

Recommendation System

A Group Project Utilizing Machine Learning

Gene Olafsen

A Software-based Project to Kickoff 2021

The self-driving car and reinforcement learning projects were both hardware-based endeavors.



Development Environment

- ▶ Python
- ▶ PyTorch
- ▶ Git
- ▶ Gitlab
- ▶ Docker
- ▶ VSCode

Resources

- ▶ [VSCode, Dev Containers and Docker: moving software development forward - Sticky Bits - Powered by Feabhas](#)
- ▶ [Developing inside a Container using Visual Studio Code Remote Development](#)

Recommendation Systems

- ▶ Three types of recommender systems are:
 - ▶ Content-Based Filtering
 - ▶ Collaborative Filtering (CF).
 - ▶ Knowledge-based System

Content-based Filtering

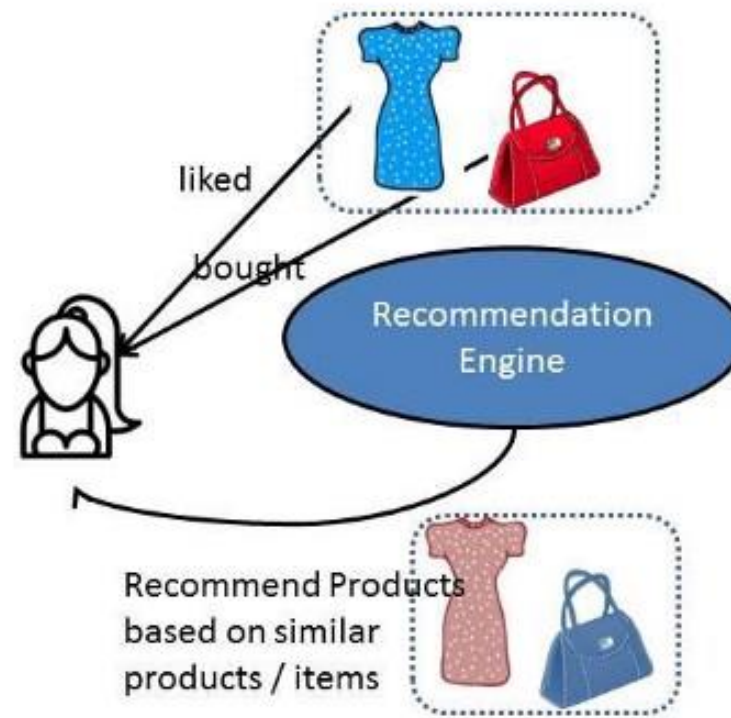
- ▶ Content-based filtering is *based on a single user's interactions and preference*.
- ▶ Recommendations are based on the data/metadata that is collected from a user's history and interactions.
- ▶ This system returns information such as products or services that relate to a user's likes or views.
- ▶ The more information that the user provides, the higher the accuracy of the recommendations.

Collaborative Filtering

- ▶ Collaborative filtering casts a wider net than content-based filtering.
- ▶ Such a system utilizes information collected from the interactions of many other users to derive suggestions for a user. This approach makes recommendations based on other users with similar tastes or in similar situations.
- ▶ Collaborative filtering uses the opinions and actions of others to recommend items to you or to identify how one product may go well with another. ‘You may also like...’ recommendations is a typical usage.



(a). Collaborative Filtering



(b). Content Filtering

Knowledge-based System

- ▶ Knowledge-based systems produce suggestions that are based on an influence about a user's needs and utilize a degree of domain expertise and knowledge.
- ▶ Rules are defined that set context for each recommendation.
- ▶ Such systems do not, by default, have to use the interaction history of a user. (as content-based systems do- but can use this information)

Cold Starts

- ▶ Most recommendations systems have at least one of the two *cold start* issues:
 - ▶ Visitor cold start
 - ▶ Product cold start

Visitor Cold Start

- ▶ Visitor Cold Start means that a new user is introduced in the dataset.
- ▶ Since there is no history of that user, the system does not know the preferences of that user. It becomes harder to make recommendations for that user.
- ▶ Each system uses different tricks to try to *place* the user in a *space* that can be *closely* identified with similar users. There is actually the concept of *distance* in these matrices which can be exploited find an appropriate fit for each user- what you need to be able to do, though, is create enough dimensions with your captured features to triangulate this location.

Product Cold Start

- ▶ Product Cold Start means that a new product is launched in the market or added to the system.
- ▶ User action is most important to determine the value of any product. The more the 'interaction' a product receives, the easier it is for a model to recommend that product to the right user.

System Startup

- ▶ Content-based Filtering
 - ▶ Personal metadata and individual transactional data can be missing at the outset.
 - ▶ Cold-start issues apply.
- ▶ Collaborative Filtering
 - ▶ Such systems usually have higher accuracy than content-based filtering.
 - ▶ Without meaningful information on others, it is difficult to be effective with any single person actions.
- ▶ Knowledge-based Systems
 - ▶ Given the way the system is built up, the recommendations can be easily explained.
 - ▶ Building up this type of framework can be expensive.
 - ▶ Tends to be better suited to complex domains where items are infrequently purchased or data is lacking.
 - ▶ Doesn't suffer the same cold-start up problems as others above.

Recommender System Challenges

- ▶ **Sparsity of data**

- ▶ Rows of values that contain blanks or zero values.

- ▶ **Labelling**

- ▶ Products with different labelling can be ignored or incorrectly consumed.

- ▶ **Scalability**

- ▶ A challenge with very large data sets which can lead to reduced performance.

Design Goal

- ▶ Quick start.
- ▶ Configuration allows data to be consumed and processed to produce recommendations quickly.