Journey to Centralising Destination Recommendations

Maria Prosviryakova[1], Gaurav Misra[2], Sébastien Le Digabel[3] and Rodrigo Villatoro[4]

[1]Senior Data Scientist, Skyscanner Limited, Barcelona, Spain
[2]Senior Data Scientist, Skyscanner Limited, London, UK
[3]Principal Engineer, Skyscanner Limited, London, UK
[4]Principal Data Scientist, Skyscanner Limited, Barcelona, Spain

Abstract

Recommending new and relevant destinations to travellers is one of the most important use cases for Skyscanner. Internal research shows that more than 51% of Skyscanner users have not yet chosen their next destination, making them more open to recommendations. Recent advances in recommendation models for the travel industry help address the question of "what" to recommend, but various practical challenges remain for companies to efficiently serve these recommendations to users. For instance, how to deal with diverse set of product items (countries, cities, airports, etc.,), how to tackle the cold start problem, and how to increase adoption of recommendations across the organisation. In this work we describe Skyscanner's approach to solving these problems by migrating from a decentralised to a centralised destination recommendation architecture. This migration has increased the number of frontends serving recommendations, reduced redundant implementation efforts, and accelerated the experimentation pace, among other benefits.

Keywords

Recommender Systems, Personalisation,

1. Introduction

Online platforms use recommender systems to present the most relevant content to their users, in an effort to minimise information overload [1]. For online businesses, recommendations not only enhance user satisfaction, but also positively impact the business. Past research [2] indicates that 75% of online shoppers are more likely to buy products based on personalised recommendations. Recommendations also boost customer loyalty, as users who click on them are nearly twice as likely to revisit the site [2]. This evolving paradigm of online behavior makes the ability to deliver personalised content a key business driver for online businesses [3].

Online travel platforms have also leveraged recommendations to serve their users [4, 5, 6]. Planning a trip is a complex task for travellers and they can benefit from receiving personalised content. Therefore, using intelligent algorithms to personalise results according to diverse user motivations, preferences, budgets, etc., expressed by them through implicit or explicit signals, becomes imperative for online travel platforms like Skyscanner [7], and has led to the development of numerous types of recommendation models [5, 8].

At Skyscanner, we leverage recommendation systems in various parts of our product, across all our verticals (i.e. flights, hotels and car-hire). In this paper, however, we focus solely on destination recommendations.

Our user research at Skyscanner indicates that 51% of travellers are undecided about their next trip when visiting our website. Furthermore, 30% of them are specifically interested in exploring various destinations. We use destination recommendations to inspire our travellers, guide them through the exploration phase, and help them find the most suitable deal on our site. We have multiple models that recommend different types of destination (i.e. airport, city and country). In the rest of this paper, we discuss the challenges we encountered when implementing destination recommendations at scale to

Workshop on Recommenders in Tourism (RecTour 2024), co-located with the 18th ACM Conference on Recommender Systems, October 18th, 2024, Bari, Italy
maria.prosviryakova@skyscanner.net (M. Prosviryakova); gaurav.misra@skyscanner.net (G. Misra); sebastien.ledigabel@skyscanner.net (S. L. Digabel); rodrigo.villatoro@skyscanner.net (R. Villatoro)

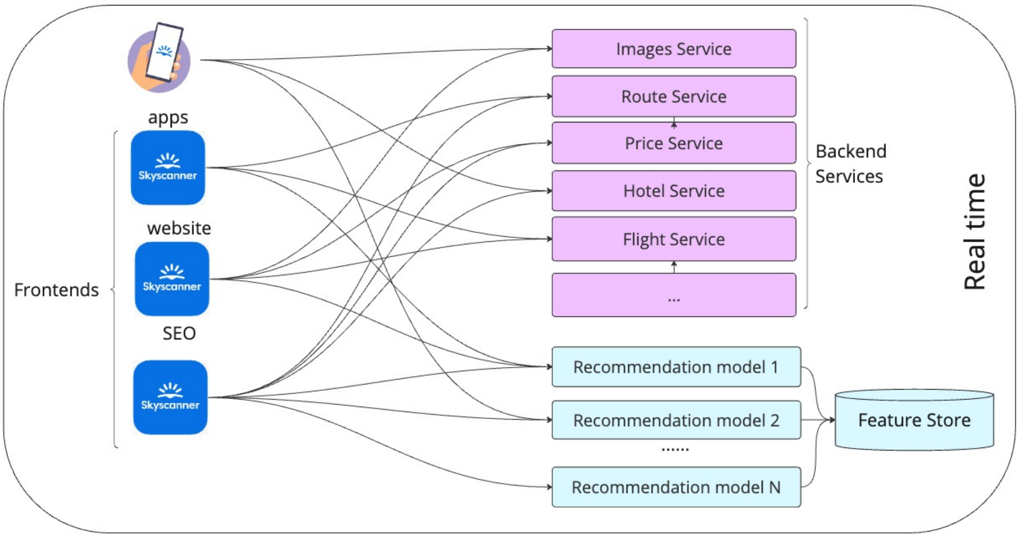


Figure 1: Previous architecture with frontend integrations with multiple backend services and recommendation models

multiple Skyscanner frontends, how we created a centralised architecture to mitigate those challenges and what impact we have seen with the new architecture.

2. Challenges for Destination Recommendations at Skyscanner

2.1. Multiple isolated frontend integrations

Recommendation models are often purpose-built for each frontend touchpoint, which is ultimately responsible for constructing the destination cards displayed to users. As a result, each frontend must separately integrate with various services—such as images, destination labels, and price services—to present enticing recommendations to travelers. This architecture is shown in Figure 1.

We realised that this architecture did not scale well when attempting to increase the adoption of destination recommendations across Skyscanner's frontend clients. It led to duplicated efforts, as each frontend had to integrate separately with recommendation models and backend services for their specific use case. For data scientists, this architecture introduced additional challenges, such as monitoring model performance, maintaining expected model behaviour, and measuring overall impact of recommendations on the business. They had limited visibility into how frontend clients were using the recommendations (e.g. whether clients applied any post-filtering logic) and how travellers interacted with the recommendations, as each frontend could implement custom logging.

2.2. Distinct Recommendation Items

Travellers visiting Skyscanner can explore and book flights, accommodations, and rental cars. This diversity of verticals means that there are numerous frontends where destination recommendations can be presented to users. These frontends can serve different search intents (e.g. "exploring everywhere" vs "searching cities in a specific geographical area") and allow travellers to express these intents in various ways. As a result, different recommendation models are required to recommend distinct types of destinations—i.e. countries, cities, or airports—depending on where the user is in their exploration journey. This introduces another challenge for the frontend client, which is determining the appropriate recommendation model.

2.3. Selecting Suitable Recommendation Models

Frontend clients had to be deliberate in choosing the appropriate recommendation model for their use case, and pass the required parameters to the model. This challenge was compounded by the lack of visibility into available recommendation models, as there was no centralised repository or model catalog. Each integrating client had to contact the recommendations team to enquire about available models, determine if any were suitable for their use case, and assess whether they could utilise them by providing the required search context. The recommendations team, in turn, had to handle these requests on a case-by-case basis, making it difficult to apply learnings from previous integrations.

2.4. Level of personalisation

Sparse interaction data is often a common problem for recommendation algorithms and this is exacerbated in the travel industry due to the inherent infrequent nature of user activity on these sites [5, 6]. While most recommendation algorithms typically personalise at the user level, travel sites often experience the cold start problem when new travellers land on the site for whom no historical interaction data is available. This also includes travellers who do not consent to having their activity tracked.

Skyscanner doesn't require users to be logged in to explore destinations, hence we can't have a long history of user interactions with destinations, because user cookies expire. Moreover, a long history of interactions with the destinations may not always be relevant, because travellers want different experiences and can have polar-opposite preferences from one trip to another (e.g. a safari get-away to Masai Mara in May and a business trip to London in September). This means that our recommendation platform needs to be flexible to provide recommendations for all types of users.

The more we know about the traveller's preferences from their past interactions, the more relevant recommendations can be provided. While most of our recommendation models have built-in fallbacks for cold start cases, it is still good practice for frontend clients to implement their own fallbacks in case a model can't generate recommendations. However, this poses the challenge of limited observability of these fallbacks being used and what recommendations they provide to the traveller. Additionally, different fallback logic across Skyscanner frontends can result in an inconsistent user experience for travellers navigating across our product.

3. Centralized Destination Recommendations Architecture

The new architecture with a centralised destination recommendations service was designed and developed to address the challenges outlined in the previous section, with the primary goal of increasing internal adoption of destination recommendations across Skyscanner frontends. To achieve this, the new architecture had to meet two key requirements:

- Simplify the integration processes for frontend clients wanting to use recommendations.
- Relieve frontend clients from having to decide which recommendation model is the most suitable for their specific use case.

Figure 2 illustrates how a traveller's query goes through the new architecture to return destination recommendations. When travellers use Skyscanner for exploration, the frontends (1) capture information about their search context (only for those who have provided consent), and send these data as a request to the Unified Search Service (USS) (2). The USS formats the request and forwards it to the Destination Recommendations Service (3) which retrieves recommendations from the Model Serving Platform (4), that hosts multiple recommendation models (5). Once the USS receives the recommendations, it calls backend services to enrich them with relevant information before sending the results back to the frontend (6).

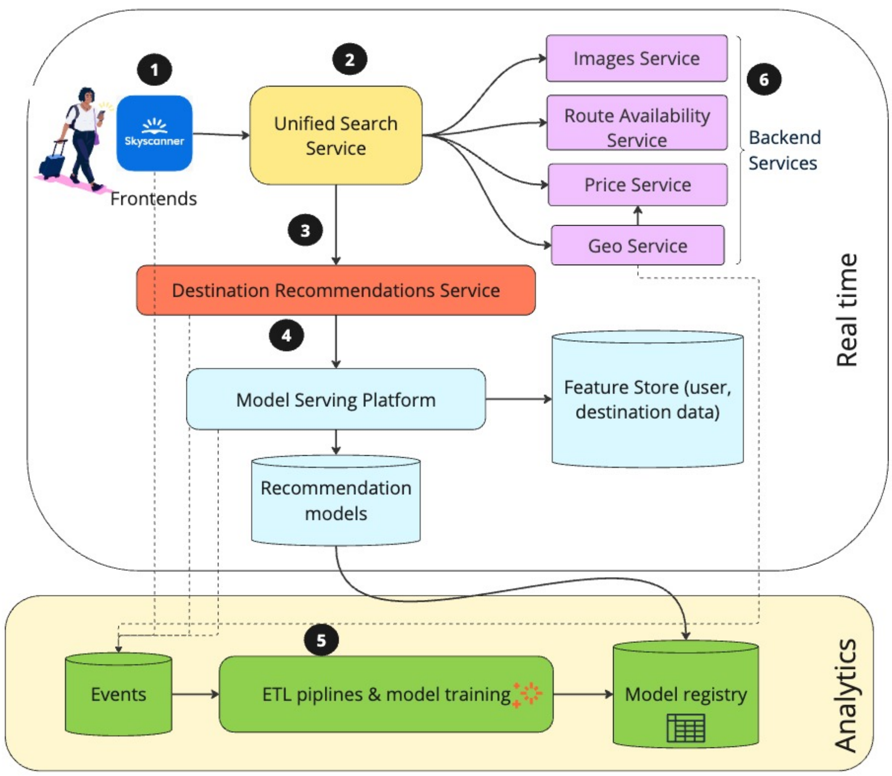


Figure 2: The new centralised architecture for destination recommendations

3.1. Skyscanner Exploration Frontends

Skyscanner travellers can begin searching for their next trip from various Skyscanner frontends (e.g. main homepage, flight, hotel, or car hire search pages, or multiple landing pages). Travellers may either search for specific destinations (e.g. "flights from London to Bari"), or explore with open-ended searches (e.g. "flights everywhere" or "flights from Bari to Spain").

The process described in Figure 2 only applies to open-ended searches. These searches help us capture various traveller intents and require different types of recommendations, such as countries, cities and airports. Examples of recommendations for different traveller intents are shown in Figure 3.

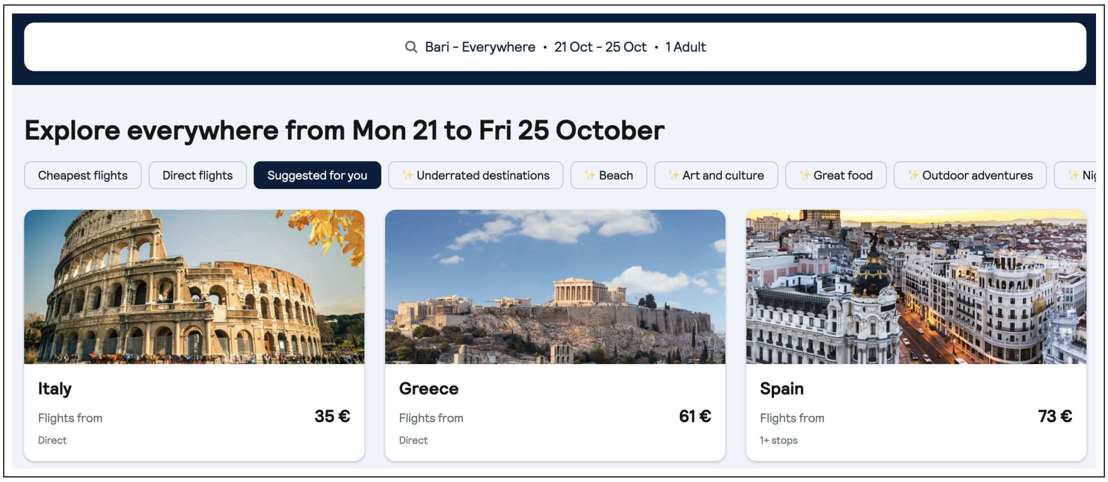
3.2. Unified Search Service (USS)

This service is a single point of entry to Skyscanner's search-related backend services. Its goal is to provide standardised and easy access to distinct types of information for flights, hotels, cars through a unified service API. In other words, it abstracts frontend clients from the need to directly integrate with each backend service.

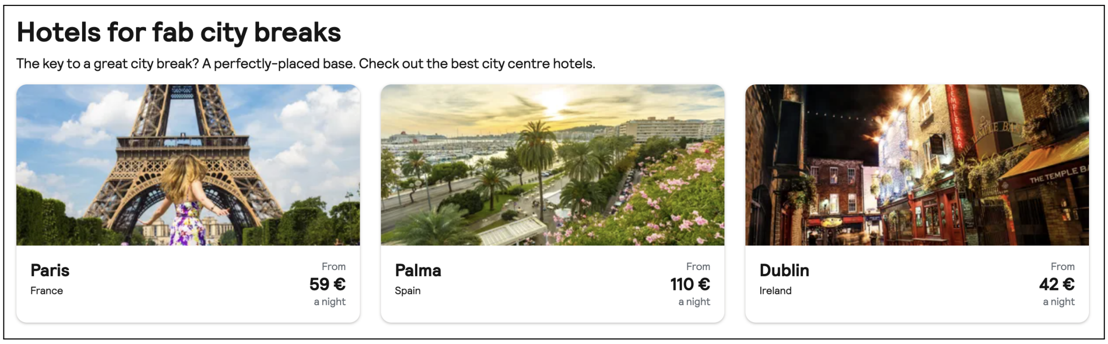
When the USS receives an open-ended search, it forwards the request to the Destination Recommendations Service to obtain relevant recommendations. Once the USS receives the recommendations, it queries multiple backend services to enrich them with geographical information, prices, images and checks flights availability.

3.3. Destination Recommendations Service

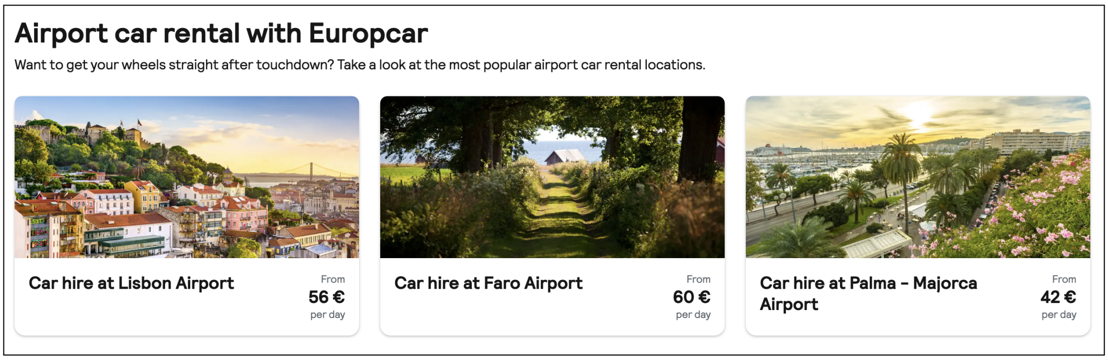
The Destination Recommendations Service acts as a routing layer with the following responsibilities:



(a) "Explore Everywhere" feature on Skyscanner recommending countries to inspire travellers



(b) Skyscanner landing page widget with relevant cities where travellers might want to book hotels



(c) Skyscanner landing page widget with most popular airports to rent cars from Europcar

Figure 3: Screenshots from Skyscanner website showing different types of destination recommendations across the platform

- It extracts the search intent from the context of the incoming request and selects the most relevant recommendation model.
- It modifies the request to retrieve recommendations from the model hosted in the Model Serving Platform.
- It returns recommendations to the USS.

The service also includes fallbacks for each model, used in some cold start cases or if a model is unable to return recommendations. These fallbacks are sensible defaults that provide some non-refined results to the average user (e.g. most popular destinations).

3.4. Model Serving Platform

This platform hosts the majority of Skyscanner's machine learning models used for real-time inference. It also provides access to a low-latency Feature Store that can be used to enrich models at inference time with traveller preferences and destination features. For example, in item-to-item collaborative filtering models, we can retrieve a traveller's recent searches from the Feature Store and use this data to recommend similar destinations.

The Feature Store can be also queried directly to obtain relevant content without needing to call a model. For example, a service that constructs marketing email to re-engage users queries Feature Store directly to obtain personalised destination recommendations generated in a batch process.

Additionally, the Model Serving Platform emits logs for model requests and responses, which are used for ML observability purposes and model optimisation.

3.5. ETL and Model Training Pipelines

Traveller interaction data and metadata for various destination types are processed using ETL pipelines. These pipelines operate at different frequencies and are also used to train various recommendation models. The trained models are stored in a model registry and are made available for inference through the Model Serving Platform.

Some outputs of these ETL pipelines are also saved to the Feature Store. Examples include user features used to personalise models based on recent traveller preferences and searches, destination popularity scores that serve as model fallbacks, or destination vibes.

4. Results

In this section, we present the results and business impact following the roll-out of the new architecture almost in 2023 (almost a year ago).

4.1. Deduplicated Integration Effort

The new architecture removed the need for each frontend client to integrate individually with recommendation models and multiple backend services. This streamlined process reduced implementation overhead and minimised duplication of efforts, leading to more efficient resource allocation.

4.2. Abstracted Model Selection

Frontends no longer need to determine which specific recommendation model is suitable for their use case. The Destination Recommendations Service now handles this by interpreting the request and routing it to the most relevant model.

4.3. Centralised Fallbacks

Since the Destination Recommendations Service includes fallbacks, frontend clients no longer need to create their own solutions for cold start cases or when a model returns no results. Centralising the fallbacks ensures the quality of the single source of truth and maintains consistency of user experience across different frontends.

4.4. Improved Observability

The new architecture has improved the observability of model performance across multiple clients, leading to better standardised monitoring and enhancements in recommendation quality. It has also standardised alerting and accelerated incident response times.

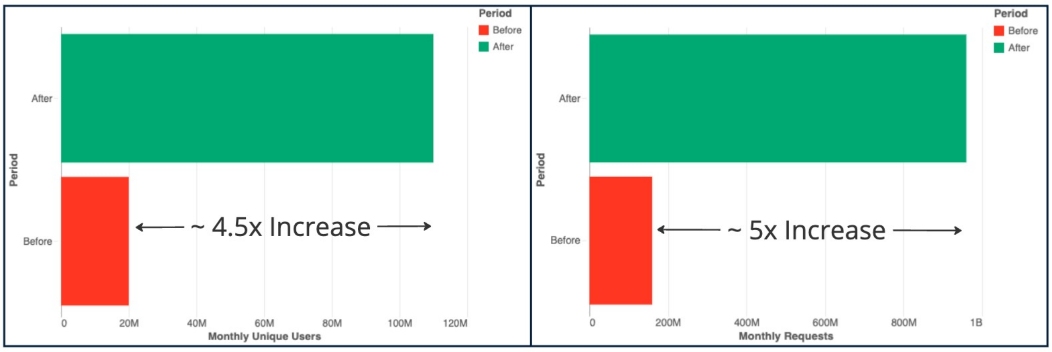


Figure 4: Total monthly unique users (left) and requests (right) received by destination recommendations models before and one year after the roll-out of the new architecture.

We have implemented Service Level Objectives (SLOs) for all integrating clients, with the service now expected to respond to all incoming requests within 100 milliseconds. These improvements significantly improved the overall experience for integrating clients.

The rate of experimentation also notably increased due to the streamlined integration and decision-making processes, as well as the accelerated model development enabled by the new architecture. For data scientists, this enhanced observability of user interactions with recommended destinations across Skyscanner frontends has offered valuable insights into the performance of recommendation models in different contexts, enabling more effective decision-making regarding model retraining and maintenance.

4.5. Enhanced Adoption of Recommendations

The simplified integration and all the added benefits of the new architecture have led to a significant increase in the use of destination recommendations across Skyscanner.

Prior to the development of the new architecture, we had 3 frontends using recommendation models in their exploration flows. Potential frontend clients, who wanted to leverage recommendations in their touchpoint, would always highlight the extra effort and complexity of integrating with multiple backend services to create the destination cards as a blocker. However, since the roll-out of the centralised service to clients in May 2023, we have seen a steady increase in the number of frontend clients that have integrated due to the simplified process.

Currently, destination recommendation models are used by **8 Skyscanner frontends** and handle **960 million recommendation requests** to **110 million Skyscanner users** from **180 countries** every month[1]. This represents a significant increase from the approximately 160 million monthly recommendation requests and 20 million users we had at the start of 2023 (Figure 4).

5. Conclusion and Future Work

Given the high proportion of users who visit Skyscanner to explore and plan new trips, recommending travel destinations is one of the platform's main offerings. Providing item recommendations has proven valuable across various fields, and the travel sector is no exception. This has led to a wide range of available state-of-the-art recommendation models that are ready to be trained and tested. However, the challenges that companies face when implementing recommender systems are often overlooked. These challenges include ensuring high internal adoption by different frontend teams, improving the developer experience during integration, dealing with the cold start problem, and interpreting traveller search intent and matching it with relevant recommendations from a diverse catalogue. In this paper,

[1]Average values across June & July 2024

we have discussed Skyscanner's journey of implementing a centralised destination recommendation system and how it has effectively addressed these challenges.

Future work will focus on further improvements to the recommendation models, exploring more advanced machine learning techniques, and expanding recommendations into new areas of the business. This work on the new architecture with centralised recommendations service has given us a template for potentially centralising architecture for other types of recommendations as well, such as e.g. hotel recommendations that have a very different stack of backend services and have their own set of challenges.

References

[1] Mckinsey, Company, How retailers can keep up with consumers, https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers, 2013. Accessed: 2024-05-30.

[2] Invesp, Using dynamic product recommendations in order to increase sales, https://www.invespcro.com/blog/using-dynamic-product-recommendations-in-order-to-increase-sales/, 2017. Accessed: 2024-05-30.

[3] Mckinsey, Company, The value of getting personalization right —or wrong— is multiplying, https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/the-value-of-getting-personalization-right-or-wrong-is-multiplying, 2021. Accessed: 2024-08-01.

[4] L. Bernardi, T. Mavridis, P. Estevez, 150 successful machine learning models: 6 lessons learned at booking. com, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 1743–1751.

[5] C. Ross, T. Ovadia, J. Mooney, A. Meitin, E. Kabilou, M. Kabalo, D. Goldenberg, Democratizing travel personalization via central recommendation platform., in: RecTour@ RecSys, 2022, pp. 92–97.

[6] A. Mottini, A. Lhéritier, R. Acuna-Agost, M. A. Zuluaga, Understanding customer choices to improve recommendations in the air travel industry., in: RecTour@ RecSys, 2018, pp. 28–32.

[7] K. Chaudhari, A. Thakkar, A comprehensive survey on travel recommender systems, Archives of Computational Methods in Engineering 27 (2020) 1545–1571.

[8] D. Goldenberg, K. Kofman, J. Albert, S. Mizrachi, A. Horowitz, I. Teinemaa, Personalization in practice: Methods and applications, in: Proceedings of the 14th ACM international conference on web search and data mining, 2021, pp. 1123–1126.