

# COLD START SYSTEM

Syeda Amna Amir

[https://github.com/S-Amna-Amir/Coldstart\\_Recommendation\\_System](https://github.com/S-Amna-Amir/Coldstart_Recommendation_System)

## PART 1 - Design

### 1. Assumptions:

I am assuming if the data for the users is not available, then at least minimal data about the products is. This means, item name, price, category etc. Moreover, I am assuming a broad online marketplace platform such as Amazon, or Daraz.

I am not relying on any sort of long term user data to be available, such as cookie data, saved items, recommending products to similar users, their age or demographic statistics etc.

### 2. Recommendation Approach:

- a. When no user data exists, the recommendation is based purely on the item's data. For example items could be recommended on the basis of:
  - i. Items designed for a wider and more generic audience, as compared to a specific niche
  - ii. Staple everyday necessities over luxury items
  - iii. Items with affordable prices over highly expensive items
  - iv. Simplistic items designed for first time users
  - v. Items from highly different categories such as technology, food, health and fitness, etc.
  - vi. Items designed for gender neutral audiences such as water bottle, keychain etc, over gender specific products such as necklaces etc.

- b. When minimal session level context is available then more user-curated recommendations can be shown. The user-preferred items and broader categories of products could be on the basis of:

- i. Search history
  - ii. Time spent on viewing a clicked item
  - iii. Product filters applied
  - iv. Clicked items

Each of the above would rerank the displayed items for the user.  
The displayed items would then be majorly a mix of items from the

user preferred category, while a few newer items from unexplored categories would also be displayed.

### 3. Fallback Behaviour

In the instance when the user scrolls for too long, but makes no interaction, then the products would be shown on the basis of the as-yet unseen items from broader categories. When a user opens the page, the most popular items are shown from a mix of different categories such as technology, entertainment, accessories, food, etc. Once the user has scrolled through the first items from each category but has not clicked on any of it yet, then different unseen products from diverse categories could be shown.

For example we show a smartwatch from the tech section, pizza from the food section, fiction book from the entertainment section, and a keychain from the accessories section. The first time recommendations should preferably be more economical. In the next round, different items from all categories would be displayed. For example, a laptop from the tech section, a cake from the food section, board game from the entertainment section, and a ring from the accessories section. And so on. The items displayed are marked as seen, and unseen items are preferred over seen items.

The items could be varied based on their characteristics such as price, audience gender, audience age etc. This ensures the items shown are not completely random, and even no interaction counts as interaction.

### 4. Explainability

The explanations would be generated based on the way the item had been chosen, for example:

- a. “Related to what you just viewed” for when user clicks are available
- b. “New this week” for cold start recommendations
- c. “Popular among new users” for first time users

### 5. Limitations

This system could perform poorly for users with highly specific needs and tastes, showing items where the item data is unavailable, products are not classified according to categories as required above, or the number of products in different categories is not balanced.

Over time, as more data about the items would be available, we could categorize them according to popularity among new users, user preferences in cookies, assumed user demographics such as age, gender etc, and show items with a higher heuristic score, compared to others. With time, the items could be made clickable, and session-level data such as time could be stored and used. A search bar, or

filters could be added. The data from these fields would also be stored and used in recommendations.

## PART 2 - Implementation

The code is on the following link:

[https://github.com/S-Amna-Amir/Coldstart\\_Recommendation\\_System](https://github.com/S-Amna-Amir/Coldstart_Recommendation_System)

### Technology Stack

Language: Python

Frontend: HTML

Backend: Flask

### Commands to run project:

1. .venv\Scripts\activate.bat
2. python app.py
3. Minimal UI: <http://127.0.0.1:5000/>