Recommender System:

Individual Assignment 2 

*Hans Alberto Franke,* [*h.a.franke@students.uu.nl*](mailto:h.a.franke@students.uu.nl)

Personalization for Public Media

Applied Data Science - Utrecht University - 2021

**Main Objective**

Build a recommender system and interface taking in account user’s values like control, transparency and acceptance

**Clarification points for assignment 2**

1. The deadline for the assignment is Monday 29th 11.00 am. Please submit via SurfDrive, WeTransfer, Dropbox or other file sharing service via e-mail to both Shakila ([shakila.shayan@hu.nl](mailto:shakila.shayan@hu.nl)) and Koen ([koen.vanturnhout@hu.nl](mailto:koen.vanturnhout@hu.nl))
2. You can use the lab session time to have to ask your questions and get feedback on your final assignment. A signup sheet will be posted on teams.

**Short video presentation:** try to keep it around 5 minutes (= ideal length to motivate your choices)  
**Technical report**: maximum 1,500 words

Your task for the assignment is to develop a recommender system for a movie dataset. This means you are in charge of the whole cycle of events and components, from (1) the interface design, to (2) deciding on the metrics (explicit and/or implicit), (3) designing a way to control and measure the user’s interaction and input, (4)to building a working recommender system engine in the back-end and (5) finally an informed decision on presenting the results. So how you need to make informed decisions at the component level as well as the whole cycle, with a convincing justification for your choices.

As for data you can just work with the Movielens dataset, or combine it with any other dataset you have found and can work with to build your system. You can also expand your data through the extra metrics and measurements that comes from your target group, (which means creating additional columns in your table which will be populated while using the system).

For your interface it is enough to make a mockup prototype, you do not need to write a code for the frontend and the interface, but you need to provide a way to measure all the features and metrics you are working with in your recommender system engine on the interface and explain them in your technical report.

In making the above choices you have to think about

* The value(s) you are creating for your stakeholders and your audience
* The algorithmic affordances and proper use of the pattern libraries in your interface design
* The corresponding metrics and best ways to measure them and design for them to get the information from the user/audience
* The most suitable algorithm and similarity measurement for your recommender engine(code) with respect to the incoming input from the user and the type of data that you collect (whether it is ratings, or likes/dislikes, etc.., you might need to use different distance measurement for your similarity algorithms in your code for example)
* The presentation of the result on the interface with respect to all the previous choices.

Your submission has two parts; for the overall presentation of the system you submit a video explaining the above choices and the rational for it as well as the cycle of events as a whole. **For the technical report you briefly explain your back-end code and how it is working with the front end and the choices you make on the back-end and the reasons for particular choices and how it connects to the designed interface from a usability and technical perspective**. The word limit is only for the report and perhaps pieces of the code that you need to include to explain your decisions. You should not put the whole code in the technical report but submit it as an appendix together with screen designs of your recommender system.

The assessment of the research report will be based on the following criteria:

* Level of research: the overall work and its effectiveness in connecting design decisions to metrics, values, and the stakeholders.
* Practical implementation of algorithmic affordances: level of technical sophistication, logic of design and choices, feasibility.
* Level of reflection: academic discussion of the central values, motivation for technical choices made in the implementation of values as well as your discussion on the shortcomings and potentials for improvements and expansion
* Level of creativity and innovation: you are not of course needed to have an original and highly creative design, but any such efforts, no matter how small would be appealing and an extra bonus for the final assessment.

# Introduction

* Why is important to allow user control: based on values, like privacy, control, transparency?
* Main goal of my work?
* How kind of data I can collect and how users check transparency
* Metrics to measure this value (users)
* How many times a user change an input()

Recommender systems have been researched extensively over the past decades. Whereas several algorithms have been developed and deployed in various application domains, recent research efforts are increasingly oriented towards the user experience of recommender systems [1]. This research goes beyond accuracy of recommendation algorithms and focuses on various human factors that affect acceptance of recommendations, such as user satisfaction, trust, transparency and sense of control.

Many different stakeholders can use or affect a recommender system. These stakeholder have different values and desired outcomes. Which leads to the question of: How this can be connected and addressed in a single recommender system? How the interface can be used to answer for human values like **control**, **transparency** and **acceptance** and increase audience metrics like **usage** and **engagement**?

In this work, we present an interface that combines recommendation with visualization techniques to support human-recommender interaction.

# Methodology

* Which patterns used and why
* Define collaborative filters
* Define user based filters
* Why is important have both?
* Metrics: quick definition and connect with previous ideas!

The core job of analytics is to help companies gain insight into their customers. Then, the companies can optimize their marketing and deliver a better product. (Without analytics, companies are in the dark about their customers.) Analytics gives businesses the quantitative data they need to make better, more informed decisions and improve their services.

Recommender algorithms are often broadly categorized in three areas: *collaborative filtering* recognizes commonalities between users or between items on the basis of explicit (ratings, tags, etc.) or implicit (actions like reading, downloading.) relevance indications [2]. A standard user-based collaborative filtering algorithm first identifies similar users based on their overlapping interactions or similar ratings of common items. It then makes recommendations based on preferences of these similar users. A standard item-based recommendation algorithm analyzes similarities between items and then uses these similar items to identify the set of items to be recommended. Collaborative filtering is the most widely implemented and most mature technology [3]. *Content-based filtering* matches descriptions of items to descriptions of users [4]. They base their predictions on information about individual users and items, and ignore con- tributions from other users. This approach relates most closely to our work on metadata [5]. *Hybrid recommender systems* combine recommendation techniques, to gain better performance with fewer drawbacks [3].

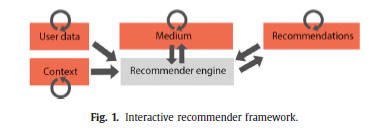
New approaches makes the user more relevant to provide inputs and context to a recommender, for example, providing feedback that is used to better predicting in the future. This is given, allowing user exert control over the algorithm, choosing your preferences or values for example. A teenager may be interested in more trand topics similar to his friend, but a adult on the other side may be interested in old very well rated movies of western genre.

EXPLAIN SOME USER VALUES!

Some well know values in literature are*,* ***Transparency***deals with the “black-box” nature of current recommender systems by explaining the inner logic of the system to end users. Similar to transparency, ***justification***helps users understand why they get certain recommendations, but it may not relate to the inner logic of the recommendation techniques. ***Controllability***strengthens user involvement by incorporating input and feedback from the end user into the recommendation process. User control can occur at any step of the recommendation process, such as providing ratings, adjusting preference data, and revising or exploring recommendations.

A very difficult value to handle is***Diversity****,* itrefers to providing recommendations with a relatively large coverage of the recommendation space [6]. For instance, it is important to recommend items that the user would prefer, but that are different from those which she has already purchased or experienced. When a new item or a new user joins a recommender system, the system has no prior knowledge about it, i.e., no item- feature data, no ratings, no preference information. The inability to make recommendation to new comers is called the ***cold start***problem. Acquiring ***contextual***information and incorporating it into recommendation processes in a flexible and fluid manner has gained increased interest over the past decades. The goal is to tailor recommendations to the current needs of the user.

There is many ways to asses these values, and the **Algorithmic Affordance pattern library** is a first attempt to give an overview over the possibilities for designers to give end-users more control over the outcomes of algorithms [6]. These patterns can show many different possibilities to allow, measure and improve recommender system with very well designed interfaces.



**Figure1**. Workflow of feedback from interaction of a **user** with **recommender.** Adapted from [1]

These patterns summarize many common patterns described in literature, therefore is a nice source to understand what is the state of art and have nice ideas of interface design. In this work the interface was design based on many patterns, the table below summarize all the approaches used in this work as well as reference from the authors [6].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pattern | Main Idea | How allow User Control | How do users understand the control? | How is used in My interface |
| Peek Picker | Users can switch between their own recommendations and those of peers which are further away. | They can browse recommendations for similar users (in some respect, but different in others) | This is a form of direct control of the algorithm. It also gives insights into the ‘inner workings’ of an algorithm. | Recommendations based on collaborative filter (selecting friends / genres ) |
| Multiple Profiles | User can select different profiles | Users are offered a choice between recommenders trained on different datasets. | Users are in a position to compare different profiles so that they can find out what works for them | Profile Change |
| Incognito | User can go “anonymous” not recording any statistics | Users tell the algorithm to temporarily ignore what they are doing. | Users may have the expectation of not sharing any data with the algoritm which may be at odds with the specific implementation. | Button Incognito on main page |
| Introduce Chance | Giving users the option to activate a chance effect into their recommendations leads to more diverse recommendations. | By activating some random process they increase diversity of the output. | They understand chance plays a big role so they may expect recommendation surprises. | Movies based on never watched genres, tags, users and random generator |
| Data Toggles | Switches may enable the user to chose which data is used by the algoritm to arrive at a recommendation. | Users are allowed to enable or disable certain information from being used by the algorithm. | Users may manipulate this information to build a dynamic model of what information is vital to their recommendation. | Allow user selection genres, actors, directors by list in the interface |
| Social Context | Social recommendation systems can be more transparent by indicating the social group their recommendations are based upon. | The algorithm offers context to a selection made by the user. | Users are aware the suggestions are context sensitive. | Show to user the friends, genres of which recommendation was based |
| Ordered List | By presenting top reccomendations in an ordered list, users can make a choice among multiple items that are recommended for them. | Users selections can be used as to tune the algorithm. | The user can choose among presented alternatives (and as such have the space to make a final decision, making the quality of the recommendation less critical). Users will mostly not be aware of the effects of a certain choice in training the algorithm. | Lists ranked by similarities and ratings |
| Liking Items | Users are given a lightweight control to express their opinion about an item, cumulatively resulting in feedback to the algorithm about user preferences. | Likes support filtering of messages in timelines. Users do not see the immediate effect of this. | Likes are primarily seen as feedback to the author of a tweet (who often gets notified) or facebook post, secundary as a message to the broader audience (social group) and only thirtiary as an algoritmic control (see Eslami et al. 2016). | Allow user rates movies (1-5) |
| Blacklist | Users can blacklist items, giving the algorithm an idea of their dislikes and preventing the algorithm from showing it again. | Blacklisting an item gives the recommender feedback about your dislikes. It also directly prevents the algorithm from delivering particular output. | Users may not have accurate ideas about the scope of the feedback they give through a blacklist action (e.g. single item or a whole categorie of content) | Users rates movies, so intuitively the algorithm will recommend less similar items. A user can filter a genre, so it will blacklist that genre from results. |
| Cold Star | Algoritms need basic information about users to be able to deliver their first recommendation. | Users can consciously decide which information they want to feed the algorithm (constrained by the questionnaire) | Users are aware the lanswers are used as an input abeit not so much on how the alogritm uses the input. | User can define during register page settings like: favorite genres, news or rated movies. |

**Table1**. Summary of patterns used in this work with connection how they are assessed on the interface

This work try to asses *Controllability* allowing user to explicitly and implicitly interact with recommender. Explicitly with the opportunity to choosing of parameters like preferred genre or actor during the navigation through the website or liking/disliking the recommendation presented, and implicitly by rating the movies after watching. This was the most relevant part of the work with many different patterns used to allow some form of control to user.

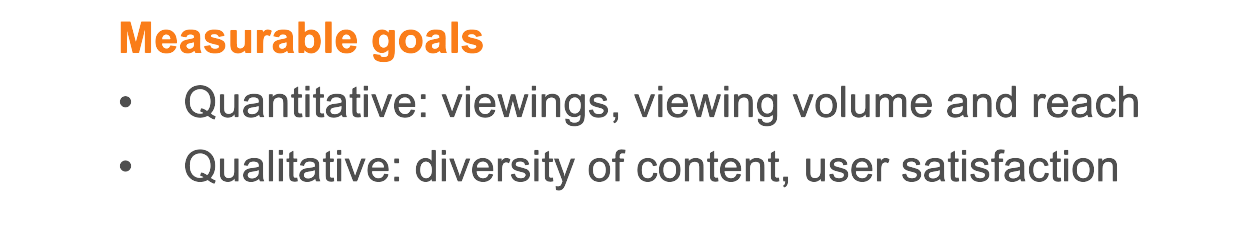
*Transparency* and *justify* were assessed but in an indirect way, with text showing to the user how the recommender was provided (i.e. based on movie you had watched or because of similar users watched), but not with high level detail (i.e what are the main similarities between users/movies). *Cold start* problem was answered in a login step with a user choosing 3 preferred genres, but as our system is static no approach to a new movies was assessed.

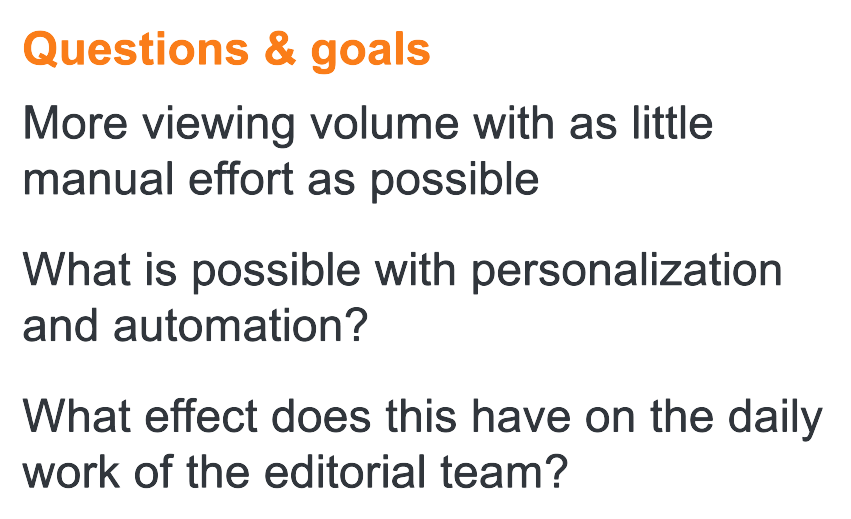
*Diversity,* was assessed allowing user receive random recommendations that he never watched, it can be a new genre or a new movie. In the login screen the user can chose if they value most his previous experiences compare to others and if he like most new content or liked content (i.e similar that what he has watched)

On the other side, a recommender must increase audience metrics. Crafting content that connects with its preferred audience requires understanding audience objectives – and how these objectives match enterprise goals. While often process can be as simple as soliciting direct feedback or taking requests, sometimes this ends up being more esoteric. Audiences may not *know* the kind of content that will have the most impact and fulfill a direct need – primarily because they may have never encountered content like it before. You can’t identify something that has yet to be named. To attempt to identify a need – naming the nameless, so to speak – start by mapping the journey your audience is taking involving your enterprise. Starting by aligning with your audience personas, use trend data to trace the lifecycle of a customer and their content needs.

# Recommender System and User’s Interface

* What kind of stakeholders I am addressing: Personas | how I will adapt to taking in account change of preferences over time
* Limitations? What I may improve in the future? Here or in conclusion?
* Clusters: trending, u may be interested in, …. Similar users like you
* Value / Metrics:
  + Diversity
  + Engagement
  + Control => % change of input / people change in login or not!
  + Views
  + User Satisfaction
  + Probability to predict correct?





* How I implement:
  + Simple algorithms: Weighted ratings
  + Content Based Recommender
  + Collaborative
  + Hibrid: The best choice!

**How the interface allow personas/users values be assessed / Control:**

Users can and should provide feedback through interface control. The interface was design focusing on assessing the main values: ***Controllability****,* ***Transparency***, ***justify*** *and* ***Diversity****.* This was design following the patterns suggested in [6]. The choices of genres, friends, tags, by search button, using anonymous mode are mechanisms to allow **control** in a **transparency** way, with users understand what is happening and with **acceptance** of the suggestions, once he participated in the recommendation through **explicit** (i.e rate, like/dislike, de-select of genre or movie) and **implicit** (i.e watching the movie, adding to his list, recommend to a friend. ) feedback .

Next sections are to describe what kind of techniques were used to model the recommender system.

**Metrics:**

On the other side of the interface there is a company that need some metrics to measure the success of the recommender, not only based on users perspective but in business as well. For example, **users** **metrics**: diversity, engagement and satisfaction.

So to address **users values** the metrics would be, % of selection of filters (i.e genres, tags, friends) this will assess how much **control** a user has**. Diversity,** can be measure with how many %random chances the user is trying in the lucky wheel. **Engagement**, can measure by how my suggestion to friends a user made, or how many days/week a user join the website. **Satisfaction** can be measure by how many positive or negative feedbacks a user made, or number of friends invited.

**Business metrics**: number or hours of views, % of correct prediction, coverage (how many genres a user has watched or % long tail movies), number of subscriptions. These metrics is not only for make more money, but they are important to measure if the user has his own metrics/values fulfilled, like a user that is increasing his average number of views probably has his values addressed.

The recommender need to take in account this metrics in the prediction, so lets define how it was implemented.

**Explanation of algorithms:**

## Simpler Recommender

One simple way to recommend a movie is based on ratings. One can user weighted ratings based on rates and number of votes (i.e. score). It can be build on “best” movies, genres, tags, and many different filters. Who never go to google and type: “best movies of all time?”, and the return is a list from **imbd** showing their famous TOP250 [ <https://www.imdb.com/chart/top/> ].

## Content Based Recommender

The recommender we built in the previous section suffers some severe limitations. For one, it gives the same recommendation to everyone, regardless of the user's personal taste. If a person who loves romantic movies (and hates action) were to look at our Top 15 Chart, s/he wouldn't probably like most of the movies. If s/he were to go one step further and look at our charts by genre, s/he wouldn't still be getting the best recommendations.

To personalize our recommendations more, an engine that computes similarity between movies based on certain metrics (i.e tags, genres, users views) and suggests movies that are most similar to a particular movie that a user liked. Since we will be using movie metadata (or content) to build this engine, this also known as **Content Based Filtering. The user can provide input to this part choosing in the interface his preferences.**

## Collaborative Filtering

Content based engine do not close all the gap, for it is only capable of suggesting movies which are close to a certain movie. That is, it is not capable of capturing **tastes** (i.e users preferences) and providing recommendations across genres. Also, the engine that we built is not really personal in that it doesn't capture the personal tastes and biases of a user. Anyone querying our engine for recommendations based on a movie will receive the same recommendations for that movie, regardless of who s/he is.

Therefore, a technique called **Collaborative Filtering** is based on the idea that users similar to me can be used to predict how much I will like a particular product or service those users have used/experienced but I have not.

In this work were used the **Surprise** library that used extremely powerful algorithms like **Singular Value Decomposition (SVD)** to minimize RMSE (Root Mean Square Error) and give great recommendations. The algorithm was improved with *gridSearch* (selection of the best parameters and errors metrics). User can provide feedback selecting friends for example, telling, well I weighting more this user than another.

## Hybrid Recommendation

Merging the 3 previous approaches in one single recommendation is the best scenario to take in account many different criteria’s to fully satisfy a user. For example, we can start by assessing similarities based on others users (collaborative filter), then search the movies with the 10 most similar users (content based), filtering for the 3 target\_user preferred genre (content based as well), ranking the movies by others users score (basic filter, weighted ratings). The final list of movies is the input for the SVD to predict based on that list which movies will have the most chance to asses the target user preferences.

Using this techniques we are assessing the users values (user can change parameters of the algorithm explained on the interface and Methodology chapter), and assessing the metrics of the recommender like: user engagement, increasing views, and one of the most important: **user** **satisfaction** because we are increasing the chance to predict correctly. For example, one can filter which friend he/she want to compare, select genres that want to include, the level of similarities (own x others), and provide feedback in each iteration (i.e accept or refuse the recommendations!).

# Conclusion

* My measures were ok?
* Which tensions I take care:
* Which tensions should focus on next working

In this work, was designed an interface using library patterns to asses **values**, allowing **control** over the recommender. The interface is crucial, for is the connection point from end\_user to your recommender, through her that you are explaining to your users how you are building the recommendations and more important take his own **preferences** into account for future recommendations.

It was built 4 different recommendation engines based on different ideas and algorithms. They are as follows:

1. **Simple Recommender:** This system used overall Vote Count and Vote Averages to build Top Movies Charts, in general and for a specific genre. The weighted Rating System was used to calculate ratings on which the sorting was finally performed.
2. **Content Based Recommender:** Built two content based engines; one that took movie title and users as input and the other which took metadata such as tags and genres to come up with predictions. We also devised a simple filter to give greater preference to movies with more votes and higher ratings.
3. **Collaborative Filtering:** We used the powerful Surprise Library to build a collaborative filter based on single value decomposition. The RMSE obtained was less than 0.9 and the engine gave estimated ratings for a given user and movie.
4. **Hybrid Engine:** We brought together ideas from content and collaborative filtering to build an engine that gave movie suggestions to a particular user based on the estimated ratings that it had internally calculated for that user.

The work assessed metrics of **users** and **business**, trying to taking in account inevitable tensions like: more views or more diversity? Metrics sometimes contrary to each other, but the tension was minimized through transparency improving user acceptance and in a natural way increasing views because simply the algorithm make better predictions, once it used users inputs and control.

# Bibliography

|  |  |
| --- | --- |
| [1] | D. P. ,. V. Chen He, "Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities," 2016. |
| [2] | R. Burke, "Evaluating the dynamic properties of recommendation algorithms.," *In Proceedings of fourth ACM conference on recommender systems,* no. doi: 10.1145/1864708.1864753, 2010. |
| [3] | R. Burke, "Hybrid recommender systems: Survey and experiments. User Modeling and User Adapted Interaction," 2002. |
| [4] | M. J. Pazzani and D. Billsus, "Content-based recommendation systems," no. http://www.springerlink.com/index/qq35wt68l6774261, 2007. |
| [5] | S. Ternier, K. Verbert, G. Parra, B. Vandeputte, J. Klerkx, E. Duval and X. Ochoa, "The ariadne infrastructure for managing and storing metadata.," *IEEE Internet Computing, 13 ,* no. doi: 10.1109/MIC.2009.90, p. 18–25, 2009. |
| [6] | "Pattern Library," no. https://aapatternlibrary.wordpress.com/. |
| [7] | Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," 2020. |
| [8] | T. Asikis, J. Klinglmayr, D. Helbing and E. Pournaras, "How value-sensitive desing can empower sustainable consumption," no. https://doi.org/10.1098/rsos.201418, 2021. |
| [9] | H. Cramer, J. G. Gathright, A. Springer and S. Reddy, "Assessing and addressing algorithmic bias in practice," *https://doi.org/10.1145/3278156,* 2018. |
| [10] | J. Davis and L. Nathan, Value Sensitive Design: Applications, Adaptations, and Critiques, 2015. |
| [11] | R. Dobbe , S. Dean, T. Gilbert and N. Kohli, "A Broader View on Bias in Automated Decision-Making: Reflecting on Epistemology and Dynamics," no. https://doi.org/10.14763/2020.4.1534, 2018. |
| [12] | B. Friedman, Human Values and the Design of Computer Technology, Cambridge University Press, 1997. |
| [13] | S. Kheirandish, M. Funk, S. Wensveen, M. Verkerk and M. Rauterberg, "HuValue: a tool to support design students in considering human values in their design," *International Journal of Technology and Design Education,* no. https://doi.org/10.1007/s10798-019-09527-3, p. 30:101, 2020. |
| [14] | J. Simon, P.-H. Wong and G. Rieder, "Algorithmic bias and the Value Sensitive Design approach," 2020. |
| [15] | StakeholderMap, "https://www.stakeholdermap.com/retail-stakeholders.html". |

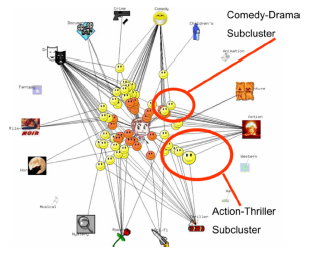
# Appendix:

## Code

## Interface

# Patterns

## Pattern 12: “Peer Picker”



**Main idea?** Recommendation systems can offer users to switch between their own recommendations and those of peers which are further away.

**Examples?** Still to be found in the wild.

**Input or output?** Input.

**How do users control the algorithm?** They can browse recommendations for similar users (in some respect, but different in others)

**How do users understand the control?**This is a form of direct control of the algorithm. It also gives insights into the ‘inner workings’ of an algorithm.

**Related patterns?**[Multiple Profiles](https://aapatternlibrary.wordpress.com/2021/02/18/multiple-profiles/)

**Academic research?**[Investigated by O’Donnovan et al. (2008)](https://sites.cs.ucsb.edu/~holl/pubs/ODonovan-2008-CHI.pdf)

## [Pattern 11: “Multiple Profiles”](https://aapatternlibrary.wordpress.com/2021/02/18/multiple-profiles/)

[](https://aapatternlibrary.wordpress.com/2021/02/18/multiple-profiles/)

**Main idea?** By allowing users to make multiple profiles they can tune the algoritms to specific behaviors, preferences or family members.

**Examples?** Netflix allows users to specify “Who’s watching”

**Input or output?** Input.

**How do users control the algorithm?** Users are offered a choice between recommenders trained on different datasets.

**How do users understand the control?**Users are in a position to compare different profiles so that they can find out what works for them

**Related patterns?**[Data Toggle](https://aapatternlibrary.wordpress.com/2021/02/17/pattern-6-enabling-inputs/), [Reset](https://aapatternlibrary.wordpress.com/2021/02/18/pattern-7-explanations/)

## [Pattern 10: “Incognito”](https://aapatternlibrary.wordpress.com/2021/02/18/pattern-10-incognito/)

[](https://aapatternlibrary.wordpress.com/2021/02/18/pattern-10-incognito/)

**Main idea?** User can be enabled to watch things without the algorithm learning from it with an incognito functionality

**Examples?** Browsing “incognito” in Google Chrome.

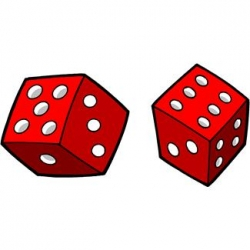
**Input or output?** Input

**How do users control the algorithm?** Users tell the algorithm to temporarily ignore what they are doing.

**How do users understand the control?**Users may have the expectation of not sharing any data with the algoritm which may be at odds with the specific implementation.

**Related patterns?**[Multiple profiles.](https://aapatternlibrary.wordpress.com/2021/02/18/multiple-profiles/)

## [Pattern 9: “Introduce Chance”](https://aapatternlibrary.wordpress.com/2021/02/18/pattern-9-introduce-chance/)

[](https://aapatternlibrary.wordpress.com/2021/02/18/pattern-9-introduce-chance/)

**Main idea?** Giving users the option to activate a chance effect into their recommendations leads to more diverse recommendations.

**Examples?**Google’s “I feel lucky” button is an early ancestor. The “[Spin the Reel](https://www.youtube.com/watch?v=4VsrS6HMX4M&feature=youtu.be)” concept also builds on this idea.

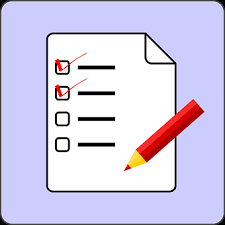
**Input or output?**  
Output.

**How do users control the algorithm?** By activating some random process they increase diversity of the output.

**How do users understand the control?**They understand chance plays a big role so they may expect recommendation surprises.

**Related patterns?**[Reset.](https://aapatternlibrary.wordpress.com/2021/02/18/pattern-7-explanations/)

## [Pattern 6: “Data Toggles”](https://aapatternlibrary.wordpress.com/2021/02/17/pattern-6-enabling-inputs/)

[](https://aapatternlibrary.wordpress.com/2021/02/17/pattern-6-enabling-inputs/)

**Main Idea?** Switches may enable the user to chose which data is used by the algoritm to arrive at a recommendation.

**Examples?** Still to be found in the wild. (example in [this](https://www.youtube.com/watch?v=XKeGvW_oDhE&feature=youtu.be) demo )

**Input or output?**Input

**How do users control the algorithm?** Users are allowed to enable or disable certain information from being used by the algorithm.

**How do users understand the control?**Users may manipulate this information to build a dynamic model of what information is vital to their recommendation.

**Related patterns?**[Decision paths](https://aapatternlibrary.wordpress.com/2021/02/18/pattern-13-descision-paths/)

## [Pattern 4: “Social Context”](https://aapatternlibrary.wordpress.com/2021/01/11/pattern-4-social-context/)

[](https://aapatternlibrary.wordpress.com/2021/01/11/pattern-4-social-context/)

**Main idea?** Social recommendation systems can be more transparent by indicating the social group their recommendations are based upon.

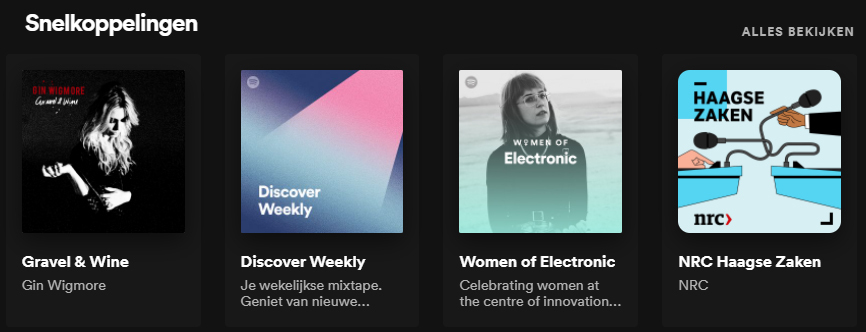
**Examples?** Many bookstores suggest items that others have looked at.

**Input or output?**Output.

**How do users control the algorithm?** The algorithm offers context to a selection made by the user.

**How do users understand the control?**Users are aware the suggestions are context sensitive.

## [Pattern 2: “Ordered List”](https://aapatternlibrary.wordpress.com/2021/01/05/pattern-2-ordered-list/)

[](https://aapatternlibrary.wordpress.com/2021/01/05/pattern-2-ordered-list/)

**Main idea?** By presenting top reccomendations in an ordered list, users can make a choice among multiple items that are recommended for them.

**Examples**? Spotify and Netflix present many of these.

**Input or output?**Both.

**How do users give input to the algorithm?** Users selections can be used as to tune the algorithm.

**How do users understand the control?**The user can choose among presented alternatives (and as such have the space to make a final decision, making the quality of the recommendation less critical). Users will mostly not be aware of the effects of a certain choice in training the algorithm.

**Related patterns?** Navigation-scapes

## [Pattern 1 “Liking Items”](https://aapatternlibrary.wordpress.com/2021/01/05/twitter-like/)

[](https://aapatternlibrary.wordpress.com/2021/01/05/twitter-like/)

**Main idea?**Users are given a lightweight control to express their opinion about an item, cumulatively resulting in feedback to the algorithm about user preferences.

**Examples**? Twitter Like (heart) or Facebook Like (thumbs-up)

**Input or output?**Input.

**How do users control the algorithm**? Likes support filtering of messages in timelines. Users do not see the immediate effect of this.

**How do users understand the control**? Likes are primarily seen as feedback to the author of a tweet (who often gets notified) or facebook post, secundary as a message to the broader audience (social group) and only thirtiary as an algoritmic control.

**Related patterns**? [Curated Lists](https://aapatternlibrary.wordpress.com/2021/01/09/pattern-3-users-curated-lists/)