**Recommendation Systems**

**Understanding Users**

**Explicit ratings**

* One way to understand your users or customers is through explicit feedback. For example, asking users to rate an online course like this one on a scale of one to five stars or rating content they see with a like or a thumbs up or a thumbs down.
* The problem with explicit ratings or feedback is that requires extra work from your users. Not everyone can be bothered to leave a rating, so this data tends to be very sparse. When your data is too sparse it leads to low quality recommendations.
* Another problem is that the same rating may mean different things between two different people.

**Implicit ratings**

* This is looking at the things you do anyhow and interpreting them as indications of interest or disinterest. For example, if you click on a link on a webpage we could consider that an implicit positive rating for that link and the content it points to or if you click on an ad it might tell the ad network that you might find other ads similar to that ad appealing.
* Advantage is that we do not have problems with data sparsity.
* Problem is that clicks are not always a reliable indication of interest. People often click on things by accident or lured in by clickbait-y images.
* Click data is also highly susceptible to fraud because there are a lot of bots out there on the internet doing bad things that can pollute your data.
* **Things you purchase** are a much better indication of interest. Resistant to fraud.
* **Things you consume** such as how many minutes you spend watching a video as an indicator of how much you like it. It requires a consumption of your time so it is also a reliable indicator of interest compared to click data.

[**Collaborative Filtering**](https://en.wikipedia.org/wiki/Collaborative_filtering#Types)

Types:

* **Memory-based**  
  Examples are neighbourhood-based, item-based, user-based top-N recommendations.
  + Pearson correlation similarity
  + Vector cosine based similarity
  + Locality-sensitive hashing to find similar users
* **Model-based**  
  Models are developed using different machine learning algorithms to predict users’ rating of unrated items.
  + Bayesian networks
  + Clustering models
  + Latent semantic models
    - Singular Value Decomposition (SVD)
    - Probabilistic latent semantic analysis
    - Multiple multiplicative factor
    - Latent Dirichlet Allocation (LDA)
    - Markov decision process
* **Hybrid**These overcome the limitations of native CF approaches such as sparsity and loss of information. However, increased complexity and are expensive to implement.
* **Deep Learning**

Most are not reproducible and some could be outperformed by simpler methods.

* + Some generalize traditional Matrix factorization algorithms
  + Some leverage new model types like Variational Autoencoders
* **Context-aware CF**

Tailor the recommendations to additional information that defines the specific situation under which recommendations are made. Contextual information such as time, location, social information, etc.

[**Content-based filtering**](https://en.wikipedia.org/wiki/Recommender_system#Content-based_filtering)

These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. Treat recommendation as a user-specific classification problem and learn a classifier for the user’s likes and dislikes based on an item’s features. These algorithms try to recommend items that are similar to those that a user liked in the past, or is examining in the present.

* Simple approaches use average values of the rated item vector
* Machine Learning techniques which estimate the probability that the user is going to like the item
  + Bayesian Classifiers
  + Cluster Analysis
  + Decision Trees
  + Neural Networks
* Opinion based system which utilizes information retrieval, sentiment analysis

**Multi-criteria recommender systems**

Recommender systems that incorporate preference information upon multiple criteria.

**Risk-aware recommender systems**

Considering the risk of upsetting the user by pushing recommendations in certain circumstances. The performance of the recommender system depends in part on the degree to which it has incorporated the risk into the recommendation process. DRARS

**Location based recommendation (Mobile systems)**

**Hybrid recommender systems**

Combining collaborative filtering, content-based filtering, and other approaches. Can be used to overcome some of the common problems such as cold start and sparsity problem.

Can be implemented in several ways:

* **Weighted**: Combining the score of different recommendation components numerically.
* **Switching**: Choosing among recommendation components and applying the selected one.
* **Mixed**: Recommendations from different recommenders are presented together to give the recommendation.
* **Feature Combination**: Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.
* **Feature Augmentation**: Computing a feature or set of features, which is then part of the input to the next technique.
* **Cascade**: Recommenders are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones.
* **Meta-level**: One recommendation technique is applied and produces some sort of model, which is then the input used by the next technique.

**Session-based recommender systems**

**Performance measures**

To measure the effectiveness of recommender systems, and compare different approaches, three types of evaluations are available:

1. User studies: A few dozens or hundreds of users are presented recommendations created by different recommendation approaches, and then the users judge which recommendations are best.
2. Online evaluations (A/B tests): recommendations are shown to typically thousands of users of a real product, and the recommender system randomly picks at least two different recommendation approaches to generate recommendations. Effectiveness is measured with implicit measures such as conversion rate or click-through rate.
3. Offline evaluations: based on historic data, e.g. a dataset that contains information about how users previously rated movies.

Commonly used metrics like RMSE. Information retrieval metrics such as precision, recall, or discounted cumulative gain (DCG) are useful to assess the quality of a recommendation method.

**Beyond accuracy**

* **Diversity** – Users tend to be more satisfied with recommendations when there is a higher intra-list diversity, e.g. items from different artists.
* **Recommender persistence** – In some situations, it is more effective to re-show recommendations, or let users re-rate items, than showing new items. There are several reasons for this. Users may ignore items when they are shown for the first time, for instance, because they had no time to inspect the recommendations carefully.
* **Privacy** – Recommender systems usually have to deal with privacy concernsbecause users have to reveal sensitive information. Building [user profiles](https://en.wikipedia.org/wiki/User_profiles) using collaborative filtering can be problematic from a privacy point of view. Many European countries have a strong culture of [data privacy](https://en.wikipedia.org/wiki/Information_privacy), and every attempt to introduce any level of user [profiling](https://en.wikipedia.org/wiki/Profiling_(information_science)) can result in a negative customer response. Much research has been conducted on ongoing privacy issues in this space. The [Netflix Prize](https://en.wikipedia.org/wiki/Netflix_Prize) is particularly notable for the detailed personal information released in its dataset. Ramakrishnan et al. have conducted an extensive overview of the trade-offs between personalization and privacy and found that the combination of weak ties (an unexpected connection that provides serendipitous recommendations) and other data sources can be used to uncover identities of users in an anonymized dataset.
* **User demographics** – Beel et al. found that user demographics may influence how satisfied users are with recommendations. In their paper they show that elderly users tend to be more interested in recommendations than younger users.
* **Robustness** – When users can participate in the recommender system, the issue of fraud must be addressed.
* **Serendipity** – [Serendipity](https://en.wikipedia.org/wiki/Serendipity) is a measure of "how surprising the recommendations are". For instance, a recommender system that recommends milk to a customer in a grocery store might be perfectly accurate, but it is not a good recommendation because it is an obvious item for the customer to buy.
* **Trust** – A recommender system is of little value for a user if the user does not trust the system. Trust can be built by a recommender system by explaining how it generates recommendations, and why it recommends an item.
* **Labelling** – User satisfaction with recommendations may be influenced by the labeling of the recommendations. For instance, in the cited study [click-through rate](https://en.wikipedia.org/wiki/Click-through_rate) (CTR) for recommendations labeled as "Sponsored" were lower (CTR=5.93%) than CTR for identical recommendations labeled as "Organic" (CTR=8.86%). Recommendations with no label performed best (CTR=9.87%) in that study.