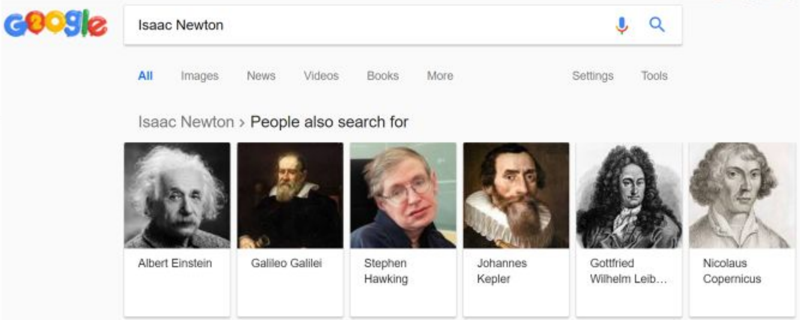
Chapter 1

History of recommendation system, Eliciting Ratings and other Feedback Contributions, Implicit and explicit Ratings, Recommender system functions. Linear Algebra notation: Matrix addition, Multiplication, transposition, and inverses; covariance matrices, Understanding ratings, Applications of recommendation systems, Issues with recommender system.

Recommender Systems

- A **recommender system** (RS) helps users that have no sufficient competence (capability) or time to evaluate the, potentially overwhelming, number of alternatives offered by a web site.
  - In their simplest form, RSs recommend to their users personalized and **ranked lists of items**

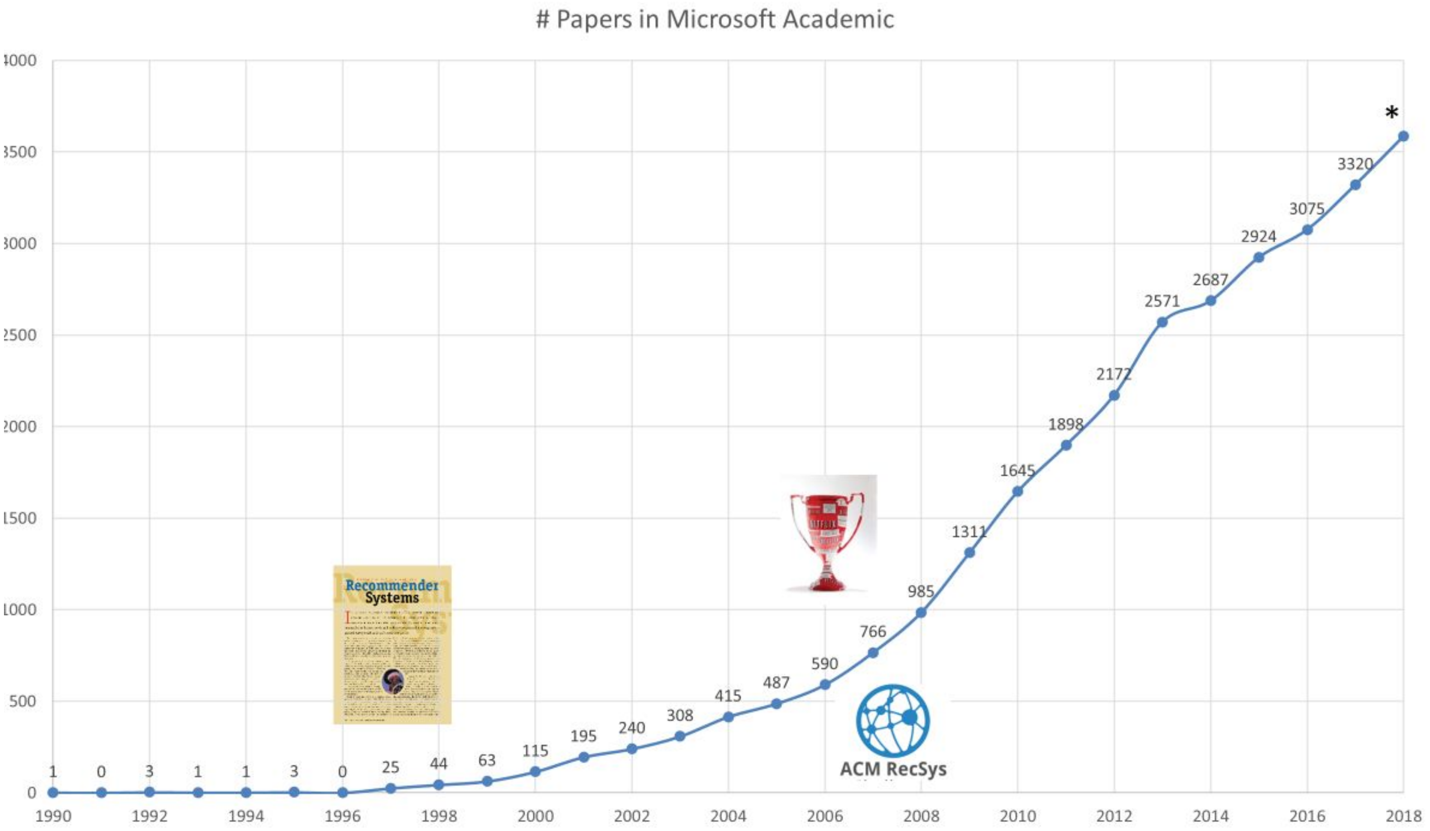


The Impact of RecSys

- 35% of the purchases on Amazon are the result of their recommender system, according to **McKinsey**.
- During the Chinese global shopping festival of November 11, 2016, Alibaba achieved growth of up to 20% of their conversion rate using personalized landing pages, according to **Alizila**.
- Recommendations are responsible for 70% of the time people spend watching videos on **YouTube**.
- 75% of what people are watching on Netflix comes from recommendations, according to **McKinsey**

https://tryolabs.com/blog/introduction-to-recommender-systems/

The Rise of the Recommender System



Estimated-2018*

History of recommendation system

Early Systems (Pre-1990s):

The concept of recommendation systems can be traced back to early attempts to provide personalized recommendations in libraries and retail settings. Traditional methods involved human experts curating lists or suggesting items based on customers' preferences.

People have been trying to give personalized suggestions in libraries and stores for a long time. In the past, this was done by experts making lists or recommending things based on what customers like.

Collaborative Filtering (1990s):

The 1990s saw the rise of collaborative filtering, a technique where users are recommended items based on the preferences of users with similar tastes. GroupLens, a research project at the University of Minnesota in 1992, is often credited with pioneering collaborative filtering for movie recommendations.

History of recommendation system

Content-based filtering is a way to recommend items to users by focusing on the characteristics or features of the items they have shown interest in before. It suggests new items that are similar to ones the user has liked in the past. Imagine you frequently watch action movies on a streaming platform. Using content-based filtering, the system would recommend more action movies to you because it identifies the similarity between the content features (genre, actors, directors) of the movies you've enjoyed in the past.

Content-Based Filtering (1990s):

Content-based filtering emerged as an alternative approach, recommending items based on the user's historical preferences and the characteristics of items.

Pandora, a music recommendation service launched in 2000, used content-based methods to suggest songs based on musical features.

Netflix Prize (2006):

The Netflix Prize competition in 2006 marked a significant milestone, offering a substantial reward for improving Netflix's movie recommendation algorithm by 10%.

This competition spurred research and innovation in recommendation algorithms.

History of recommendation system

Introduction of Hybrid Models (2000s):

Hybrid recommendation systems, combining collaborative filtering and content-based filtering, gained popularity for their ability to provide more accurate and diverse suggestions.

Companies like Amazon and Netflix started incorporating hybrid approaches into their recommendation engines.

Matrix factorization became important for recommendation systems. It was a technique that helped these systems work well with sparse data (where not all user-item interactions are known) and efficiently handle large datasets.
This method involved breaking down a large matrix of user-item interactions into smaller matrices, making it easier to understand and work with the data.

Matrix Factorization and Deep Learning (2010s):

Matrix factorization techniques gained prominence, enabling recommendation systems to deal with sparse data and handle large datasets more efficiently.

The advent of deep learning in the 2010s brought about improvements in recommendation accuracy, with neural networks being applied to model complex patterns in user behavior.

Deep learning, a subset of machine learning, became prominent in the 2010s and brought improvements to recommendation systems.
Neural networks, a key component of deep learning, were applied to understand intricate patterns in user behavior. This led to more accurate recommendations as the systems could learn and adapt to complex user preferences.

In summary, Matrix Factorization improved the efficiency of handling large datasets, while Deep Learning, especially through neural networks, enhanced the accuracy of recommendations by capturing intricate user behavior patterns.

History of recommendation system

Recommendation systems began taking into account contextual information, including factors like location, time, and device, to offer more personalized and relevant suggestions to users.
This approach recognized that user preferences and needs can vary based on the specific context in which they are interacting with the system.

Context-Aware Recommendations (2010s):

Mobile apps and e-commerce platforms, in particular, started utilizing context-aware recommendations extensively to enhance user experience by tailoring suggestions to the user's current situation or environment.

Recommendation systems started considering contextual information, such as location, time, and device, to provide more personalized and relevant suggestions.

Mobile apps and e-commerce platforms increasingly leveraged context-aware recommendations.

In essence, Context-Aware Recommendations in the 2010s aimed to make suggestions more adaptable to the user's immediate context, acknowledging that recommendations could be more effective when they consider factors beyond historical user interactions.

Current Trends (2020s):

Providing clear explanations for why a particular recommendation is made helps build user trust by making the system's decision-making process more transparent.

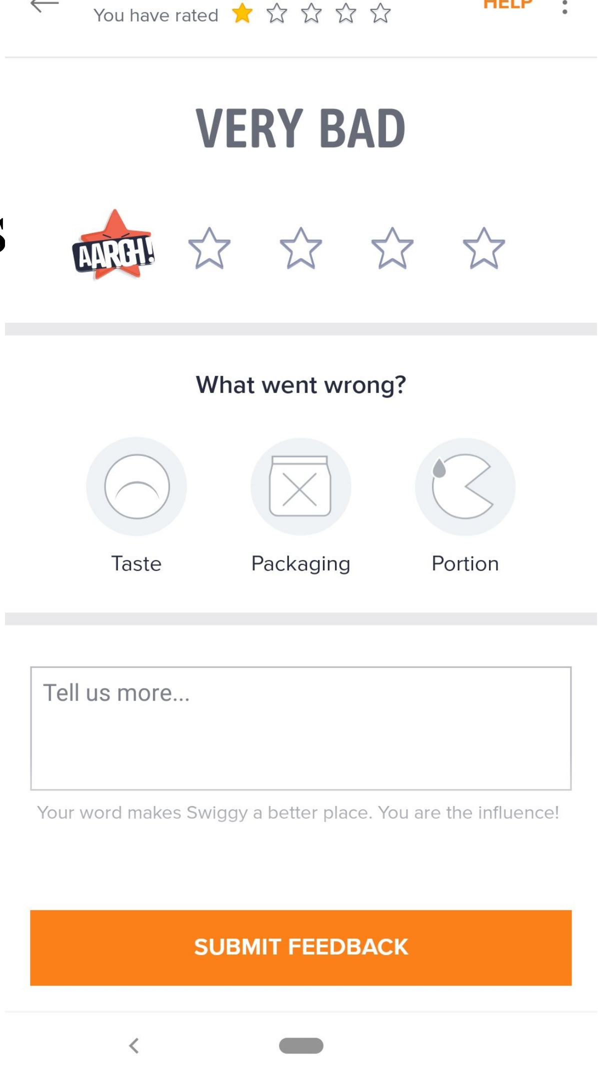
Recent advancements include the use of reinforcement learning for recommendation systems and the integration of explainability features to enhance user trust.

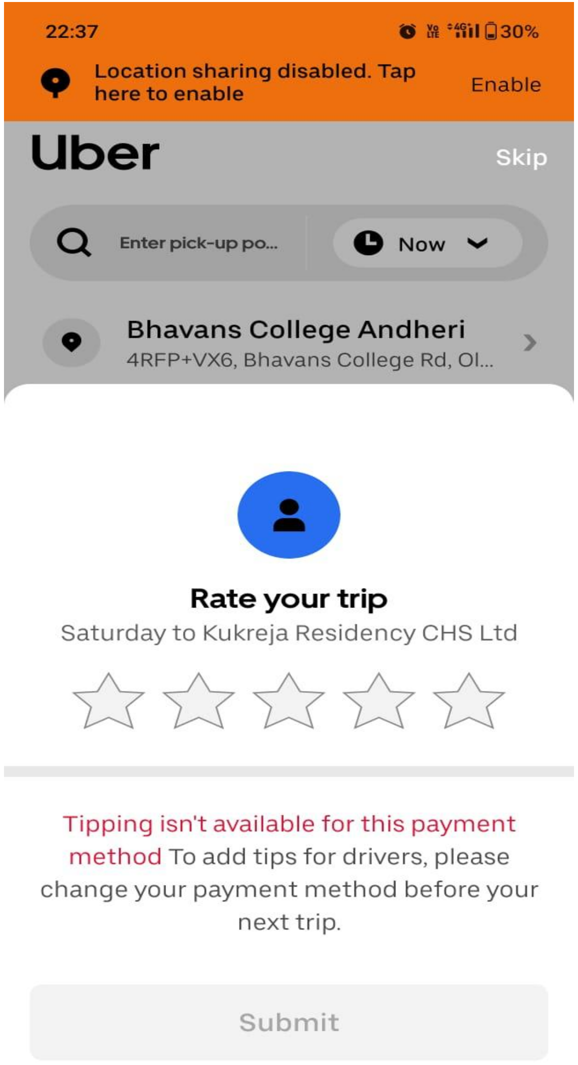
Many recommendation systems now leverage large-scale data, machine learning, and AI to continually improve accuracy and adapt to changing user preferences.

The use of sophisticated algorithms and AI techniques allows recommendation systems to adapt to changing user preferences, providing more dynamic and personalized suggestions.

the act of getting information

Eliciting Ratings and other feedbacks





Explicit Ratings:

- Star Ratings: Users are asked to provide ratings on a scale (e.g., 1 to 5 stars) for items they have experienced. This is a straightforward way to collect explicit feedback.

User Reviews:

- Encouraging users to write reviews provides more detailed insights into why they liked or disliked a particular item. Natural language feedback can be valuable for understanding nuances.

Thumbs Up/Down:

- A binary feedback mechanism, where users indicate whether they liked (thumbs up) or disliked (thumbs down) a recommendation. It's a quick way to gather feedback without the need for a numerical rating.

Surveys and Questionnaires:

- Periodically sending surveys or questionnaires to users can provide structured feedback. Questions may cover aspects like satisfaction, preferences, and suggestions for improvement.

Badges:
Example Badge: "Cinema Buff"
Criteria: Awarded to users who have rated 50 or more movies.
Rewards:
Example Reward: "VIP Access"
Criteria: Users who consistently provide feedback by rating or reviewing movies are granted VIP access, unlocking exclusive content or early access to new releases.
Challenges:
Example Challenge: "Weekend Watcher"
Objective: Watch three movies over the weekend.
Reward: Users who complete the challenge receive a special weekend watcher badge and a discount on their next movie rental.

Implicit Feedback:

- Click-Through Rate (CTR): Analyzing user clicks on recommended items provides implicit feedback. Higher click-through rates suggest that users find the recommendations relevant.
- Dwell Time: Monitoring how much time users spend on recommended items can indicate interest and satisfaction.

Preference Learning:

For example, in a streaming service, a preference learning system might allow users to like or dislike movies, genres, or actors. As the user interacts more, the system learns from these preferences and refines its recommendations to better align with the user's taste.

- Interactive interfaces that allow users to actively express preferences and customize recommendations. For example, users might be able to like, dislike, or customize their preferences.

Gamification:

- Incorporating gamified elements such as badges, rewards, or challenges can motivate users to engage with the recommendation system and provide feedback.

In-App Feedback:

For example, a mobile shopping app might include a simple star rating system or a comment box after a user completes a purchase. This allows users to share their experience or provide feedback on specific products without leaving the app.

- Integrating feedback mechanisms directly within the application or platform makes it convenient for users to share their opinions without leaving the environment.

Social Media Integration:

- Allowing users to share their favorite items or recommendations on social media platforms not only promotes the system but also serves as indirect positive feedback

In this scenario, the gamified elements are integrated into the movie recommendation app to enhance user engagement:
Users earn badges as a visible representation of their achievements.
Rewards, such as VIP access, provide tangible benefits for active participation.
Challenges introduce short-term goals to encourage regular interaction with the app.
By incorporating these gamified elements, the movie recommendation app aims to make the user experience more enjoyable and motivate users to explore and contribute to the platform

.

A/B Testing:

- Experimenting with different recommendation algorithms and interfaces in an A/B testing framework allows for the comparison of user engagement and feedback between different versions.

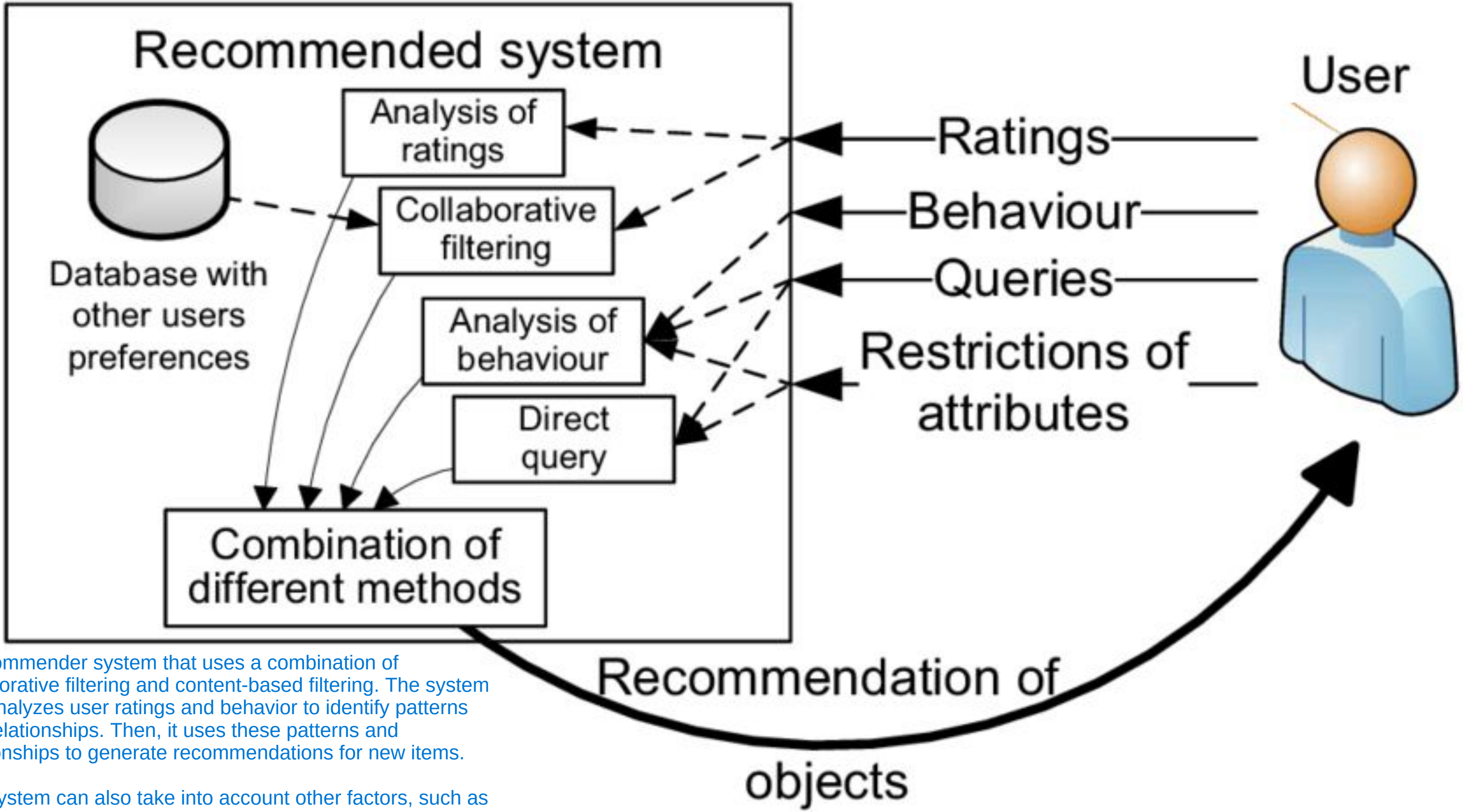
Contextual Feedback:

- Gathering feedback in specific contexts, such as after a purchase or when a user completes a particular action, can provide more targeted insights.

Continuous Monitoring:

- Regularly monitoring user interactions and adapting the system based on real-time feedback ensures that the recommendations stay relevant over time.

Implicit and explicit Ratings



a recommender system that uses a combination of collaborative filtering and content-based filtering. The system first analyzes user ratings and behavior to identify patterns and relationships. Then, it uses these patterns and relationships to generate recommendations for new items.

The system can also take into account other factors, such as the user's queries and restrictions of attributes. These additional factors can help to improve the accuracy of the recommendations.

Explicit Ratings:

- Definition: Explicit ratings are direct and clear expressions of a user's preference for an item. Users consciously assign a score or provide feedback, typically on a numerical scale (e.g., 1 to 5 stars).
- Examples: Star ratings on a movie (e.g., a 4-star rating for a film), a numerical score for a product review (e.g., rating a book as 8 out of 10), or a thumbs-up/thumbs-down indication.
- Pros:
    - Clear and direct indication of user preference.
    - Quantifiable and easy to interpret. able to be expressed as an amount, quantity, or numerical value
    - Useful for collaborative filtering and matrix factorization.
- Cons:
    - Users may not always provide ratings for all items.

Implicit Ratings:

- Definition: Implicit ratings are derived from user actions and behaviors, indicating preferences indirectly based on interactions with items. These actions may include clicks, views, purchases, or time spent on an item.
- Examples: Click-through rates (CTR) on recommended items, the number of times a song is played, the frequency of purchasing certain products.
- Pros:
    - Captures user preferences without requiring explicit input.
    - Less effort from users, as preferences are inferred from behavior.
    - Often more abundant and representative of user interactions.
- Cons:
    - Interpretation may be less clear compared to explicit ratings.
    - Implicit feedback may not always reflect user satisfaction accurately.

Movie: The title of the movie.
Ordered Rating: The average rating of the movie on a scale of 1 to 5.
Unary Rating: The average rating of the users who have seen the movie, where 1 indicates that the user liked the movie and 0 indicates that the user did not like the movie.

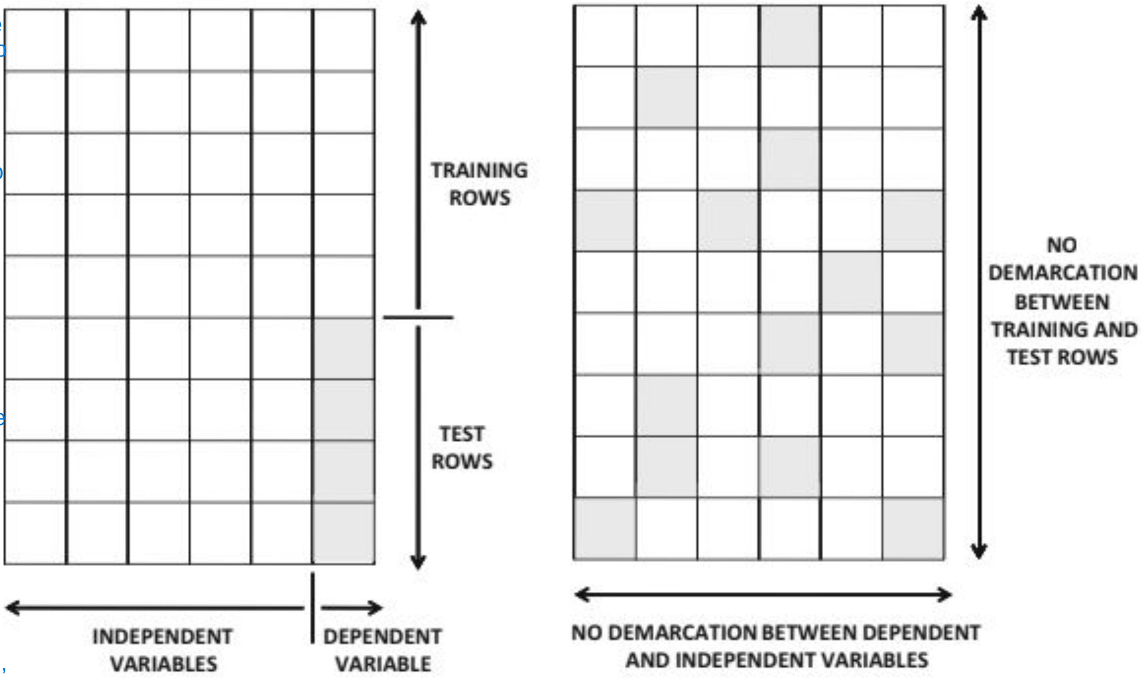
	GLADITOR	GODFTHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ordered ratings

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

(b) Unary ratings

Training and Test Rows: These represent the data that is used to train and test the recommender system models. The training rows are used to learn the patterns in the data, while the test rows are used to evaluate how well the models perform on unseen data.
Independent Variables and Dependent Variable: These are the features of the data that are used to make predictions. In the traditional classification problem, the independent variables are the features of the items (e.g., movie genre, director), and the dependent variable is the user's rating of the item. In the collaborative filtering problem, the independent variables are the features of the users (e.g., age, gender) and the features of the items, and the dependent variable is the rating of the item by other users.
Demarcation Between Training and Test Rows: This indicates whether there is a clear separation between the training and test data. In the traditional classification problem, there is a clear demarcation, while in the collaborative filtering problem, there is not. This is because collaborative filtering algorithms typically use all of the available data to make predictions.



(a) Classification (b) Collaborative filtering

Figure 1.4: Comparing the traditional classification problem with collaborative filtering. Shaded entries are missing and need to be predicted.

The diagram also highlights some of the key differences between the two models:

Traditional classification: This model assumes that there is a clear relationship between the features of the items and the user's rating of the item. This means that the model can be trained on a set of items that the user has rated, and then used to predict the user's rating of new items.
Collaborative filtering: This model does not assume that there is a clear relationship between the features of the items and the user's rating of the item. Instead, it assumes that users with similar tastes will tend to rate items similarly. This means that the model can be trained on the ratings of all users, and then used to predict the rating of a new item for a specific user based on the ratings of other users who have similar tastes.

Recommender system functions

Collaborative filtering

Content based filtering

Matrix factorization

Deep learning approaches

Hybrid recommender systems

Context aware recommender system

Reinforcement Learning

Collaborative Filtering:
User ratings: This refers to the ratings (e.g., stars, likes/dislikes) that a specific user has provided for certain items (e.g., movies, products). These ratings serve as explicit indications of the user's preferences.
Community ratings: This encompasses the ratings provided by other users in the system. By analyzing similarities between the target user's ratings and those of other users, the system identifies users with similar preferences and leverages their ratings to recommend items the target user might also enjoy.

Content-Based:
User ratings: Similar to collaborative filtering, user ratings provide information about the user's preferences.
Item attributes: This refers to the features and characteristics of the items themselves. It could include genre information for movies, artist and style for music, or brand and material for products. By analyzing the user's past ratings and matching them with item attributes, the system recommends items with similar characteristics.

Knowledge-Based:
User specification: This goes beyond ratings and enters the realm of explicit user requests. Imagine a user searching for "adventure movies with strong female leads released in 2024." This search query becomes the user specification, explicitly outlining their preferences.
Item attributes: Similar to content-based filtering, item attributes are crucial for matching user specifications. Here, they help identify movies that possess the desired characteristics (adventure genre, strong female lead, 2024 release).
Domain knowledge: This refers to additional information the system possesses about the items and the domain they belong to. For example, knowledge about movie directors, awards, or release dates can further refine recommendations based on user specifications.

Table 1.2: The conceptual goals of various recommender systems

Approach	Conceptual Goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings + community ratings
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions.	User ratings + item attributes
Knowledge-based	Give me recommendations based on my explicit specification of the kind of content (attributes) I want.	User specification + item attributes + domain knowledge

Collaborative Filtering: This approach leverages the ratings and actions of other users to suggest items to a particular user. For instance, if user A and user B share similar taste in movies and user B highly rates movie C, then a collaborative filtering system might recommend movie C to user A as well.

Content-Based: This approach focuses on the characteristics of the items themselves. If a user frequently listens to rock music, the system might recommend other rock music albums based on the genre or similar attributes.

Knowledge-Based: This approach takes into account the user's specific requests and domain knowledge to suggest items. For example, if a user explicitly searches for "romantic comedies released in 2023", the system would use its knowledge of movie genres and release dates to recommend relevant titles.

Applications of recommendation system

1. E-Commerce Is an industry where recommendation systems were first widely used. With millions of customers and data on their online behavior, e-commerce companies are best suited to generate accurate recommendations.

2. Retail Target scared shoppers back in the 2000s when Target systems were able to predict pregnancies even before mothers realized their own pregnancies. Shopping data is the most valuable data as it is the most direct data point on a customer's intent. Retailers with troves of shopping data are at the forefront of companies making accurate recommendations.

3. Media Similar to e-commerce, media businesses are one of the first to jump into recommendations. It is difficult to see a news site without a recommendation system.

4. BankingA mass-market product that is consumed digitally by millions. Banking for the masses and SMEs are prime for recommendations. Knowing a customer's detailed financial situation, along with their past preferences, coupled with data of thousands of similar users, is quite powerful.

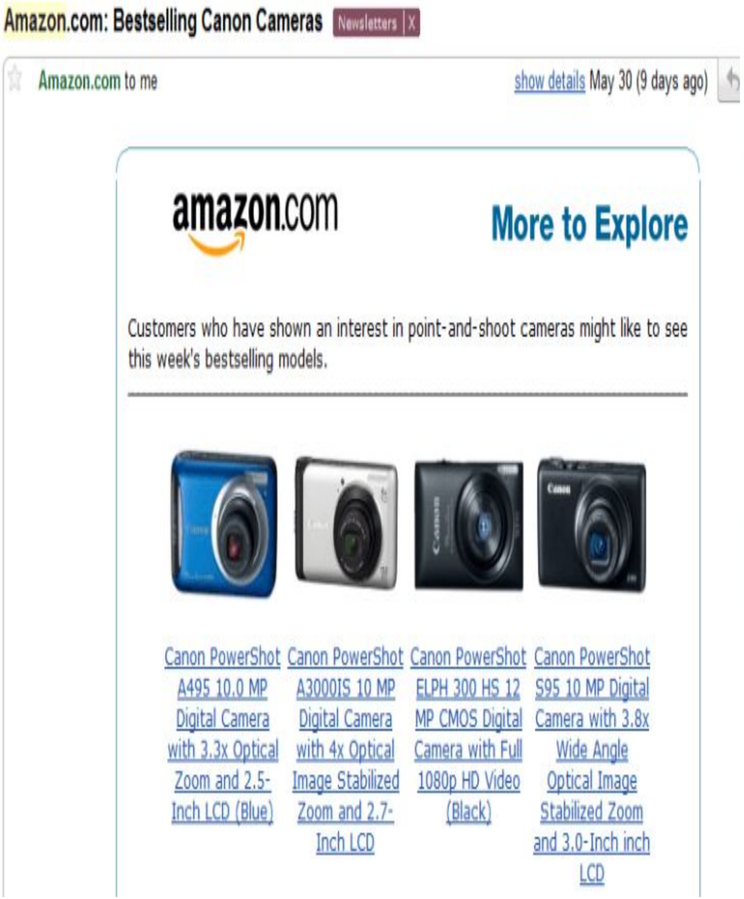
5. TelecomIt Shares similar dynamics with banking. Telcos have access to millions of customers whose every interaction is recorded. Their product range is also rather limited compared to other industries, making recommendations in telecom an easier problem.

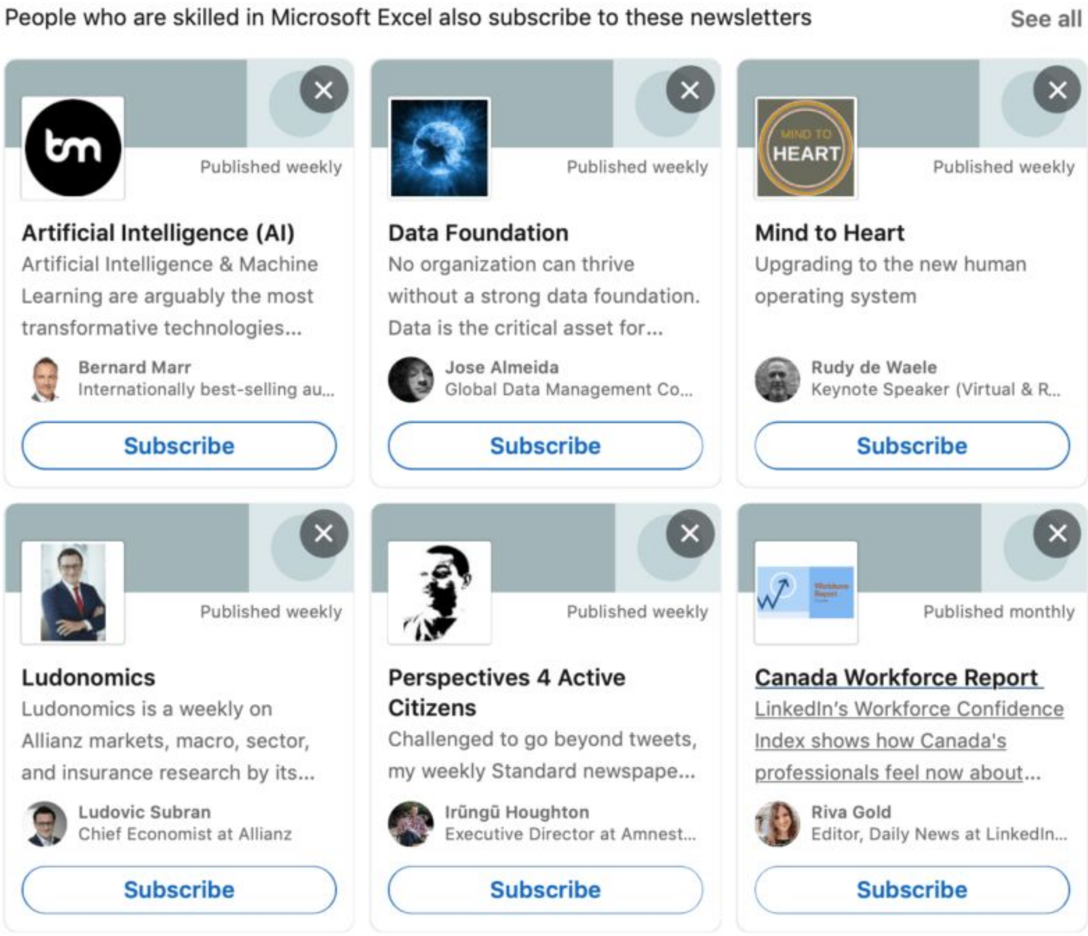
In the telecom industry, recommendation systems share similarities with banking due to access to vast customer data and the relatively limited product range. Telecom companies, with millions of customers and detailed records of every interaction, find recommendations more straightforward compared to industries with broader product offerings. The focused nature of telecom services allows recommendation systems to efficiently suggest relevant plans, features, or upgrades based on individual customer behaviors and preferences. This targeted approach enhances customer satisfaction and the overall telecom experience.

6. Utilities
Similar dynamics with telecom, but utilities have an even narrower range of products, making recommendations rather simple.

In the context of banking, leveraging recommendation systems proves highly effective, particularly for mass-market products used digitally by millions, such as banking services for individuals and small to medium-sized enterprises (SMEs). By understanding a customer's comprehensive financial situation and past preferences, and combining this information with data from thousands of similar users, recommendation systems gain significant power. This enables banks to offer tailored financial products and services, optimizing customer experiences, and providing relevant suggestions based on individual needs and broader market trends.

Figure 1.5: A hypothetical example of an initial user interface for a constraint-based recommender)


Amazon.com: Bestselling Canon Cameras
Newsletters | X
Amazon.com to me
show details May 30 (9 days ago)
amazon.com
More to Explore
Customers who have shown an interest in point-and-shoot cameras might like to see this week's bestselling models.
Canon PowerShot A495 10.0 MP Digital Camera with 3.3x Optical Zoom and 2.5-Inch LCD (Blue)
Canon PowerShot A3000IS 10 MP Digital Camera with 4x Optical Image Stabilized Zoom and 2.7-Inch LCD
Canon PowerShot ELPH 300 HS 12 MP CMOS Digital Camera with Full 1080p HD Video (Black)
Canon PowerShot S95 10 MP Digital Camera with 3.8x Wide Angle Optical Image Stabilized Zoom and 3.0-Inch inch LCD


People who are skilled in Microsoft Excel also subscribe to these newsletters
See all
Published weekly
Artificial Intelligence (AI)
Artificial Intelligence & Machine Learning are arguably the most transformative technologies...
Bernard Marr
Internationally best-selling au...
Subscribe
Published weekly
Data Foundation
No organization can thrive without a strong data foundation. Data is the critical asset for...
Jose Almeida
Global Data Management Co...
Subscribe
MIND TO HEART
Published weekly
Mind to Heart
Upgrading to the new human operating system
Rudy de Waele
Keynote Speaker (Virtual & R...
Subscribe
Published weekly
Ludonomics
Ludonomics is a weekly on Allianz markets, macro, sector, and insurance research by its...
Ludovic Subran
Chief Economist at Allianz
Subscribe
Published weekly
Perspectives 4 Active Citizens
Challenged to go beyond tweets, my weekly Standard newspape...
Irũngũ Houghton
Executive Director at Amnest...
Subscribe
Published monthly
Canada Workforce Report
LinkedIn's Workforce Confidence Index shows how Canada's professionals feel now about...
Riva Gold
Editor, Daily News at LinkedIn...
Subscribe

Problems with Recommendation System

- Data Sparsity. Data sparsity is a common issue in recommendation systems where there are many users and items, but only a small number of them interact with each other. ...
- Cold Start Problem. ...
- Scalability. ...
- Overfitting. ...
- Diversity. ...
- Privacy

Problems with Recommendation System

Data sparsity is a common challenge in recommendation systems that arises when there are many users and items, but only a limited number of interactions or transactions between them. In other words, the available data about user preferences or item usage is sparse, with a large portion of the user-item matrix being empty or lacking sufficient information.

- Data Sparsity. Data sparsity is a common issue in recommendation systems where there are many users and items, but only a small number of them interact with each other. ...
- Cold Start Problem. ...
- Scalability. ...
- Overfitting. ...
- Diversity. ...
- Privacy

The "Cold Start Problem" is a challenge in recommendation systems that occurs when the system struggles to provide accurate or meaningful suggestions for new items or users who have limited or no interaction history. This problem can be categorized into two main types: the new user cold start and the new item cold start.

New User Cold Start:
Occurs when a new user joins the system and has not yet provided any preferences or interactions.

New Item Cold Start:
Occurs when a new item is introduced to the system, and there is limited or no historical user interaction data for that item.

Scalability in the context of recommendation systems refers to the system's ability to handle an increasing amount of data, users, or requests while maintaining or improving its performance. As a recommendation system scales, it should be able to efficiently process and deliver recommendations without significant degradation in speed or quality.

In recommendation systems, overfitting can result in models that provide accurate predictions on the training data but struggle to generalize to diverse user preferences, leading to less effective and less robust recommendations in real-world scenarios.

Diversity in recommendation systems refers to providing users with a varied and balanced set of recommendations, encompassing different genres, novelty, and unexpected yet relevant items. It aims to enhance user experience by avoiding monotony, encouraging exploration, and catering to diverse user preferences. Achieving diversity involves balancing popular and niche recommendations, adapting to changing user preferences, and ensuring ethical considerations in the recommendation process.

Privacy in the context of recommendation systems refers to the protection of users' personal information and preferences while still providing effective and personalized recommendations. It involves ensuring that user data is handled responsibly, respecting individual privacy rights, and preventing unauthorized access or misuse of sensitive information.

Matrix operations

Matrix addition,

Multiplication,

Transposition, and

Inverses https://byjus.com/maths/inverse-matrix/#:~:text=We%20can%20find%20the%20matrix,determinant%20of%20the%20given%20matrix.

What Is Covariance?

Covariance measures the directional relationship between the returns on two assets. A positive covariance means asset returns move together, while a negative covariance means they move inversely.

Covariance is calculated by analyzing at-return surprises (standard deviations from the expected return) or multiplying the correlation between the two random variables by the standard deviation of each variable.

+ive
-ive

Covariance is a statistical measure that quantifies the degree to which the returns of two assets move in relation to each other. It indicates the directional relationship between the returns of the assets. Here's a breakdown of the key points mentioned:

Definition:

Covariance measures the joint variability of returns between two assets.
Positive Covariance:

A positive covariance indicates that the returns of the two assets tend to move together.
When one asset has a positive return surprise, the other asset is likely to also experience a positive return surprise.
Negative Covariance:

A negative covariance suggests that the returns of the two assets move in opposite directions.
If one asset has a positive return surprise, the other asset is likely to have a negative return surprise.
Calculation:

Covariance can be calculated by analyzing return surprises, which are the standard deviations from the expected return.
Alternatively, it can be calculated by multiplying the correlation coefficient between the two assets by the standard deviation of each asset.
Correlation and Standard Deviation:

Covariance is related to correlation, which measures the strength and direction of a linear relationship between two variables.
Multiplying the correlation by the standard deviation of each variable gives the covariance.
In summary, covariance provides insights into how the returns of two assets co-move. A positive covariance suggests a tendency to move together, while a negative covariance indicates an inverse relationship. The calculation involves analyzing return surprises or using the correlation coefficient and standard deviations of the assets.