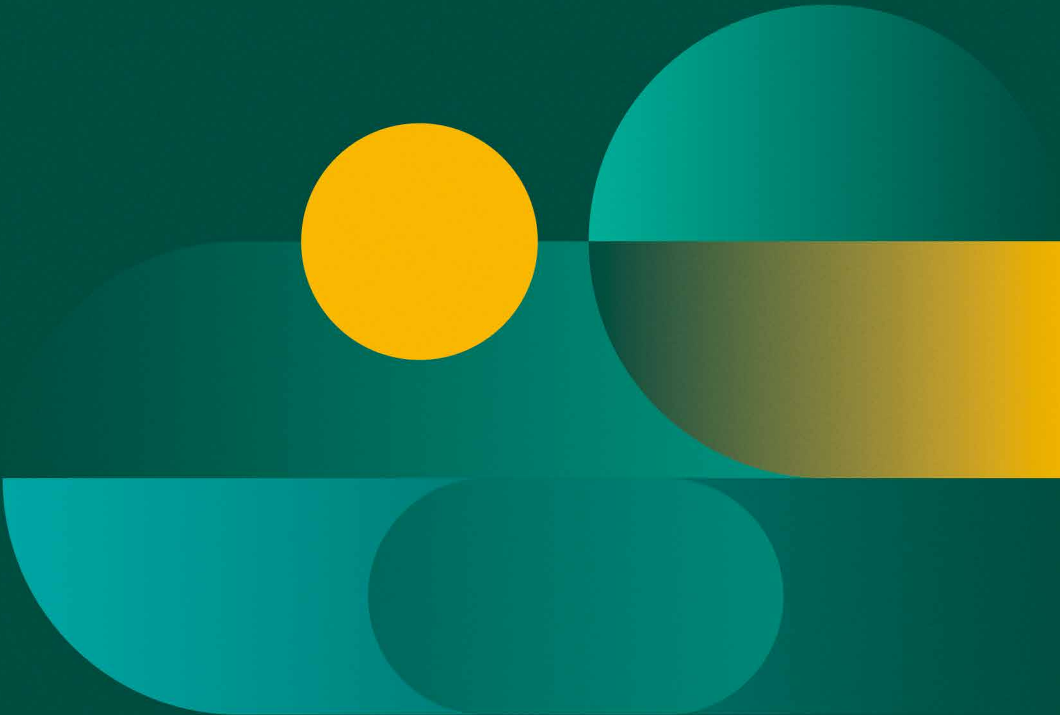
Recombee



Modern Recommender Systems – Part 1

Perspectives, Data, and Objectives: Making Recommender Systems Aligned With Users, Business, Product, and Content Goals

Pavel Kordik

Published by Recombee

What's Inside

Preface

The rapid evolution of digital platforms has made recommender systems a cornerstone of user experience and business growth. Over the past decade, Recombee has partnered with hundreds of organizations across diverse industries, witnessing firsthand the opportunities and challenges that come with deploying recommendation technology at scale. This booklet distills our collective experience into practical guidance for domain experts, product leaders, and technical decision-makers.

Our motivation is simple: to demystify the process of building and operating effective recommender systems, bridging the gap between technical complexity and strategic value. Whether you are evaluating your first deployment or seeking to optimize an existing solution, we hope this resource will help you avoid common pitfalls, adopt best practices, and unlock the full potential of personalized recommendations for your users and your business.

Modern Recommender Systems – Chapter 1: Introduction

Modern Recommender Systems – Chapter 1: Introduction

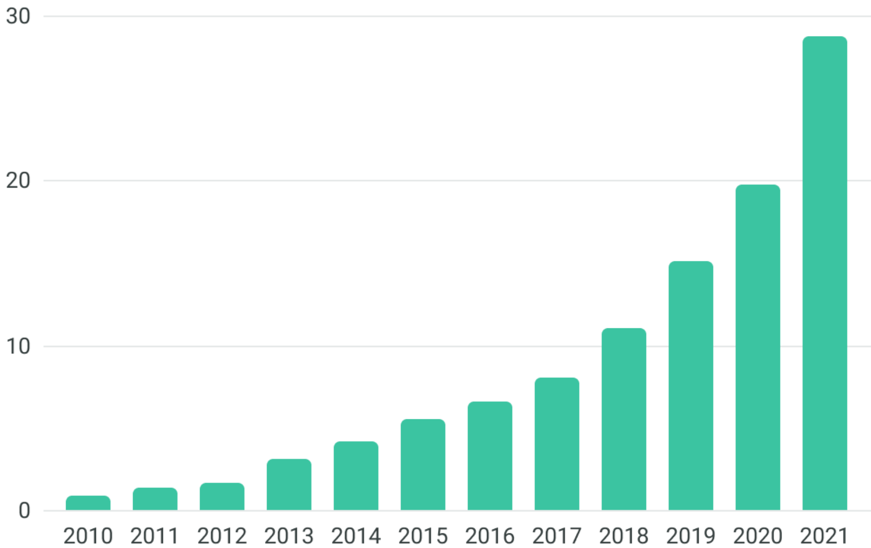
Over the last ten years, we at Recombee have been building a universal and domain-agnostic recommender system, serving hundreds of customers including major brands across diverse industries. This experience has taught us valuable lessons, which we share here to provide a comprehensive overview of the technology behind modern recommenders that power almost every major site you use for online discovery and search. We will also discuss related challenges and ethical aspects of this technology.

In this initial chapter, we explore the emergence of recommender systems and clarify how they differ from advertising technology, which is often mistakenly seen as the same. Our goal is to highlight the various objectives that recommender systems fulfill for different stakeholders. In the following chapters, we will delve into the crucial data and signals essential for modern recommenders, followed by an in-depth exploration of the technology behind them.

Recommender systems have become pervasive and are arguably the most influential machine learning technology today, as everyone receives hundreds of recommendations daily, whether it‘s news to read, songs to listen to, movies to watch, items to purchase, or social media content to see.

The number of recommendations an average active online user receives has been growing steadily over the years, accelerating exponentially in the last decade, with no sign of saturation. This trend is driven by:

- The increasing number of internet users
- More time spent online per person
- The widespread adoption of recommender systems by websites and online services



For example, YouTube's revenue growth is closely tied to the number of recommendations served and active usage. We observe similar trends among Recombee's customers, where effective recommendations directly impact engagement and business outcomes.

Recommender systems have flourished and improved, but let's first look at how it all started.

History

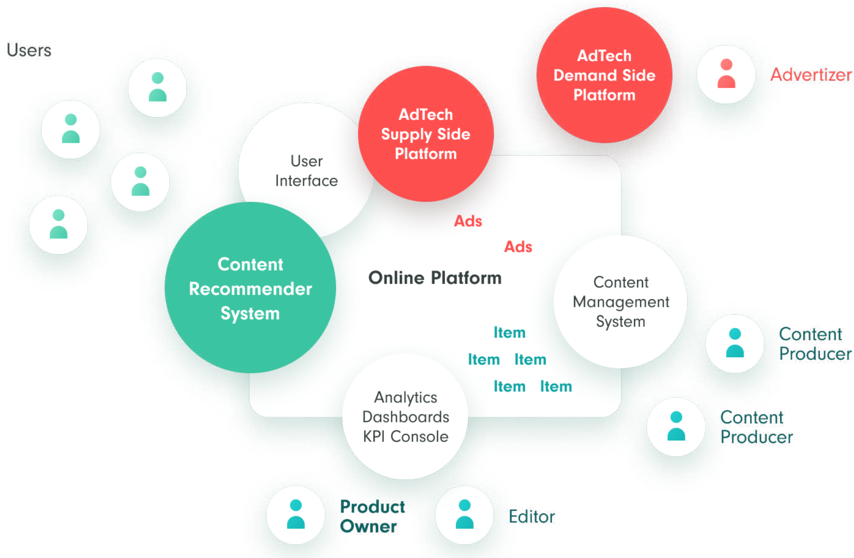
Recommender systems emerged alongside information retrieval (IR) systems in the early seventies, enabled by the availability of computers and the internet.

Traditional IR systems focused on assisting users in searching large catalogs using text queries. These systems produced the same, non-personalized output for everyone. Even these early systems were gradually improved by sequential learning (Information retrieval: A sequential learning process, 1983) and learning to rank algorithms (Learning to rank using gradient descent, Burges, 2005). We describe these machine learning algorithms for IR systems in our blogpost on personalized search.

The advent of personal computers and widespread internet access enabled personalized recommendations based on user actions. One of the first systems relying exclusively on user historical interactions (explicit ratings) was GroupLens (1992).

Since then, recommender systems have developed in many directions. Modern personalized recommendation and search systems combine multiple techniques to optimize various objectives. In practice, the most successful deployments are those that align technology with business strategy, data realities, and user needs; not just those with the most sophisticated models.

Do Recommenders Really Spy on Users and Generate Targeted Ads? Many people associate recommender systems with annoying targeted advertisements. However, such ads are often based on simple heuristics and do not use AI-based recommenders at all.



A typical example is abandoned cart retargeting, where e-commerce sites display ads for products left in a user's cart as they browse the web. These are usually simple reminders, not machine learning-driven recommendations.

Similarly, ads on media websites are typically auctioned by AdTech platforms based on context and user profiles, not by recommender systems. AdTech relies on collecting and analyzing large amounts of data about a user's online activities to create detailed profiles, raising ethical concerns about privacy and data protection.

In contrast, recommender systems are primarily used to help users find relevant content within a website or app. They typically only consider anonymized user interactions with on-site content and do not utilize external data from other websites or user attributes. As a result, there are generally fewer ethical concerns compared to AdTech systems, which may use a wider range of data to personalize ads.

Goals

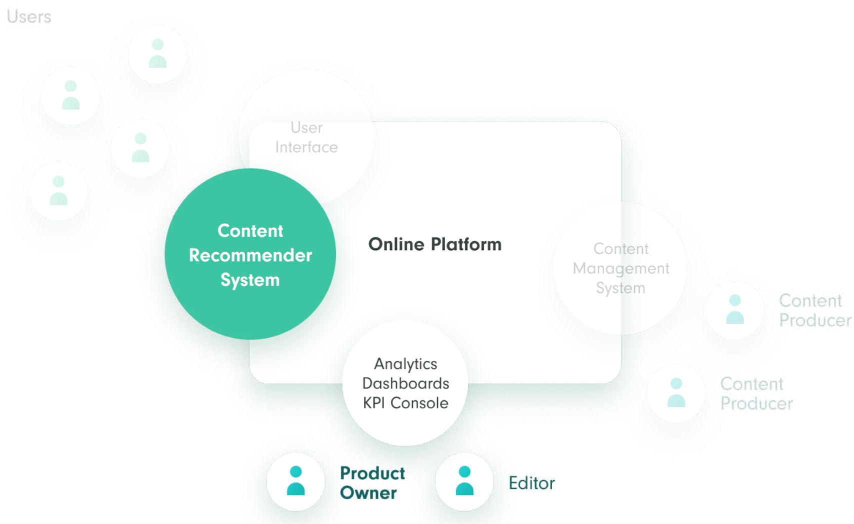
Classical IR methods aim to make search faster and more accurate, assuming users know what they want and can formulate a query. Recommender systems, on the other hand, focus on inspiring users or helping them discover new items they may not even know about. Modern recommenders incorporate both search and discovery, allowing users to start typing a query and receive intelligent suggestions. The ambition is to improve user experience and optimize engagement.

While users are the main beneficiaries, the objectives of a recommender system are defined by its owners, designers, and developers. Recommenders are often set up to optimize for certain metrics, such as click-through or conversion rate. However, optimizing only for simple metrics can lead to unintended consequences, such as promoting clickbait or optimizing for short-term gains at the expense of long-term value. It is important to balance business, user, and content creator objectives for sustainable success.

Product Owner Perspective

A great product grows its user base and keeps users engaged, but product owners also focus on increasing revenue. Subscription-based services optimize for loyal subscribers, aligning with user experience.

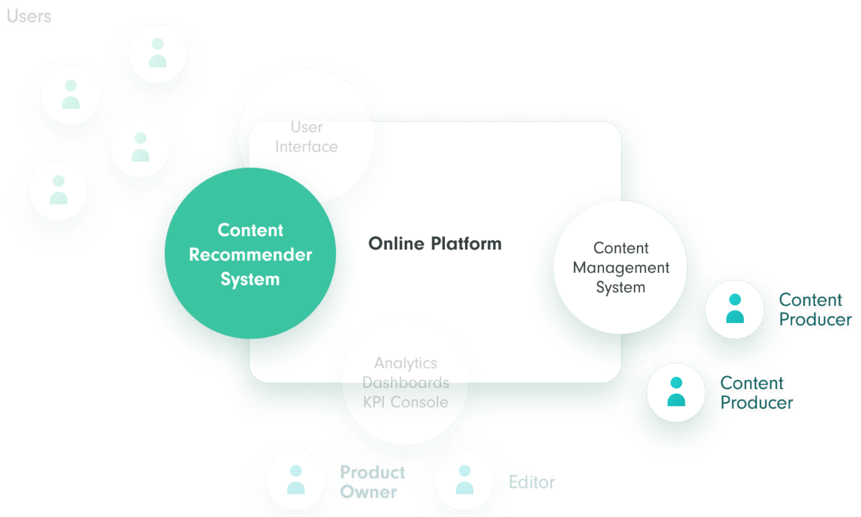
Ad-supported businesses may maximize page views, but too much emphasis on short-term revenue can reduce user loyalty and long-term value. Many media organizations now use hybrid models, motivating conversion of „freemium“ users into subscribers by recommending relevant premium content.



For online aggregators or retailers, the goal may be to recommend third-party content or high-margin products. However, over-optimizing for these objectives can decrease user engagement. Manual curation uses similar „tricks,“ but recommenders allow for more systematic and efficient targeting, such as boosting high-margin content only for relevant users.

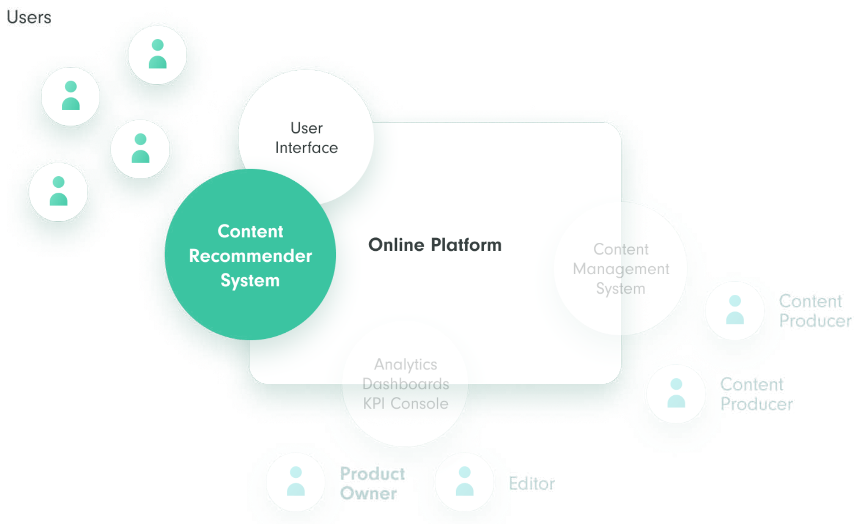
Content Producer Perspective

Content creators (artists, writers, vendors) aim to reach an audience that will engage with and appreciate their work. They rely on recommenders to distribute content to the right users. Creators want their content to be seen by those likely to respond positively, and may tailor their work to maximize certain metrics. Recommenders should help new or niche content get discovered, not just promote what is already popular.



User Perspective

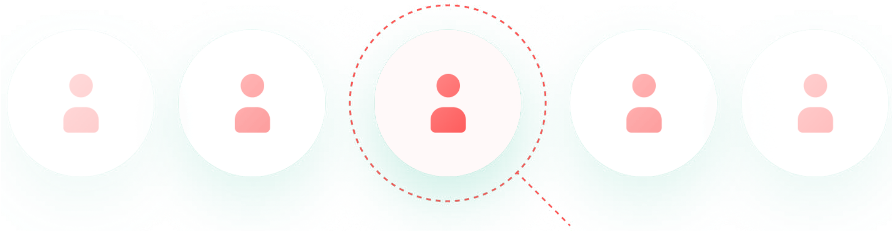
Users expect recommenders to help them reach their goals, which can change even within a single session. For example, a user may start by reading serious news, then switch to lighter content. In e-commerce, a user may browse for inspiration or seek to buy quickly. Good recommenders support these shifting intents, sometimes by offering multiple recommendation scenarios (e.g., „get inspired,“ „your favorites,“ „best value products for you,“ „premium picks“).



The task of the recommender is to predict user intent in real time and support their goals, even when explicit feedback is limited. This is challenging, especially with anonymous users or sparse data, but is essential for aligning recommendations with user needs.

Problems and Ethical Aspects

When stakeholder goals are aligned, deploying a recommender is straightforward. Problems arise when product owners emphasize objectives that conflict with user or content creator goals.



For example, a job board paid per application might optimize for application count, recommending jobs users are likely to apply for regardless of fit. This leads to frustration for users and employers, and degrades the product. Shifting the objective to successful hires aligns interests and improves outcomes, but may require changes to business models and data collection.

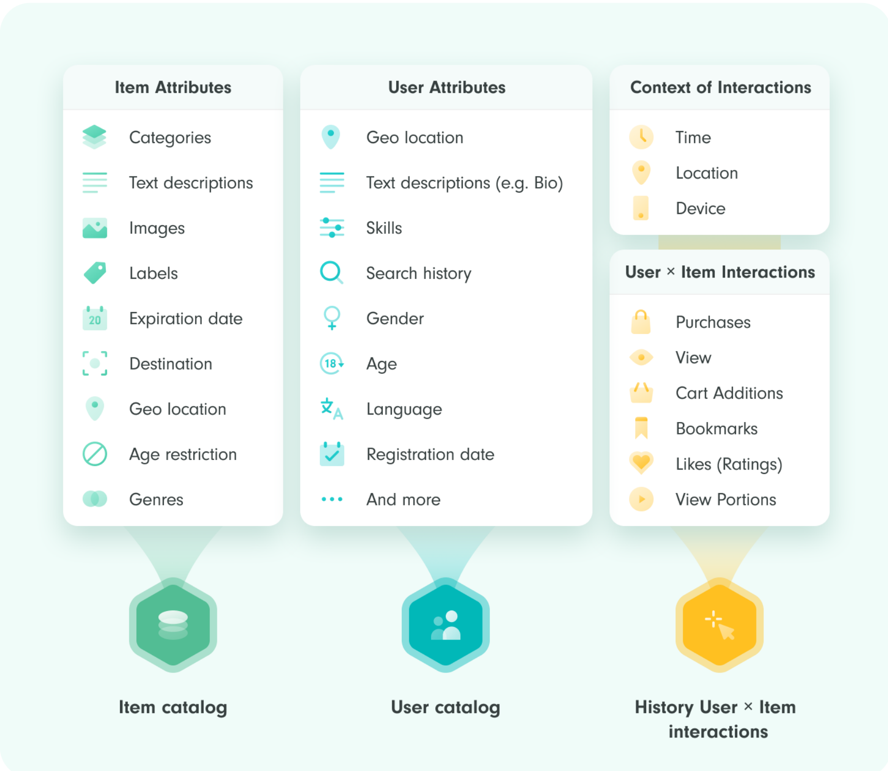
Similarly, optimizing for page views in media can boost short-term ad revenue but harm user experience and content diversity. In our experience, considering long-term criteria, such as user loyalty and subscription conversion, leads to better outcomes. Products that deviate from user or creator interests are likely to lose market share over time.

Modern Recommender Systems – Chapter 2: Data

Modern Recommender Systems – Chapter 2: Data

Data Is Crucial

Data plays an essential role in the functioning of a recommender system, as it is the primary source of information used to generate accurate and personalized recommendations. In this chapter, we discuss the importance of data for recommender systems, the various types of data sources used, and how data can be leveraged to improve the accuracy and effectiveness of recommendations. The data that can be used for recommendations can be categorized into 1) Item catalog, 2) User catalog, and 3) History of user × item interactions.



Attributes of items are stored in the item catalog, user catalog holds information about users, and there are several types of user-to-item interactions that are recorded in different contexts.

Item Catalog

First, it's important to define what types of items can be recommended to users. A database of all items is called an item catalog. In this catalog, we store not only items that can be recommended (active items), but also historical items that were recommended in the past and are not available to users anymore. Those historical items are important when measuring similarity of users who interacted with them in the past.

Attributes of items help recommenders understand how items are related and identify which ones are more alike. Here are a few examples of the most important item attributes (or item properties):

Categories: Items can be categorized into distinct groups, however you might also come with a hierarchical system of categories where one item can belong to multiple categories. Categories can be used to create item segments so you can recommend particular categories to a given user. You can also filter out items from the recommendation based on their category labels or boost probability that items from a particular set of categories are recommended to a user.

Text descriptions: When you recommend articles, the text of the article can be used in a text description attribute of the item. Modern recommenders have capabilities to process text using advanced neural networks. Similarities of text neural item embeddings can be very important especially when recommending cold-start items that do not have many interactions yet.

Images: Modern recommenders can use multiple images of an item to create an image neural item embedding. Again, such information is super important for recommendation systems especially when images play a significant role for users (e.g online art gallery) or when interactions and text descriptions are missing. Imagine an online marketplace where users can upload images of items for sale. As they use their smartphones, it is not likely that they will also add rich and informative text descriptions. Another example would be a real-estate portal, where users like to find similar listings based on images of properties. Or a fashion e-commerce site that decided to utilize visual similarity to recommend alternatives from the product catalog.

User Catalog

Similarly to the item catalog, the user catalog holds attributes and properties of users. Most important user attributes are the following:

Location of user: Geographic location of users is important in recommendation scenarios, when users are interested in items that are located nearby (such as real estate, job or event recommendation). Even users with no interaction history can then get relevant recommendations such as popular items in their region.

User search history: One can suggest relevant items based on historical user search queries. Also, user search history is instrumental for personalized query suggestions, where reminding users about their past similar queries is very helpful.

User bio, interests or skills: In some domains, it is important to take into consideration not just user interactions with items, but also additional background information that can reveal user interests and help to select relevant items. Again, this is particularly important in cold-start scenarios where we need to recommend items to users who lack historical interactions.

Problems and Challenges of User Catalog

User catalogs, while important for personalizing recommendations in modern recommender systems, face several significant challenges. These issues primarily revolve around data privacy concerns, user identification difficulties, and the dynamic nature of user attributes. Addressing these challenges is crucial for maintaining the effectiveness and trustworthiness of recommender systems.

Data Privacy Concerns

In the context of increasing data privacy concerns, it‘s crucial for recommender systems to responsibly collect and utilize user data to deliver optimal user experiences and enhance product offerings. Regulatory frameworks like the GDPR provide essential guidelines for data handling, yet these should be viewed not as obstacles but as opportunities to foster trust and transparency in the digital ecosystem.

Responsible recommenders are pivotal in striking a balance between personalization and privacy. By employing data minimization strategies, pseudo-anonymizing user information, and ensuring robust data protection measures, recommender systems can offer highly personalized experiences without compromising user privacy.

User Identification Difficulties

Accurately identifying users is fundamental to creating and maintaining useful user profiles. However, several issues complicate user identification:

Multiple Profiles: Users may create multiple accounts on the same platform, leading to fragmented data that hinders a unified view of user preferences and behavior.

Shared Profiles: Accounts shared among several users, common in streaming services and online shopping platforms, present a challenge in discerning individual preferences, resulting in less personalized recommendations.

Cross-Device Identification: Users frequently access services across multiple devices, making it challenging to link these interactions to a single user profile accurately. These identification challenges can lead to inaccuracies in user profiles, impacting the relevance of recommendations and potentially diminishing user satisfaction.

Maintaining Up-To-Date User Attributes

User preferences, interests, and even geographic locations can change over time. Keeping user attributes up-to-date is important for the accuracy of recommender systems.

Changing Preferences and Interests: As users evolve, so do their preferences and interests. A recommendation system that fails to adapt to these changes may continue suggesting irrelevant items, leading to user disengagement.

Skills and Professional Changes: In domains like job recommendation systems, users' skills and professional interests may develop, requiring the system to adapt to these changes to remain relevant.

Mood Variability: User mood, which can influence content preference (such as music or movies), varies significantly. Capturing and adapting to these transient states poses an additional layer of complexity.

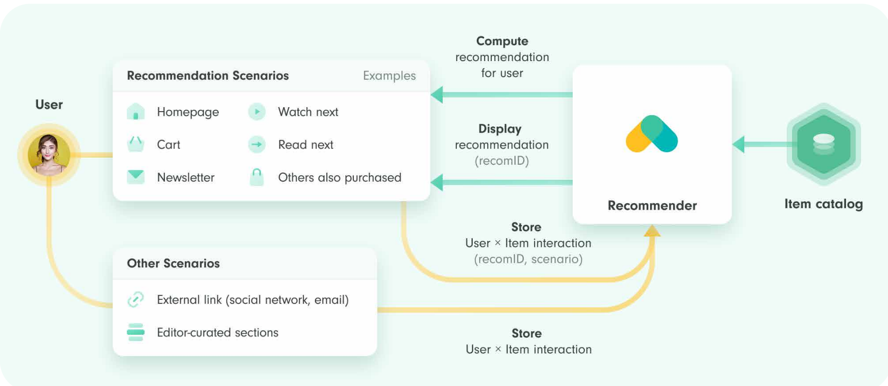
These challenges require online platforms to implement mechanisms for regularly updating user catalog and explicit user preferences. The alternative solution is to reduce reliance on user attributes and let recommender systems infer preferences of users from their interactions with items, incorporating feedback loops, and employing adaptive algorithms capable of adjusting to changes in user behavior and preferences.

Where subscription based services can typically supply recommender system with rich user profiles, online platforms that rely on advertising revenue can have as much as 70 percent of anonymous active users with short and recent interaction history. For such users, recommender systems rely on simple session based algorithms such as multi-armed bandits. When an anonymous user logs into the platform, the recommender system should be able to merge browsing histories.

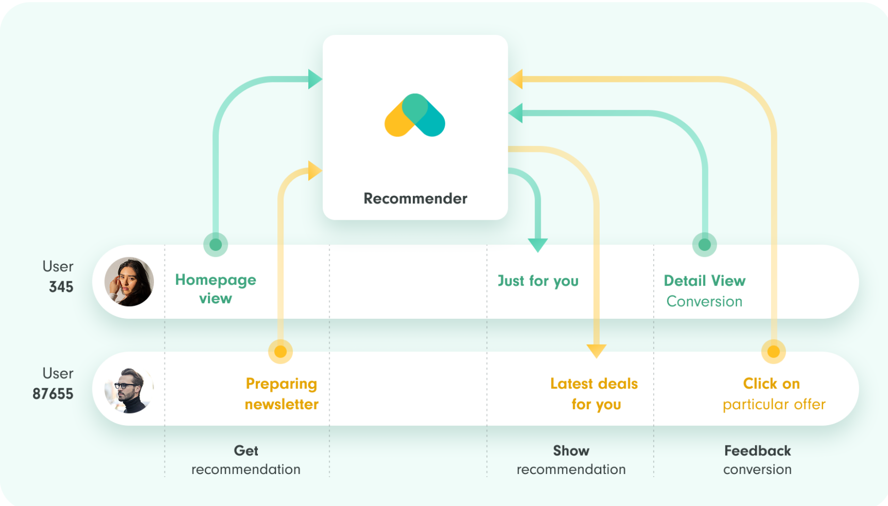
Modern platforms should be able to balance personalization with privacy and transparency. As recommender systems evolve, so too must the strategies for managing user catalogs, ensuring that they continue to offer relevant, timely, and engaging recommendations in a privacy-conscious manner. A good source of inspiration is recent developments in managing personal profiles for large language models.

User to Item Interactions

Interactions of users with items is the most important data source for recommender systems. In extreme cases, reasonable recommendations can be produced exclusively based on the interaction (or rating) matrix, where user to item interactions are typically stored. Such recommendations can be computed for anonymous users interacting with anonymous items, meaning that neither item attributes nor user attributes are used.



User interactions with items are collected in different scenarios, some of which are powered by a recommender system. There are a variety of user interactions that can be used to derive implicit feedback for recommender systems. These include ratings, browsing history, clicks, interactions with content (such as watching a video or liking a post), purchase history, search history and more. The data collected from these interactions can then be used to create user profiles and model user behavior, which can then be used to create personalized recommendations.



Typically, the recommender system is called upon to generate suggestions for a specific user in a given scenario. This request–response interaction follows a time sequence: the system returns a personalised list of items, which is then displayed to the user. If the user interacts with one of the recommended items, it's important to inform the recommender about this event, especially if the interaction was influenced by a particular recommendation. Depending on the scenario, this feedback can be nearly immediate or delayed by days (e.g. when the recommendation is part of a personalised newsletter sent via email).

Problems With Collecting User Feedback

Collecting and interpreting user feedback accurately is a cornerstone for the efficiency of recommender systems. However, several challenges complicate this process, impacting the quality of recommendations. Among these challenges, caching recommendations, biased user interactions, and the lack of explicit user feedback are particularly significant.

Caching Recommendations and Its Impact: To economize on the costs associated with recalculating recommendations for frequent users, some platforms employ a strategy of caching recommendations. This method can lead to reduced costs, improved response times, and provides users the opportunity to explore recommended items more thoroughly. However, this practice introduces a significant issue: users may repeatedly encounter the same items. If the recommender system is not notified of these repeated exposures and cannot adjust accordingly, it misses the critical opportunity to refine recommendations based on the user's demonstrated lack of interest in these repeated items. This oversight often results in a decline in user experience, as the system fails to recognize and adapt to the evolving preferences of the user.

Biased User Interactions: Bias in user interactions can significantly skew the data that recommender systems rely on. One form of bias, editorial bias, occurs when some recommendation scenarios are curated by editors and presented the same way to all users. Users typically click on several items from these curated lists, creating an artificial interaction similarity among items that are not genuinely similar. This phenomenon can mislead the recommender system into overestimating the relevance of certain items, thereby distorting the recommendation process.

Lack of User Feedback: Addressing the challenge of collecting user feedback, it's important to acknowledge that most users are reluctant to provide explicit feedback, such as rating items with stars or indicating likes and dislikes. A critical challenge for recommender systems is the absence of explicit or even implicit feedback in many scenarios. For instance, when users are recommended a list of articles and only read the excerpts without further interacting, they may still be satisfied with the recommendations. However, the recommender system receives no feedback signal to reflect this satisfaction. Similarly, in „autoplay“ scenarios for music or short videos, users often continue to watch or listen without active engagement, reacting only if the recommendation is particularly unsuitable. This passive consumption can falsely signal to the RS that the user is engaged, leading to misinterpretations of user interest and satisfaction.

Additionally, in scenarios where a recommender system generates a vast array of items but presents only a select few to the user, it becomes essential for the system to recognize that users may not view the recommendations positioned lower on the list. Misinterpreting a user‘s non-interaction with these less-visible items as a lack of interest can skew the system‘s perception of user preferences. Furthermore, there are instances where recommendations may not be seen by the user at all, such as when they are placed far down on a webpage and the user does not scroll sufficiently to encounter them. In such cases, the system‘s assumption that the user has seen and disregarded these recommendations is flawed. Ideally, recommendations should be requested and displayed to the user dynamically, minimizing the time gap between generation and presentation to ensure that users are exposed to relevant recommendations in a timely manner.

To effectively address these challenges, it‘s critical to enhance the quality of feedback loops and the accuracy of data provided to the recommender system. The more precise and comprehensive the user feedback, the more tailored the recommendations can be. For instance, tracking engagement metrics such as the portions of a video watched, segments of a song listened to, or parts of an article read can offer deeper insights into user preferences. Additionally, recommender systems need to employ advanced techniques to identify and correct biases, improve data quality, and develop methods for gauging user satisfaction beyond their immediate interactions. Furthermore, fostering an environment of transparency and encouraging users to offer direct and explicit feedback on the recommendations they receive can significantly improve the feedback loop, thereby elevating the overall performance of the recommender system.

Data stands at the core of modern recommender systems, fueling the generation of personalized and precise recommendations. The effectiveness of these systems hinges on their ability to leverage diverse data sources, including item catalogs, user catalogs, and user-item interactions. By understanding the attributes of both items and users, along with their interaction history, recommender systems can navigate the complexities of personalization, privacy, and changing user preferences to provide relevant recommendations. However, challenges such as difficulty of user identification, and the dynamic nature of user attributes necessitate advanced strategies to maintain the data useful for improving user experience. Furthermore, accurate collection and interpretation of user feedback are essential for refining recommendation algorithms and enhancing user satisfaction.

Modern Recommender Systems – Chapter 3: Objectives

Modern Recommender Systems – Chapter 3: Objectives

In the previous chapters, we explored the fundamental concepts of modern recommender systems and the crucial role of data in their functioning. While early recommender systems often relied on simple heuristics like „recommend what‘s popular“ or „suggest similar items“, modern approaches leverage sophisticated machine learning techniques to address complex recommendation challenges. Simple heuristics can work well for basic use cases; they are easy to implement, interpretable, and often computationally efficient. However, machine learning approaches can capture subtle patterns in user behavior, adapt to changing preferences, and optimize for multiple objectives simultaneously. In this third installment, we focus on objectives of machine learning in the area of recommender systems and search. Before you can use any machine learning system, it should be clear which learning objectives should be optimized. As existing literature often fails to cover this topic in sufficient depth, we explore learning objectives in detail here.

Examples of Learning Objectives

To give you an impression how broad objectives in recommender systems and personalized search can be, we will start with examples in various domains.

Content streaming services (e.g., music, video, podcasts) prioritize objectives centered around user engagement and retention:

Maximizing User Engagement: Keeping users actively consuming content (e.g., total view time, session duration, content completion).

Reducing Churn Rate: Minimizing users canceling subscriptions or ceasing to use the service.

Accelerating Content Discovery: Helping users easily find new, enjoyable content, showcasing catalog breadth.

Balancing Mainstream vs. Niche Content Exposure: Promoting diverse content to cater to varied tastes and support a healthy content ecosystem.

User Satisfaction and Perceived Value: Ensuring users feel recommendations are enjoyable and justify subscription costs or time spent.

Supporting Creator Ecosystem: Ensuring fair exposure and monetization for content creators.

In general, subscription-based content streaming services focus on optimizing user satisfaction with the service. For free users, the main objective would be to convert them into subscribers (e.g., by recommending highly relevant content beyond the paywall). For ad-powered content streaming services, watch time maximization might be a good strategy to increase revenue from displaying ads. Controversies are discussed in Chapter 1, Introduction.

E-commerce platforms typically prioritize objectives focused on driving sales and enhancing customer value:

Increasing Conversion Rates: Maximizing the percentage of users who make a purchase after viewing a recommendation or visiting the site.

Increasing Average Order Value (AOV): Encouraging users to purchase more items or higher-value items per transaction.

Reducing Cart Abandonment: Minimizing instances where users add items to their cart but leave without completing the purchase.

Maximizing Customer Lifetime Value (CLV): Fostering long-term customer loyalty and repeat purchases through sustained relevance.

Improving Product Discovery Across the Catalog: Helping users find relevant products beyond popular items or their immediate search.

Optimizing Inventory Turnover: Promoting overstocked items or those nearing end-of-season, balancing business needs with user experience.

Waking-up Inactive Customers: Offering targeted discounts or suggesting highly relevant products with limited availability.

In e-commerce, it is more about making the customer buy products rather than any other objectives. However, some e-commerce platforms and marketplaces are focusing on generating traffic for other e-shops rather than selling directly. Their focus is therefore shifted towards producing outclicks, especially when purchases associated with outclicks are not reported and rewarded by partner sites.

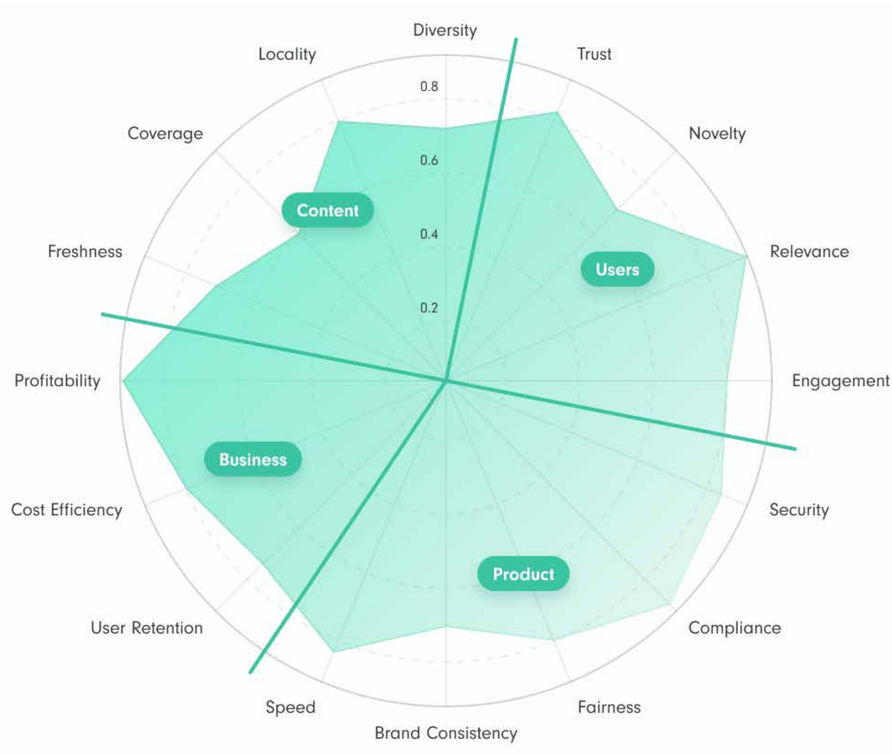
In other domains, objectives might be even more complex. Imagine job boards or dating sites that need to optimize for satisfaction of multiple parties under constraints.

Also, there are general level objectives that apply to most scenarios where users interact with personalized recommendations or search. User Satisfaction and Task Completion optimize for successful user sessions that result in items found in a reasonable time (Time-to-Result Optimization). One might also optimize for Relevance, Quality, Diversity, and Freshness of items. In all scenarios, we strive for Abandonment Reduction (e.g., search query, cart, watch next recommendation), which might lead to unsuccessful user sessions. Note that online platforms observe just partial user feedback signals, so all these objectives are typically optimized in a noisy environment.

One might ask how particular objectives are defined for a specific online platform and individual use cases. Typically, this is done through careful analysis of user needs, business requirements, and strategic objectives. These goals typically emerge from stakeholder discussions, user research, and business strategy sessions.

The objectives of modern recommender and search systems involve multiple stakeholders often with conflicting interests. Effective optimization seeks to balance and align these. We can broadly categorize these critical objectives as follows:

Taxonomy of Learning Objectives



User Objectives

Focused primarily on satisfying and engaging end-users, these metrics capture how effectively the system delivers personalized experiences.

Engagement: Measures the extent of active user interactions (clicks, views, session length, and return visits) indicating user interest and commitment to the content.

Relevance: Ensures recommended items align closely with user preferences, past behavior, and explicit user feedback.

Novelty and Serendipity: Goes beyond traditional relevance to introduce users to fresh, unexpected yet satisfying recommendations, keeping user experiences interesting and avoiding monotonous or predictable content.

Trust and Transparency: Users prefer transparent, explainable recommendations that build trust and confidence in the system's decisions, especially important in sensitive or high-stakes scenarios.

Content Objectives

These objectives ensure the breadth, richness, and balanced representation of available content.

Diversity: Guarantees variety within recommendations, preventing repetition, echo chambers, or overly similar content.

Coverage: Refers to the proportion of the content catalog effectively recommended and utilized, ensuring both niche and popular items have a fair opportunity for exposure.

Freshness: Prioritizes new or timely content, critical for domains where recency significantly impacts user satisfaction (e.g., news, trends, social media).

Locality: Ensures content relevance based on geographic, cultural, or regional context, where content that's highly relevant for users in one area may be irrelevant or inappropriate for others (e.g., local news, regional events, location-specific services, cultural content).

Business Objectives

Reflecting economic and strategic goals of an online platform, these metrics typically justify the investment in a recommender system or personalized search solution.

Profitability: Recommendations should directly or indirectly enhance revenue by increasing sales, upselling, cross-selling, or improving monetization opportunities.

Cost Efficiency: Systems should optimize resource utilization, reducing computational costs and data processing overhead.

User Retention and Loyalty: Strong recommendation systems support long-term customer relationships, reducing churn and boosting lifetime customer value.

Product Objectives

These objectives ensure that the recommender system contributes positively to the overall product experience, reputation, and ethical considerations.

Speed and Responsiveness: Recommendations must be fast and timely, ensuring that latency does not degrade user experience, especially critical in real-time scenarios.

Brand Consistency: Recommendations must align with the overall brand identity, supporting brand image and maintaining consistent messaging and quality expectations.

Fairness and Ethics: Recommenders should proactively avoid biases, stereotypes, or unfair treatment of user groups or content providers. Fairness also encompasses equitable representation and opportunities for less prominent content providers.

Compliance: Systems must responsibly handle user data and adhere to legal/ethical frameworks (e.g., GDPR, AI Acts), ensuring privacy and lawful processing.

Security: Systems must be resilient to malicious activities (e.g., attacks, hacking), safeguarding integrity, data, and reliability.

Balancing Objectives in Real-World Recommender Systems

The key to operationalizing these diverse objectives is aligning them with measurable metrics that can, in turn, be optimized through specific machine learning tasks (see next chapter). For example, user engagement might be measured via session length and click-through rates, while content diversity could be quantified using intra-list similarity scores. The star plot in Figure 4 illustrates how different objective categories like user objectives (engagement, relevance, novelty), content objectives (diversity, coverage, freshness), business objectives (profitability, cost efficiency, retention), and product/ethical objectives (speed, brand consistency, fairness) form a multi-dimensional optimization space.

A modern recommender system typically optimizes multiple objectives simultaneously through sophisticated multi-stage architectures. For example, Netflix's recommendation pipeline might first generate candidate items using collaborative filtering to maximize relevance, then apply diversity-aware re-ranking to ensure content variety, followed by business rule filtering to promote high-margin content, and finally apply fairness constraints to ensure equitable representation across different user demographics, all while maintaining sub-second response times to satisfy speed requirements.

The strategic importance of a data-driven approach to product development cannot be overstated in modern recommender systems. When organizations can systematically measure how their recommendation systems perform across different objectives, they gain unprecedented insights into user behavior patterns, system effectiveness, and optimization opportunities. This measurement capability transforms product development from intuition-based decisions to evidence-driven iterations.

The ability to quantify the impact of recommendations on user behavior, whether through A/B testing, user surveys, or behavioral analytics, enables teams to make informed decisions about which objectives to prioritize and how to allocate resources effectively. For instance, measuring how changes in recommendation relevance affect user engagement metrics allows teams to understand the causal relationships between different objectives and optimize accordingly.

Fortunately, many objectives in recommender systems exhibit positive correlations, creating virtuous cycles of improvement. When a system successfully improves relevance, it often leads to increased user engagement, longer session durations, and higher retention rates. Similarly, enhancing content diversity can simultaneously improve user satisfaction while maintaining or even boosting engagement metrics. This interconnected nature of objectives means that strategic improvements in one area can cascade into benefits across multiple dimensions, making the optimization process more efficient and impactful than it might initially appear.

The key to leveraging these correlations lies in establishing comprehensive measurement frameworks that track both primary and secondary metrics, enabling teams to identify which improvements create the most significant positive ripple effects across the entire objective landscape.

Platforms like Netflix and Spotify employ multi-stage pipelines where collaborative/content-based models generate initial candidates, then complex objective-aligned models re-rank them. Recommendation scenarios are explicitly defined (e.g., „homepage carousel," „watch next," „trending now") with dedicated optimization strategies. Open-source frameworks provide modular components for scenario and task definition, while regular evaluation against offline metrics and online experiments maintains alignment.

The key insight is that modern recommender systems don't optimize single objectives in isolation; they use sophisticated architectures that simultaneously balance multiple competing goals through careful task formulation and multi-stage processing.

Customizing Recombee to Meet your Objectives

Recombee's recommendation engine uses modular Logics (algorithms/ ensembles) and Scenarios (named use-cases) to optimize for a wide range of objectives.

Scenarios
A Scenario in Recombee represents a specific place in the application where recommendations are shown, such as a box on a product detail page, a watch-next screen, or a newsletter slot. Each Scenario defines the context and purpose of recommendations for that particular use case, creating a named configuration that can be easily managed by product or editorial teams within the Recombee web interface.
When an application requests recommendations using a particular Scenario ID, Recombee executes the defined configuration to deliver a precisely tailored and contextually appropriate list of items for that specific use case.

Logics

At the heart of every Scenario is a Logic—a named ensemble of recommendation models. Recombee provides a variety of Logics that are either universal or domain-specific, enabling targeted optimization for each industry. Many of these Logics have additional parameters for tuning their behavior (e.g., whether to recommend already watched content or not).

Universal Logics

These are applicable across domains and address general-purpose personalization:

- **recombee:personal** – Personalized ranking of items for a user, based on the user‘s interaction history and user properties, typically used on homepages or dashboards.
- **recombee:similar-items** – Items similar to a given item (both interaction-wise and content-wise), commonly used on detail pages.
- **recombee:popular** – Items that get a lot of interactions within the whole user base, or within a specific user segment.

Domain-Specific Logics

Recombee also provides Logics fine-tuned for specific verticals:

- **Video & OTT:** video:watch-next, video:continue-watching, video:editors-picks, etc.
- **News & Media:** news:daily-news, news:trending, news:categories-for-you, etc.
- **E-commerce:** ecommerce:cross-sell, ecommerce:similar-products, ecommerce:bestseller, etc.

These Logics incorporate domain-specific behaviors, signals, and diversity models out of the box.

Custom Settings and Rules

In addition to selecting an appropriate Logic for each Scenario, Recombee allows fine-tuning each recommendation request through various custom settings and rules that help align the system with specific objectives:

- **Filters:** Rules to limit which items can appear (e.g., hide out-of-stock products, recommend only articles from certain categories and of a certain age).
- **Boosters:** Rules that bias the recommender engine toward recommending certain items more (e.g., promote discounted items or recent articles).
- **Constraints:** Rules that enforce diversity in recommended items (e.g., limit the number of items from a single brand in a recommendation).

These customizable elements allow organizations to adapt the recommendation behavior to their specific business requirements, editorial policies, and user experience goals without modifying the underlying machine learning models.

How Recombee Logics & Scenarios Optimize Diverse Objectives

User Objectives

Focused primarily on satisfying and engaging end-users, these metrics capture how effectively the system delivers personalized experiences that create value for the people actually using the platform.

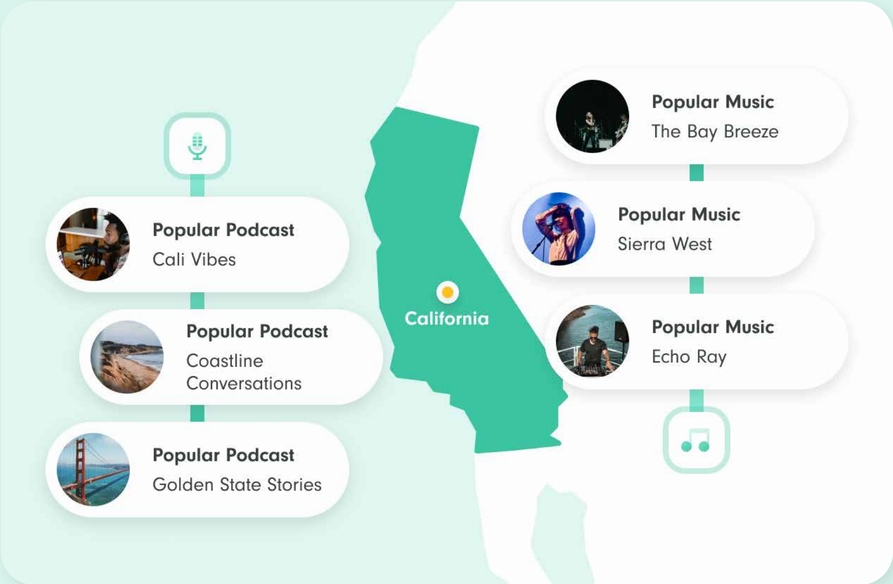
- **Engagement:** Infinite feed scenarios with fresh content after refresh on next visit. Customer Lifetime Value (CLV) optimization through sustained interaction patterns. Logics like video:continue-watching and news:daily-news maintain user interest across sessions, while the automatic exploration algorithms prevent content fatigue.
- **Relevance:** Automatic optimization through recombee:personal and similar logics that learn from user behavior patterns and preferences. Many logics that do not have “personal” explicitly stated in their names still utilize smart algorithms to ensure recommended items are relevant for a particular user.
- **Novelty and Serendipity:** Automatic optimization through user history analysis and exploration algorithms that introduce users to unexpected but relevant content. Diversity constraints prevent filter bubbles, while logics like video:editors-picks surface curated content users might not discover organically.
- **Trust and Transparency:** Built-in data protection (avoiding external data enrichment) with comprehensive tools and insights for recommender system operators and editors to understand and explain system behavior. Clear scenario naming and logic selection help users understand why certain content is being recommended.

Content Objectives

These objectives ensure the breadth, richness, and balanced representation of available content, preventing the marginalization of niche or emerging content while maintaining editorial quality and brand standards.

- **Diversity:** Automatic diversity optimization through exploratory algorithms and configurable constraints that ensure recommendation slates are diverse across multiple dimensions (genre, topic, creator, recency). Constraints prevent over-concentration of similar items, while boosters can promote underrepresented categories.
- **Coverage:** Automatic recall-coverage tradeoff optimization ensuring niche users discover niche content, preventing the long-tail from being overlooked. Special algorithms like beeFormer are capable of recommending content without interactions using semantic attribute similarity.
- **Freshness:** All news logics incorporate automatic exploration of recent content. Dedicated scenarios with time-based filters ensure timely content surfacing, while boosters can prioritize newly published items. This prevents recommendations from becoming stale and ensures users stay current with latest developments.
- **Locality:** Boosting content by geographic distance and user location preferences, enabling region-specific and culturally relevant recommendations. Filters can restrict content to specific regions, while location-aware logics surface content that resonates with local interests and cultural context. See Recombee online blogpost for more.

ONLINE
BLOGPOST



Business Objectives

Reflecting economic and strategic goals of online platforms, these metrics typically justify the investment in recommender systems and align recommendation strategy with revenue generation and operational efficiency.

- **Profitability:** E-commerce upsell and cross-sell through business rules and logics like ecommerce:cross-sell; subscription-based services balance engagement-only content for free users with premium content promotions to drive conversions through strategic boosters; increased page views generate more ad impressions through optimized infinite scroll scenarios; affiliate and outbound click optimization through targeted boosting of monetizable content.
- **Cost Efficiency:** Recombee runs a private cloud with almost thousand servers across the globe. All algorithms and data storage systems are implemented in an extremely efficient way to reduce operational overhead while providing enterprise-grade performance and reliability.
- **Customer Retention and Loyalty:** Personalized experiences through recombee:personal and news:daily-news foster loyalty through niche content

discovery and habit formation. Diversification models prevent filter bubbles and maintain long-term engagement by introducing variety that keeps users returning over extended periods.

Product Objectives

These objectives ensure that the recommender system contributes positively to the overall product experience, reputation, and ethical considerations while maintaining technical excellence and regulatory compliance.

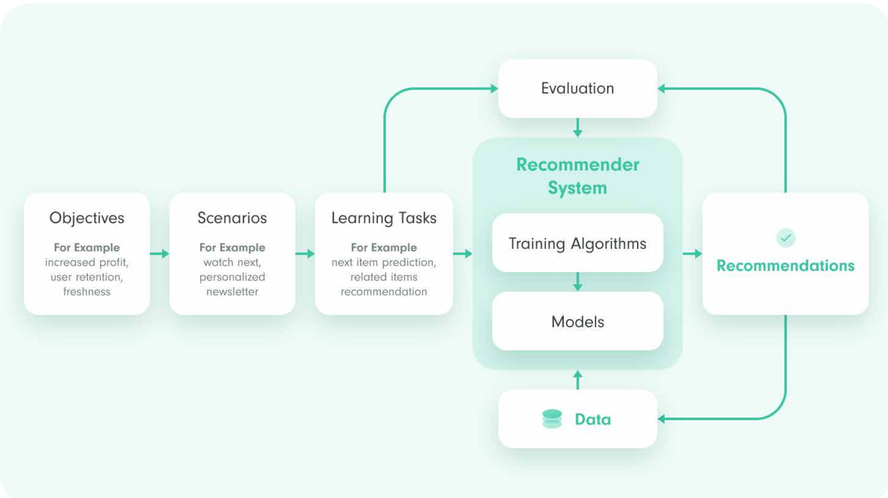
- **Speed and Responsiveness:** Automatic performance optimization ensuring low-latency recommendations across all scenarios, with sub-100ms response times that don‘t degrade user experience. Efficient massively parallelized algorithms and data pipelines maintain performance even under high load.
- **Brand Consistency:** Logics like video:editors-picks combined with filters, boosters, and constraints enable curated content that aligns with brand values and editorial standards. Custom filters ensure only brand-appropriate content appears in recommendations, while boosters can promote content that reinforces brand identity.
- **Fairness and Ethics:** Built-in algorithmic fairness measures and bias mitigation strategies deployed automatically across all recommendation scenarios. Diversity constraints prevent discrimination, while balanced exposure algorithms ensure equitable treatment of content creators and fair representation across demographic groups.
- **Compliance:** Automatic adherence to data protection regulations (GDPR, CCPA) and industry standards without requiring manual configuration. Privacy-by-design architecture ensures user data is handled securely, while audit trails provide transparency for regulatory review.
- **Security:** Automatic security measures protecting against malicious attacks (recommendation poisoning, data breaches) and ensuring system integrity. Rate limiting, input validation, and secure data handling protect both the platform and its users from potential threats.

Beyond the configurable logics, Recombee inherently manages several critical aspects to ensure a high-quality, reliable service. The system‘s architecture is built from the ground up for speed and scalability, consistently delivering recommendations with low latency, even under high-demand scenarios. Security and user privacy are foundational pillars, handled at the platform level in adherence with best practices and regulatory requirements, without necessitating direct user configuration. Furthermore, many core recommendation approaches, such as those powering homepages or email campaigns, incorporate built-in mechanisms to promote diversity and fair exposure of items. This proactive approach helps prevent users from being confined to filter bubbles and ensures a broader range of content gets a fair opportunity to be discovered.

Recombee‘s modular Logics and Scenarios provide a flexible, robust way to optimize for a broad spectrum of recommendation objectives—many of which are handled „out of the box“ by the system, freeing teams to focus on high-level strategy rather than low-level tuning.

In summary, defining objectives in recommender systems is an iterative, stakeholder-driven process that balances competing user, content, business, and product goals. Modern systems must navigate complex trade-offs between engagement and diversity, relevance and coverage, profitability and fairness; all while maintaining technical performance and ethical standards.

Key lessons from industry practice show that successful objective definition requires measurable metrics that can be tracked over time, clear prioritization when objectives conflict, and regular reassessment as business priorities evolve. The most effective systems establish objective hierarchies where primary goals (like user engagement) are supported by secondary objectives (like content diversity) that prevent long-term degradation.



The next booklet will be about scenarios, learning tasks, training algorithms and evaluation procedures.

In the next booklet, we will examine how these diverse objectives can improve your product in multiple scenarios. How we translate them into specific machine learning tasks, how to train models using different algorithms and how to evaluate the performance of recommender systems.

Explore Upcoming Chapters and Access the Full Web Version, Including a Downloadable PDF.

recombee.com/handbook/modern-recommender-systems

How to Use This Booklet

How to Use This Booklet

This booklet is designed for product managers, technical leads, data scientists, and business strategists who want to understand what it really takes to deploy a modern recommender system. Whether you are planning your first deployment or optimizing an existing solution, you'll find practical insights, common pitfalls, and best practices drawn from real-world experience. Technical readers will appreciate the depth of data and objectives, while business readers will find guidance on aligning strategy and technology. Use the checklists and sidebars to quickly assess your readiness and avoid common mistakes.

Checklist: Aligning Stakeholder Objectives

- Are your business, user, and content creator goals clearly defined and not in conflict?
- Have you identified potential misalignments (e.g., over-optimizing for ad revenue at the expense of user experience)?
- Is there a process for regularly reviewing and updating objectives as your product evolves?

Lesson Learned: A media client once focused solely on maximizing page views, only to see user churn increase. After shifting to a balanced set of objectives, including user satisfaction and content diversity, long-term engagement improved.

Checklist: Is Your Data Ready for Recommendations?

- Is your item catalog complete, accurate, and regularly updated?
- Are user profiles and interaction histories being collected in a privacy-compliant way?
- Do you have a plan for handling anonymous users and merging session data?

- Are you tracking both explicit and implicit feedback?
- Have you identified and addressed common data quality issues (e.g., missing attributes, duplicate items)?

Lesson Learned: An e-commerce client improved conversion rates by cleaning up duplicate product entries and enriching item metadata, which led to more relevant recommendations and fewer user complaints.

Common Pitfall: Relying only on click data can lead to misleading signals. Track deeper engagement metrics (e.g., watch time, scroll depth) for a more accurate picture of user preferences.

Checklist: Are Your Objectives Measurable and Actionable?

- Have you translated high-level business goals into measurable metrics (e.g., CTR, session length, diversity scores)?
- Are you balancing short-term and long-term objectives (e.g., engagement vs. retention)?
- Do you regularly review objective metrics with all stakeholders?
- Do you revisit scenarios and logics assignment according to your current objectives?
- Is your system architecture flexible enough to adapt as objectives evolve?

Lesson Learned: A job board using Recombee initially optimized for application count, but after switching to successful hires as the main metric, both employer satisfaction and user trust increased.

Common Pitfall: Optimizing for a single metric (like clicks or purchases) can create unintended negative effects. Always monitor secondary metrics to catch emerging issues early.

Glossary

Glossary

Abandonment
When a user disengages from a process (like a shopping session, article, or video) before completing the intended action (e.g., purchase, full view, sign-up). Important signal in measuring user satisfaction or frustration.

Active Items
Items currently available for recommendation. Opposite of historical items, which are no longer available but still useful for similarity computation.

Anonymous User
A user who interacts with a platform without being logged in or identified. Presents a challenge for personalization due to limited or no historical data.

Attribute (Item/User Attribute)
Structured information associated with items or users—such as genre, price, location, or age—that helps model their characteristics and improve recommendation quality.

Booster
A rule or function that increases the likelihood of certain items being recommended by adjusting their ranking score based on defined criteria (e.g., recency, margin, editorial picks).

Candidate Generation
The first stage in many recommender pipelines where a broad set of potentially relevant items is retrieved before re-ranking.

Catalog
See Item Catalog or User Catalog. A structured representation of items or users, containing metadata used in recommendations.

Cold Start
The challenge of recommending items or content to new users or for new items with little to no historical interaction data.

Collaborative Filtering
A family of algorithms that use historical interactions from multiple users to predict what an individual user might prefer. Includes user-based, item-based, and matrix factorization techniques.

Content-Based Filtering
A technique that recommends items similar to those a user previously liked, based on item attributes such as tags, category, or description.

Context-Aware Recommendation
Recommending items based not only on user/item history but also on situational factors like time, location, or device used.

Coverage
The proportion of the item catalog that is recommended over time. A key metric for ensuring diverse and fair item exposure.

CTR (Click-Through Rate)
A standard metric used to evaluate recommender systems. Measures how often users click on recommended items relative to how often they are shown.

Data Minimization
A principle in data privacy advocating for the collection of only the data necessary for a given task. Often supported by pseudo-anonymization and aggregation techniques.

Diversity
A measure of variety within a recommendation list. Helps prevent repetitive suggestions and filter bubbles, and ensures exposure to a wider range of items.

Engagement
Any measurable user interaction that indicates interest (e.g., views, clicks, scroll depth, watch time). Often used as a proxy for user satisfaction.

Explicit Feedback
Direct user input such as ratings, thumbs up/down, or likes, indicating preferences. Easier to interpret than implicit feedback, but less common.

Fairness
The principle that recommendation outcomes should not systematically disadvantage any user group or content category. Can be enforced through constraints and diversity mechanisms.

Freshness
How recently an item was created or interacted with. Important in domains like news or trends.

Implicit Feedback
Indirect evidence of preference gathered from behavior such as viewing, watching, listening, scrolling, or purchasing.

Infinite Feed
An interface design pattern where content is continuously loaded as the user scrolls. Often used with recommender systems to keep users engaged.

Item Catalog
A structured database of items that can be recommended, including both active and historical items. Each item has a set of attributes.

Logics (Recombee)
Named ensembles of recommendation algorithms in Recombee tailored for particular use-cases or industries (e.g., video:watch-next, ecommerce:cross-sell).

Long Tail
The portion of the item catalog that consists of less popular, niche items. Recommender systems aim to surface relevant long-tail content to the right users.

Multi-Armed Bandit
An algorithmic framework for balancing exploration (trying new items) and exploitation (recommending known favorites), particularly useful for anonymous or new users.

Objective (Learning Objective)
A measurable goal a recommender system tries to optimize, such as user engagement, profitability, or content diversity.

Offline Metrics
Evaluation scores (e.g., precision, recall, MAP, NDCG) computed on historical data, used to assess model quality before deployment.

Online Metrics
Live performance indicators such as CTR, conversion rate, or bounce rate, measured during real-world usage via A/B testing.

Outclick
A user clicking a link that leads to an external destination (e.g., third-party site). Common in aggregators or affiliate platforms.

Personalization
The process of tailoring recommendations to individual user preferences, behaviors, or context.

Popularity Bias
A phenomenon where recommenders over-recommend popular items at the expense of niche or new content, reducing diversity and fairness.

Relevance
The extent to which a recommended item aligns with the user's current preferences or needs.

Re-ranking
The process of taking a candidate list of recommended items and adjusting their order to better meet multiple objectives (e.g., diversity, profitability).

Scenarios (Recombee)
Named contexts in which recommendations are requested (e.g., homepage, email newsletter, product detail page). Each Scenario can use specific Logics and configuration rules.

Serendipity
The quality of providing surprising yet useful recommendations that users weren't actively seeking but appreciate.

Session-Based Recommendation
A type of recommendation that relies solely on current session behavior rather than long-term user history. Useful for anonymous users.

Similarity (Item/User)
A computed score indicating how alike two items or users are, based on attributes or interactions. Used in collaborative and content-based filtering.

Stakeholder
Any party involved in or affected by the recommendations—typically users, content creators, product owners, advertisers, and business leaders.

Trust and Transparency
The degree to which users and operators understand how a recommender system works. Critical for sensitive domains and for building confidence in personalization.

User Catalog
A structured collection of user attributes, such as location, interests, or past queries. Helps enrich recommendations, especially in cold-start situations.

User Satisfaction
A subjective but critical goal in recommender systems. Can be inferred from engagement metrics, retention, feedback, or direct surveys.

Watch Time
A metric used particularly in video content platforms to track how long users watch recommended items. Often used to optimize engagement.

Recombee

recombee.com