based on users' historical behavior. It tries to predict user preference for a set of items based on past experience.

To achieve this task, many methods exist.

Averaging

One simple way of doing this is to assume that every user is the same and we predict values simply by the average of the column, or in our case, the prediction is the average rating given to some movie by all the users who watched that movie.

Content-based Filtering

The content-based approach uses additional information about users and/or items. If we consider the example of a movie recommender system, this additional information could be the age of the user, the gender, the job, or any other personal information for users. It could also be additional information about the movie, such as the genre, the cast, the duration, and other details.

Then, the idea of content-based methods is to try to build a model, based on the available “features”, that explain the observed user-item interactions. Still considering the example of users and movies, we will try, for example, to model the fact that young women tend to better rate some movies, young men tend to rate some movies better, and so on. If we manage to get such a model, then making new predictions for a user is pretty easy, we just need to look at the profile (age, sex, etc.) of this user and, based on this information, determine relevant movies to suggest.