**Retail connect work bench**

## Introduction :

We tend to buy products recommended by people because we trust the person. And nowadays in the digital age, any online shop you visit utilizes some sort of recommendation engine. And if set up and configured properly, it can significantly boost revenues, CTRs, conversions, and other important metrics. Moreover, they can have positive effects on the user experience as well, which translates into metrics that are harder to measure but are nonetheless of much importance to online & offline businesses, such as customer satisfaction and retention.

# Types of recommendation Engines:

1. Collaborative filtering
2. Content –Based filtering
3. Hybrid Recommendation System

# 2.1 Collaborative Filtering:

This filtering method is usually based on collecting and analyzing information on user’s behaviors,

their activities or preferences and predicting what they will like based on the similarity with other users to identify new user-item associations. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and thus it is capable of accurately recommending complex items such as movies without requiring an “understanding” of the item itself.Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. For example, if a person A likes item 1, 2, 3 and B like 2,3,4 then they have similar interests and A should like item 4 and B should like item 1.

Further, there are several types of collaborative filtering algorithms:

* **User-User Collaborative filtering:** Here, we try to search for lookalike customers and offer products based on what his/her lookalike has chosen. This algorithm is very effective but takes a lot of time and resources. This type of filtering requires computing every customer pair information which takes time. So, for big base platforms, this algorithm is hard to put in place.

“identifying similar users is key here”

* **Item-Item Collaborative filtering**: It is very similar to the previous algorithm, but instead of finding a customer look alike, we try finding item look alike. Once we have item look alike matrix, we can easily recommend alike items to a customer who has purchased any item from the store. This algorithm requires far fewer resources than user-user collaborative filtering. Hence, for a new customer, the algorithm takes far lesser time than user-user collaborate as we don’t need all similarity scores between customers. Amazon uses this approach in its recommendation engine to show related products which boost sales.
* **Other simpler algorithms**: There are other approaches like market basket analysis, which generally do not have high predictive power than the algorithms described above.

## 2.2 Content Based Filtering:

Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user.

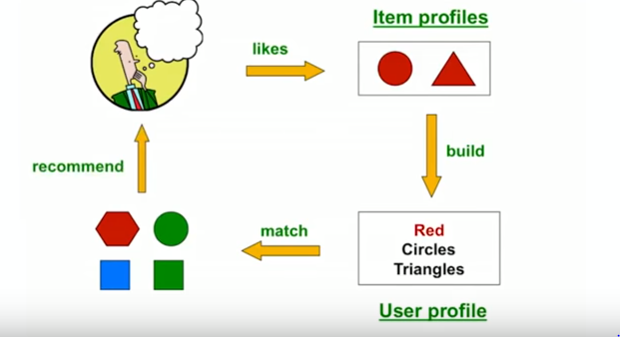
Several issues have to be considered when implementing a content-based filtering system. First, terms can either be assigned automatically or manually. When terms are assigned automatically a method has to be chosen that can extract these terms from items. Second, the terms have to be represented such that both the user profile and the items can be compared in a meaningful way. Third, a learning algorithm has to be chosen that is able to learn the user profile based on seen items and can make recommendations based on this user profile.

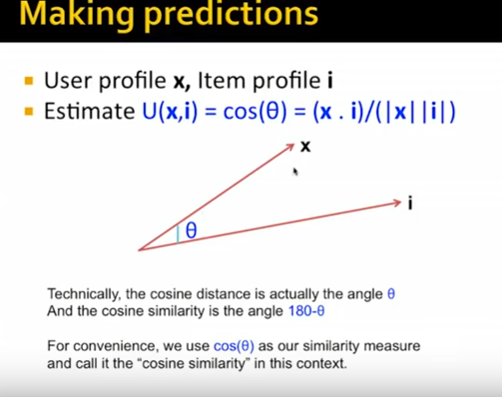
The information source that content-based filtering systems are mostly used with are text documents. A standard approach for term parsing selects single words from documents. The [vector space model](http://recommender-systems.org/vector-space-model/) and [latent semantic indexing](http://recommender-systems.org/latent-semantic-indexing/) are two methods that use these terms to represent documents as vectors in a multi dimensional space.

“Initially form the item profile and collect the user information and make user profile”

Simple user profile = mean(item profile)

User profile build on item profile..normally based on the temssss cegory profile

****

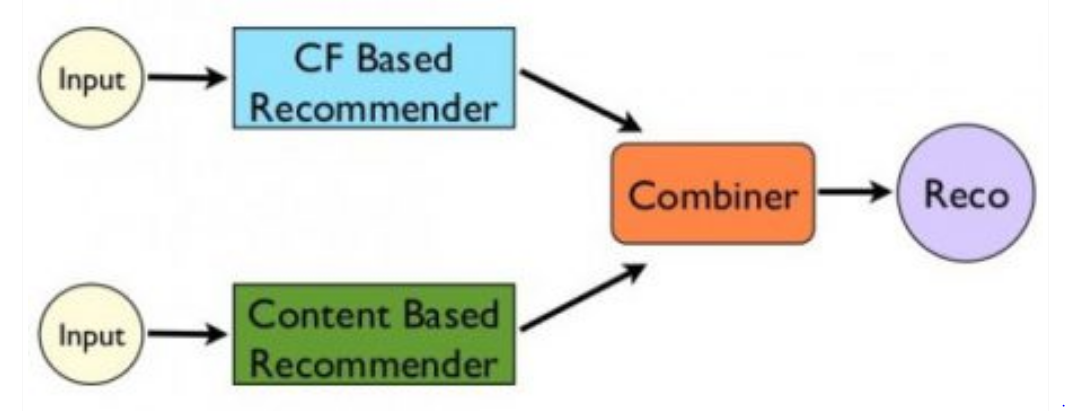


# 2.3 Hybrid Recommendation Systems :

[Recent research shows](http://dataconomy.com/2015/03/an-introduction-to-recommendation-engines/) that 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. Further, by adding content-based capabilities to a collaborative-based approach and vice versa; or by unifying the approaches into one model.

Several studies focused on comparing the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that hybrid methods can provide more accurate recommendations than pure approaches. Such methods can be used to overcome the common problems in recommendation systems such as cold start and the data paucity problem.

[Netflix is a good example of the use of hybrid recommender systems](https://en.wikipedia.org/wiki/Recommender_system). The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering).

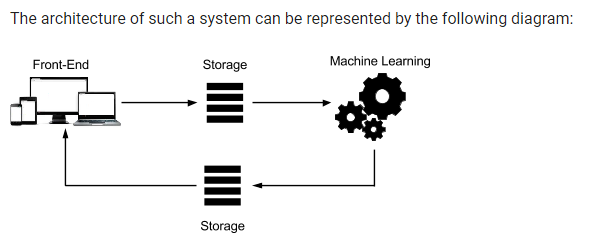


Note :

## 3.1 How Does Recommendation Engine Works:

According to the article [Using Machine Learning on Compute Engine to Make Product Recommendations](https://cloud.google.com/solutions/recommendations-using-machine-learning-on-compute-engine), a typical recommendation engine processes data through the following four phases namely collection, storing, analyzing and filtering.





# 3.2 Data Analysis:

* **Real-time systems** can process data as it’s created. This type of system usually involves tools that can process and analyze streams of events. A real-time system would be required to give in-the-moment recommendations.
* **Batch analysis** demands you to process the data periodically. This approach implies that enough data needs to be created in order to make the analysis relevant, such as daily sales volume. A batch system might work fine to send an e-mail at a later date.
* **Near-real-time analysis** lets you gather data quickly so you can refresh the analytics every few minutes or seconds. A near-real-time system works best for providing recommendations during the same browsing session.

<https://www.youtube.com/user/PyDataTV/search?query=recommendation>

# Matrix Factorization :

matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns

