IT492: Recommendation Systems

Lecture - 01

Introduction

Arpit Rana

17th Jan 2022

Course Logistics

Course Logistics

Instructor	**Arpit Rana** Room-3105, Faculty Block-3 Email: arpit_rana@daiict.ac.in
Teaching Assistants	**Shruti Singh** (201301452@daiict.ac.in) **Krunal Botadara*** (202111009@daiict.ac.in)
Prerequisites	IT216, IT623 (Algorithms & Data Structures) IT565, IE406, (Machine Learning) Basic Python Programming

*subject to change…

Course Logistics

Credit Weighting	4
Lectures	Monday: 10:15 – 11:30 Friday: 08:30 – 09:45
Labs	Wednesday: 14:00 – 16:00
Private Study	At least 5 hrs per week

Course Logistics

Assessment	Mid-term: 25% End-term: 35% Course Projects: 40% (10 x 4) **Extra Credit(EC): 3%**
How to Fail	Skip lectures; avoid private study; cram just before the exam; expect the exam to be a memory test; be inactive on discussion forum
How to Pass	Attend lectures; summarize the notes; expect a problem-solving exam; **be active and accurate on discussion forum**

Course Logistics

Assignment Submission	**Project submissions:** ▪ Project submissions will be online through github (instructions will be provided in lab). ▪ Projects up to 48 hrs late will be given a 40% penalty. **The following constitute plagiarism on project submissions:** ▪ Copying any segment of code from any source ▪ Submitting code that you did not write yourself personally Students suspected of plagiarism on an assignment will be given a **ZERO**.

Tentative Course Plan

Introduction to Recommender Systems	Definition, objectives, components, approaches, and challenges	2
Recommendation Techniques: Collaborative Filtering	Neighborhood-based Collaborative Filtering (User-User, Item-Item, Graph-based) Model-based Collaborative Filtering (Latent Factor Models: MF and its variants, , SLIM)	8
Recommendation Techniques: Content-based Filtering	Content-based Recommendation, Content-based vs. Collaborative Filtering	4
Recommendation Techniques: Hybrid Techniques	Ensemble-based and Hybrid Recommendation -- Switching hybrids, Weighted hybrids, and Cascade hybrids	4
Re-ranking Approaches	Re-ranking for Diversity, Explainability, etc.	3

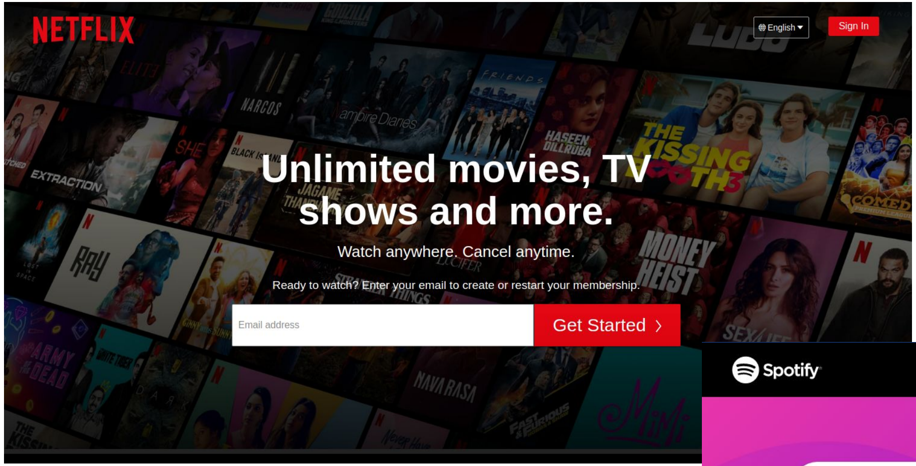
Tentative Course Plan

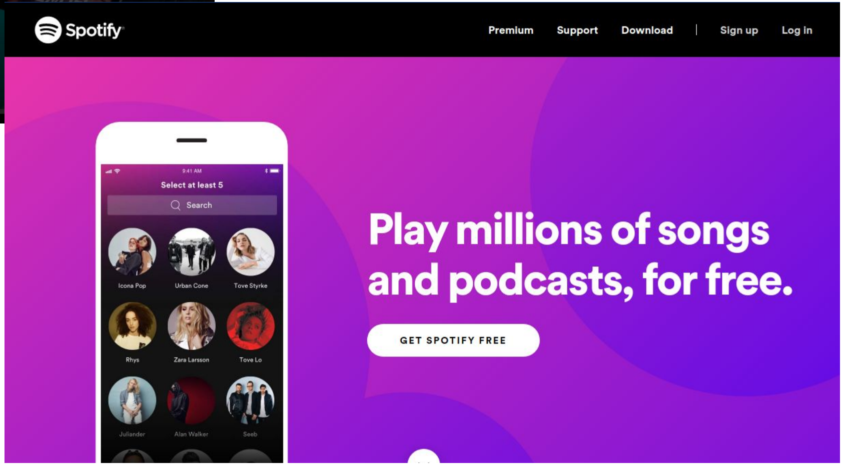
Evaluation	Evaluation Paradigms: User Studies, Online, and Offline	4
Advanced Topics* in Recommender Systems	Context-sensitive Recommendations	4
	Time-sensitive Recommendations	4
	Conversational Recommendations	4
	Explaining Recommendations	5

*Advanced Topics will be covered in accord to the availability of lectures.

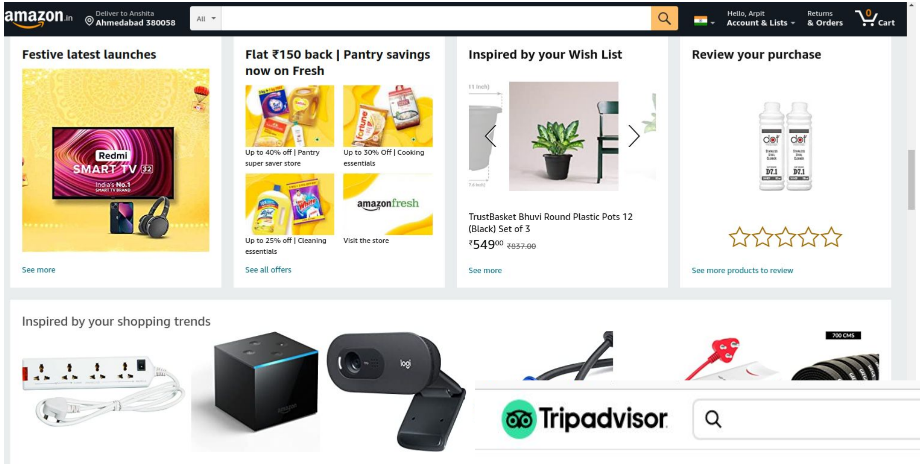
Introduction

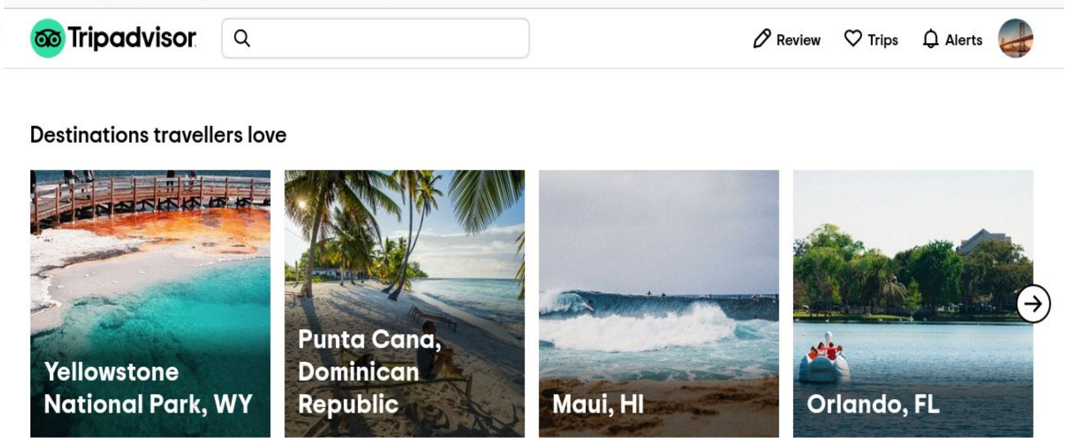
Recommender Systems



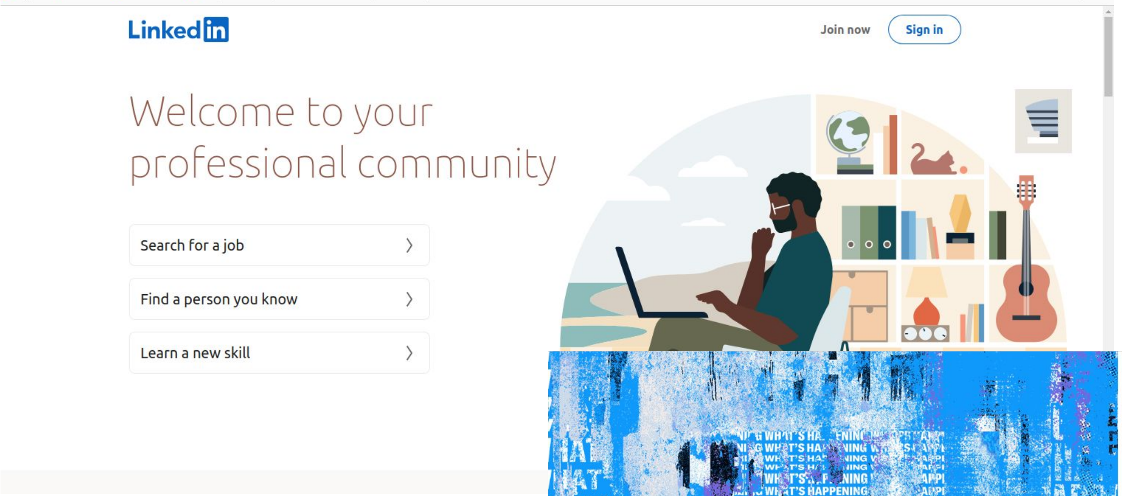


Recommender Systems

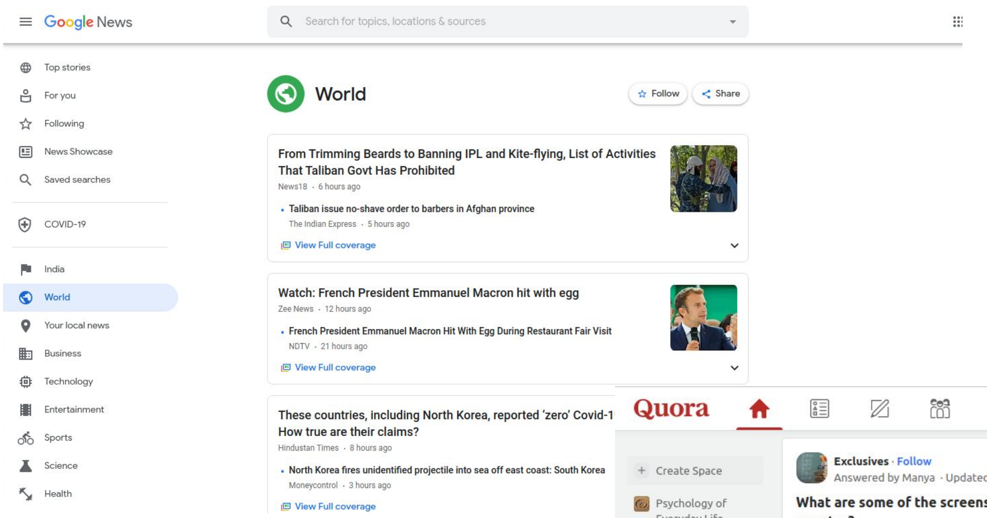


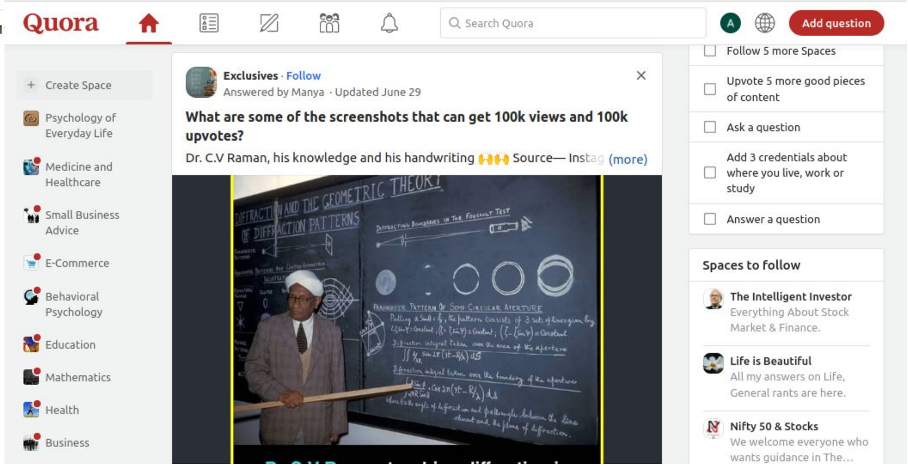


Recommender Systems



Recommender Systems





Definition

Recommender Systems

- *collect* data about user behaviour,
- *infer* the user’s preferences from her behaviour, and
- *suggest* items that they think will match these inferred preferences.

Definition

Recommender Systems

- Recommender systems do not make choices for the user.
- Instead, recommender systems help the user to manage choice.

- Question: Why not *browse*?
- Question: Why not *search*?

Why Recommender Systems?

Ideally, from users' perspective, to create joy

- Alleviate choice overload
- Offer better customer experience

Why Recommender Systems?

Business Objective: Increase Revenue

- Increase sales,
- Increase profit,
- Increase the number of customers,
- Retain existing customers,
- Improve *cross-selling* and *up-selling,*
- Increase repeat visits, and so on.

Upselling is a sales strategy that involves encouraging customers to buy a higher-end version of a product than what they originally intended to purchase.

Cross -selling : sell (a different product or service) to an existing customer.

Recommendations

- ***Personalized*: as per the collected user information:**
  - her tastes, interests, preferences;
  - her personality;
  - her long-term goals; and
  - her skills, knowledge.

Recommendations

- ***Contextualised*: as per the user’s circumstances:**
  - the time;
  - the location (physical or virtual);
  - the weather conditions;
  - the user’s companions;
  - her mood; and
  - her short-term goals.

Basic Formulation

Let $I = \{i_1, i_2, i_3, \ldots, i_n\}$ be a set of ***items*** and

$U = \{u_1, u_2, u_3, \ldots, u_m\}$ be a set of ***users***.

A recommender system attempts to find an item $i^* \in I$ for user $u \in U$ such that the ***utility*** of item i^* for user u, □(u, i^*), is maximum:

$$i^* = \arg\max_{i \in I} \; □(u, i)$$

Items

- **Physical products**, e.g. books, phones, laptops;
- **Non-physical products**, e.g. movies, music, ringtones, ebooks;
- **Services**, e.g. a hotel to stay in, a restaurant, a school or university;
- **People**, e.g. a person to ‘friend’ or ‘follow’, an expert (e.g. a plumber, a dentist);
- **Sources of information**, e.g. news stories, web pages, a blog to read, recipes, lessons, tutorials;
- **Events, actions and activities**, e.g. a museum to visit, a concert to go to, a job to apply for, an exercise regime to follow;

Item Features

- **Structured:** a finite and typically small set of attributes
  - e.g. For products: size, weight, manufacturer, etc.
    for movies: director, length, language, guidance certificate, etc.
    for songs: artist, producer, record label, etc.
- **Unstructured:** no explicit structure, often processed to obtain meaningful information
  - e.g. Keywords extracted from a movie description or user reviews; user assigned tags to an item;
- **Semi-structured:** mixture of structured and unstructured information
  - e.g. movie genres (comedy, thriller, romance, …) with movie keywords

Users

- **A Single User,**
- **A Small Group of Users**, e.g. friends, family members, colleagues;
- **A Large Group of Users,** e.g. communities

Features: In systems where users must create an account, the values of features can be obtained during the sign-up process,

For example,

- demographic features, such as sex, age, level of education;
- interests, maybe as categories (given by domain experts) or as keywords.

User-Item Interaction

This records how users have interacted with items in the past, e.g. *clicks, shares, likes, downloads, purchases, ratings, reviews, . . .*

- A user opinion is characterized as ***explicit*** or ***implicit*** feedback.
  - Directly stated opinions are ***explicit*** feedback,

    e.g., a star rating between 1 and 5 stars;
    a binary rating: +/-   or   like/dislike   or   ^/v;
    a binary comparison: item A is preferred over item B.

  - ***Implicit*** feedback is derived from user's other interactions with the system. Typically, they do not contain negative observations.

    e.g. inferring preferences from purchase actions, from clicks, from dwell-time, from consumption frequency.

Recommendations are *domain-specific*

What applies in one domain may not apply in another domain -

- **The unit of recommendation:**
  - individual items, packages, sequences (e.g. playlists, tours).
- **The target consumer:**
  - individual users, small groups (e.g. families, housemates), larger groups (occupants of a shared space, communities).
- **Level of interaction:**
  - passive, confirmation (e.g. skipping a song), selection from a list.

Recommendations are *domain-specific*

What applies in one domain may not apply in another domain -

- **The nature of the item:**
  - *high-value* versus *low-value*;
  - *high consumption cost* versus *low consumption cost*;
  - *rivalrous* versus *non-rivalrous*;
  - *perishable* versus *non-perishable*;
  - *one-off consumption* versus *repeated consumption*. . .

Recommendations are *domain-specific*

What applies in one domain may not apply in another domain -

- **The nature of the recommendation, e.g.:**
  - items that could be ***alternatives*** to the one the user is viewing;
  - items that are ***complementary*** to the one the user is viewing;
  - items that might ***come next*** after consuming the item the user is consuming. . .

Recommender System Architecture (RSA)

Recommender systems typically proceed through (at least) three steps:

- Candidate generation
- Scoring
- Top-*N* recommendation

RSA: Candidate Generation

Since there are so many items, we choose a smaller subset of candidate items depending on the context.

Examples:

- Candidates for user u might be her un-rated items, i.e. items i for which $r_{ui} = null$. This has two potential problems. What are they?
- For "linear TV", candidates might be programmes that are being broadcast this evening.
- For movie-going, candidates might be films that are being screened at the user's multiplex this week.
- For online news, candidates will be recent stories.
- For on-the-go travel (e.g. restaurants, hotels and points-of-interest), candidates must be nearby and open.

RSA: Scoring

Each candidate item is scored, e.g. for how relevant it is to this user, allowing the candidates to be ranked in order of decreasing score.

There are many ways of doing this.

- ***Collaborative methods*** recommend items that either users with similar tastes liked in the past or that, according to the other users, are similar to items that are liked by the active user.
- ***Content-based methods*** recommend items which, according to the item descriptions, are similar to items that are liked by the active user.
- ***Hybrid methods*** combine collaborative and content-based methods.

RSA: Top-N recommendation

The last step is to select the N candidates whose scores are highest and recommend these to the user.

Additional criteria to take into account at this stage -

- ***Business rules:*** e.g., there may be some items the business is trying to push (e.g. think about sponsored content).
- ***Ensemble/Hybrid recommendations***: combine scores of more than one recommender model
- ***Re-ranking:*** to ensure the degree of *diversity* or some notion of *fairness*.

IT492: Recommendation Systems

Next lecture -

Collaborative Methods