**Machine Learning Based Recommendation Systems**

### Introduction to Recommendation System

The basic purpose of a recommendation system is to find and recommend items that a user is most likely to be interested in. When we visit e-commerce sites such as Amazon, Apple Music, or Netflix, your host recommends you some products. These recommendations are based on your past purchases or the products you might be interested in. The system or application behind calculating these endorsements is called recommendation system. It saves users’ time by giving them the best of their choice and increases the potential sale of the business.

Formally, we define a recommendation system as:

The Recommendation System is a computer program that filters and recommends product or content to users by analyzing their behavior of rating or preference they had given in the past.

Examples:

* Recommendation of Movies and shows by Netflix.
* Recommendation of music by Apple music store.
* Social connection recommendations by Facebook, LinkedIn, or Instagram.
* Recommendation of dates by dating applications.
* Banking and insurance products recommendation.

## How does a recommendation engine work?

### Data collection

This is the first and most crucial step for building a recommendation engine. The data can be collected by two means: explicitly and implicitly. Explicit data is information that is provided intentionally, i.e. input from the users such as movie ratings. Implicit data is information that is not provided intentionally but gathered from available data streams like search history, clicks, order history, etc.

### 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.

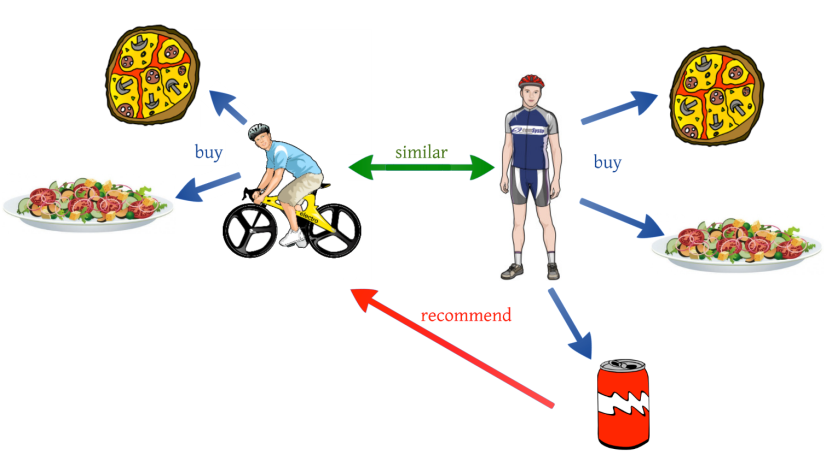
**Collaborative Filtering:**

Use knowledge of user’s past purchase/selection or similar decisions by other users to recommend products (User-based recommendation).

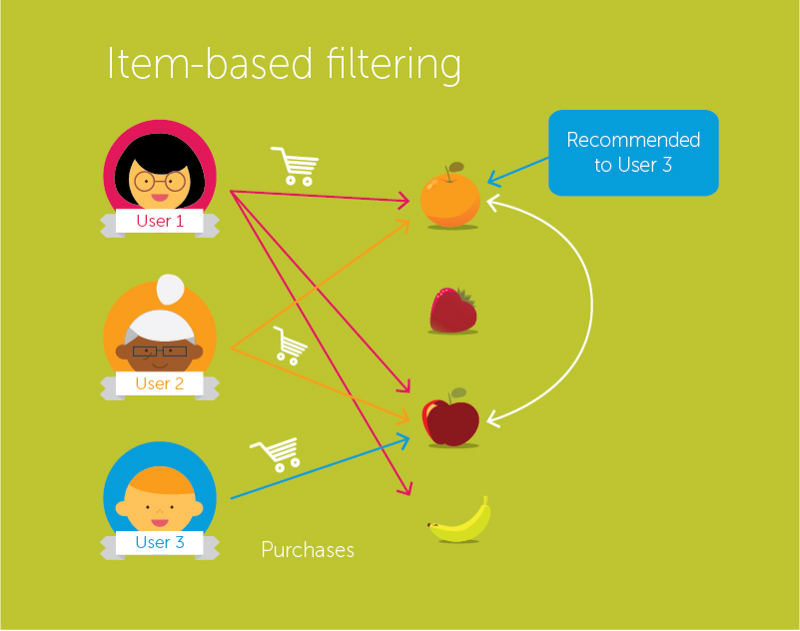
* Advantage – Product knowledge not required.
* Disadvantage – Can’t recommend products if no user reviews available so difficult to make recommendations for new users. It may be biased towards products with high reviews than the product with low reviews.

Example: If person A likes 3 movies, say Interstellar, Inception and Predestination, and 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. 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

#### User-User collaborative filtering



#### Item-Item collaborative filtering



**Content-Based Recommender:**

Use knowledge of each product to recommend a similar product (Product based recommendation).

* Advantage – Even works without user reviews.
* Disadvantage – Need descriptive data for every product so difficult to implement for large inventory products.

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?

Consider the example of Netflix. They save all the information related to each user in a vector form. This vector contains the past behavior of the user, i.e. the movies liked/disliked by the user and the ratings given by them. This vector is known as the *profile vector*. All the information related to movies is stored in another vector called the *item vector*. Item vector contains the details of each movie, like genre, cast, director, etc.

### Based on credits genres and keywords

* Based on movies with best MDB or tomatometer rating
* Recommend movies to user based on other users rating Using clustering
* Car recommendation

**BASED ON CREDIT, GENRES etc**

This is an example of content based filtering

In this section, you will try to build a system that recommends movies that are similar to a particular movie. More specifically, you will compute pairwise similarity scores for all movies based on their plot descriptions and recommend movies based on that similarity score.