Decision Analytics Journal 10 (2024) 100414

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

# An investigation of the exposure effect of recommender systems in hospitality

Rik van Leeuwen [a,b,*], Koen Hoogkamp [b], Ger Koole [b]

[a] Ireckonu, Olympisch Stadion 43, 1076DE, Amsterdam, The Netherlands

[b] Department of Mathematics, Vrije Universiteit Amsterdam, De Boelelaan 1111, 1081HV Amsterdam, The Netherlands

ARTICLE INFO

*Keywords:*
Recommender systems
Machine learning
Exposure effect
Hospitality
Reinforcement learning

ABSTRACT

This study aims to investigate the role of a recommendation system (RS) in personalizing the guest journey in the hospitality industry and its impact on decision making, particularly the exposure effect. The study has two main objectives: (a) to quantify the additional value of using a RS based on an empirical dataset and (b) to explore how to leverage the exposure effect based on a simulated dataset to increase the number of services chosen by guests. The study compares machine learning models and naive strategies using several evaluation metrics to evaluate the added value. The study uses an empirical case study containing over 33,000 reservations to compare the effectiveness of different methods. We show that the eXtreme Gradient Boosting (XGBoost) model outperforms other approaches based on accuracy (0.905), AUC (0.908), and simulated revenue. The study also employs a reinforcement learning approach, as an extension of a RS, to quantify the exposure effect and leverage it to increase the number of services chosen by guests. The approach is found to improve the number of services chosen by 7.60% compared to a RS without this extension in a simulation environment.

## 1. Introduction

Recommender systems (RSs) suggest items to users to ultimately make decision-making easier [1,2]. The primary challenge is that the number of available items exceeds the number of slots or positions for showing recommended items to users. The suggestions are *personalized*, which is the ability to understand users individually [3]. The advantages of personalization are a satisfying user experience and an increase in revenue [4–6].

Since there is limited research on the accuracy and effect of a RS in hospitality, this will be the focus of this study. A benefit of a RS in the hospitality industry is three-fold, since it would (1) provide an enhanced guest experience because there is a wider variety to choose from [7,8], (2) allow the potential to sell additional services to increase revenue [7,9], and (3) help the industry stay competitive [10,11]. The importance of each benefit depends on the business objective; where the loyalty of guests is low, the focus is on maximizing the revenue during their stay. Where guest loyalty is paramount, the guest experience is critical since there is a direct link between guest experience and guest loyalty, which drives revenue in the long term [12].

The main objective of this study is two-fold. Firstly, this work approach is to compare statistical and machine learning techniques, with a data set of a hospitality company, to recommend services to users. To date, a thorough review of the literature has not yielded a single documented use case for the application of this technology in hospitality. This data set contains guest features, e.g., length of stay, room type, rate amount, and chosen services. This data set only contains choices and the recommendations are unknown. The overall and item-specific performance is discussed in detail. It provides a guideline on how the hospitality industry can implement a RS to ultimately increase revenue.

Secondly, this work will focus on the exposure effect, which is the impact of showing items during the decision-making process on users by exploring through the deliberate recommendation or omission of services. This requires the subsequent storage of recommendations and user choices. To incorporate this bias into the recommendation system, a simulation environment is utilized due to the lack of information regarding the recommended items in the empirical case study.

The main contribution of the proposed work can be summarized as follows:

- The novelty of this study lies within the application of hospitality, where the feasibility of utilizing guest features to make personalized recommendations is examined, with a primary emphasis on the accuracy of the RS.
- Furthermore, the objective to learn and quantify the exposure effect with a reinforcement learning approach, as an extension of a machine learning model, is a novel and innovative approach that aims to increase to number of chosen items per user.

The remainder of this paper is organized as follows: an overview of relevant background is given in Section 2, the empirical case study is

* Corresponding author at: Department of Mathematics, Vrije Universiteit Amsterdam, De Boelelaan 1111, 1081HV Amsterdam, The Netherlands.
*E-mail addresses:* rik@ireckonu.com (R. van Leeuwen), koenfhoogkamp@gmail.com (K. Hoogkamp), ger.koole@vu.nl (G. Koole).

https://doi.org/10.1016/j.dajour.2024.100414
Received 21 September 2023; Received in revised form 23 January 2024; Accepted 3 February 2024
Available online 6 February 2024

described in Section 3, the effect of the RS is outlined in Section 4, the theoretical and practical implications are given in Section 5, and the conclusion are presented in Section 6.

## 2. Background

At first, the research on RSs primarily focused on the accuracy of recommendations and chosen items, also known as recommendation precision [13]. Several industries have successfully implemented RSs in terms of accurate recommendations. For instance, 80% of movies watched on Netflix originate from recommendations given by recommender system [4]. In addition, between 20% and 40% of Amazon's sales come from recommendations of products that are not part of the 100,000 most popular items [5].

The standard evaluation methods of RSs, such as accuracy and precision, focus on user' choices based on recommendations instead of user' interests. These choices are the items chosen by the users when exposed to the system's recommendations. There is a distinction between recommendations and choices because not all recommendations are chosen, and not all choices are recommended. For example, users can still choose items which are not part of the recommendation because they are interested in those items. A RS influences the user' interests with recommendations, and the effects of recommendations originating from a RS gain additional attention [13]. However, there have been fewer studies on understanding RS biases and the effect of those biases. One of the most studied biases is the popularity bias: recommendations include popular items repeatedly, while less popular are recommended rarely or not at all [14]. It is difficult to isolate the effects of the RS itself, and to understand the interaction between unobserved user interest in an item and the RS.

The studies dedicated to RSs in hospitality have focused mainly on recommender precision and not the effects. The applied RSs are for destination management, where a model is trained on users' behaviour to recommend hotels to others. This approach, one of the three main approaches, is known as *collaborative filtering* [15]. It filters items that a user might choose based on reactions by similar users. The other two approaches are *content-based filtering* and *hybrid recommendation approach* [16], which combine collaborative and content-based filtering. The content-based approach takes the description or keywords of items of historical choices to recommend similar items. Typically, hotels offer a limited but diverse number of services without any relation between services. Therefore, a collaborative filtering technique suits the research objective to enhance a stay by recommending extra services based on guest similarity.

Collaborative filtering is divided into two main groups [17]. *Model-based* estimates or learns a model based on historical data and then applies this model for recommendations of items to users; *memory-based* establishes correlations or similarities between users or items by accessing the historical data directly. The computational time of a memory-based approach is higher than a model-based approach due to an increasing amount of data. However, the memory-based approach is adaptive to changes since it directly accesses the data. A model-based approach is consistent with the computational time but not adaptive to data changes. In addition, a model-based approach outperforms a memory-based approach within a software engineering setting [18] and an e-commerce setting [19].

Regardless of the approach, the contribution of a RS is two-fold: (1) the service provider can achieve its objective, and (2) users tend to make decisions with greater ease [1]. Objectives of the service providers can differ from one to another [20]. The most common goals are to increase the number of items sold, sell more diverse items, increase user satisfaction, increase user fidelity, and obtain insights into what the user wants. Ultimately, a RS balances the objectives of the service provider and user to offer an implementation that works both ways.

A constant evaluation of the performance is a requirement to guarantee the contribution of a RS [21]. Evaluation metrics are dependent on the filtering technique. Several metrics exist for calculating the performance on an overall level and an individual item level. By taking items individually, the hotel chain can decide to change the available services if the service is not chosen by guests. The evaluation metrics translate a business strategy into performance metrics [22].

Underlying effects or biases of RSs influence the evaluation metrics [23], and it is difficult to isolate these hidden effects or biases. Two main approaches, online and offline, are applied to analyse the behaviour of users' choices when users are recommended items by a RS [13]. The online approach uses an online platform, and the offline approach uses a simulation to obtain results. A variety of scenarios can be tested with an offline approach, while with an online approach, a specific scenario is tested. A drawback of the offline approach is that the choice behaviour is simpler due to assumptions in the simulation framework. Therefore, the results of an online experiment are probably more reliable than an offline experiment due to the complexity of modelling and partial reconstruction of user choices [13]. Regardless of the approach to understanding effects, the growing attention confirms interest from a research perspective.

The popularity bias is not as relevant given a small set of available products because, for example, a web page provides an overview of all products. However, showing fewer items is advantageous to the conversion process due to the time reduction of the process by avoiding the paradox of choice [24,25]. Showing an item during the decision process influences the users. The effect of merely showing items is known as the exposure effect [26].

The effect of interest in this study is the exposure effect, which relates to the item awareness to users. The awareness set for a specific user is defined as the user's knowledge of the available items before recommendations are provided [13]. Certain items have a higher probability to be chosen because these are known to the user before seeing any items. Nonetheless, users can choose items without them being initially recommended. The exposure effect is low when recommendations influence the decision, but minimal to none when the probability of choosing does not change significantly. The exposure effect is high if it influences the user's choice because the item was not included in the awareness set. For example, the location of a hotel or season impacts the available services and therefore influences the user-specific awareness set. An opportunity arises to leverage this effect to increase the performance of a RS.

The process of collecting data is essential to encountering or leveraging an effect of the RS. User interactions generate data, and therefore promoting items influences users' choices. This interaction between the user and RS is a closed feedback loop since historical interactions are the backbone of updating or retraining the mathematical model [27]. Promoting random items affects this feedback loop. These random recommendations contribute to a diversified data set, which enables the RS to learn undiscovered user behaviour.

An additional selection policy deals with effects and random recommendations, also known as a post-learning technique [28]. This technique varies the selection of items after obtaining the predictions given by a mathematical model. An exploration component of the policy is essential to affect the feedback loop. By incorporating random recommendations, the data set is diversified, which leads to an increase in performance [29].

## 3. Empirical case study

This section outlines the hospitality case study and its results. The hospitality case is a supervised machine learning task, with performance evaluated retrospectively for a comprehensive assessment of the recommendation system's effectiveness. The results presented provide insights into both the overarching system performance and service-specific performance metrics. These findings, in turn, serve as crucial input for the simulation framework, facilitating a deeper understanding and quantification of the exposure effect, as detailed in Section 4.

**Table 1**
Services overview.

| Service | Abbreviation | Description | Number of reservations |
|---|---|---|---|
| Amenities | AMN | Hotel franchise products | 596 |
| Breakfast | BF | Standard breakfast buffet | 12,387 |
| Food & Beverage Credit - Low | CRD-L | Credits for food & beverage <100 dollars | 7239 |
| Food & Beverage Credit - High | CRD-H | Credits for food & beverage ≥100 dollars | 9486 |
| Fitness | FIT | Access to fitness facilities | 1634 |
| Internet | INT | Premium internet | 1897 |
| Kids Club | KID | Service that takes care of kids | 2377 |
| Parking | PRK | Parking at the hotel | 2863 |

**Table 2**
Services per booking channel.

| Booking channel | AMN | BF | CRD-L | CRD-H | FIT | INT | KID | PRK | Total services |
|---|---|---|---|---|---|---|---|---|---|
| OBS | 0.77% | 35.05% | 18.61% | 24.86% | 3.34% | 4.09% | 5.81% | 7.47% | 24,711 |
| OTA | 0.32% | 90.53% | 1.27% | 3.81% | 0.01% | 0.01% | 4.05% | 0.00% | 12,308 |
| VS | 2.13% | 20.02% | 17.05% | 25.51% | 9.19% | 10.07% | 4.55% | 11.49% | 8225 |
| HOUSE | 0.50% | 36.61% | 28.64% | 17.50% | 0.83% | 0.97% | 13.39% | 1.56% | 4218 |
| VIP | 15.85% | 40.30% | 2.26% | 38.26% | 1.28% | 1.28% | 0.23% | 0.53% | 1325 |

### 3.1. Dataset

The dataset utilized in this section is contingent upon the integration of an RS at specific guest touchpoints. These touchpoints within a guest's journey significantly influence the RS's functionality based on the data and services accessible. Only with the pre- and mid-stay touchpoints [30] can additional services be recommended to enhance a stay. Examples of pre- and mid-stay touchpoints are the booking process, pre-arrival emails, kiosk check-ins, and mid-stay interactions via an app. Each touchpoint can have a different number of placeholders available to show recommendations. Therefore, the performance per touchpoint can differ due to the available data and the number of placeholders.

The data set originates from the Property Management System (PMS) of a single hotel property located in the United States of America. A PMS registers and regulates reservations in the hospitality industry. The date range of the data set is from the 1st of January 2018 until the 31st of December 2021. Personal Identifiable Information (PII) is excluded due to the General Data Protection Regulation (GDPR).[1]

There are 293,628 reservations available, where 33,521 reservations include one or more additional services. The available attributes of a reservation during the booking process is a subset of all attributes available within the PMS. The availability is dependent on the requirements of the hotel chain's booking process. The following attributes are available for this case study: booking date, arrival date, departure date, room type, room rate, number of persons, number of children, guest ID, part of a group booking, and part of a company booking. Additional attributes, such as the length of stay and the average rate per room night, are generated based on existing ones, known as feature engineering. Table A.1 in Appendix A provides an overview of the 41 available features during the booking process.

After a reservation creation, the additional services are also registered in the PMS. The relationship between the reservations and services is 1-to-many since a stay can include multiple services, e.g., breakfast and fitness. The hotel bundled services to eight different categories (see Table 1). These additional services are dependent on specific hotel facilities. The Food and Beverage Credit are service categories that a guest can spend over their full stay. There is a disparity in the number of services sold. For example, breakfast is sold 20 times more than amenities. The fraction of reservations that have one, two, or three services sold are respectively 0.6, 0.25, and 0.15.

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), [2016] OJ L 119/1.

The data is generated without having a RS in place, therefore certain services are sold differently. For example, fitness and premium internet could be sold via a deal package or via the front desk. This brings into question the validity of implicit decision-making of the guests: did hotel guests purchase these services because they wanted to purchase them, or did they buy them because it was offered in a certain way (e.g., as a package or as a complementary deal)?

### 3.2. Exploratory data analysis

There are a total of 41 guest features created based on the available data. The categorical and numerical data types are analysed, and this section contains the main characteristics and findings.

The booking channel influences the set of available services (see Table 2). For example, the majority of reservations created via the booking channel Online Travel Agencies (OTA) have breakfast as an extra service (90.53%). Usually, this is the only available option for a guest during the booking process. Hotels deliberately provide specific options due to commission for the OTA's for reservations booked via their platform. Data originating from an OTA booking channel is excluded because of (1) the limited variety of services, and (2) influence to dictate content on such a platform. The number of available reservations is reduced to 16,365, which includes additional services.

There are a total of 35 categorical features available, these categorical features are one hot encoded to ensure a binary nature. The Cramér's statistic $V_c$ [31] calculates a correlation, in this case between categorical guest features and services purchased. The outcome, $V_c$, has a value between 0 and 1, where 0 means zero association and 1 means full association. The calculated $V_c$ values of each categorical feature with the corresponding service are shown in Table C.1 in Appendix C. The majority of features have a weak association, between 0.01 and 0.1 [32]. However, a combination of features might contain predictable power. The features with the highest $V_c$ are the binary values for company and unit type.

The services bought are also one-hot encoded into binary features. Fig. 1 shows the number of services bought by month, where the data is normalized per service. Parking spots are popular during the summer months and less popular in November and December. Most amenities are sold in December and fewer are sold in the summer months. The breakfast service, as well as and both credit services, are chosen consistently throughout the year.

Fig. B.1 in Appendix B visualizes the six numerical features in box plots, where outliers are excluded. The box plots show a difference in mean and variance between services. For example, Fig. B.1(e) shows that the average daily rate for fitness is greater than that of credit. Fig. B.1(d) and Fig. B.1(c) show the distribution of the number of

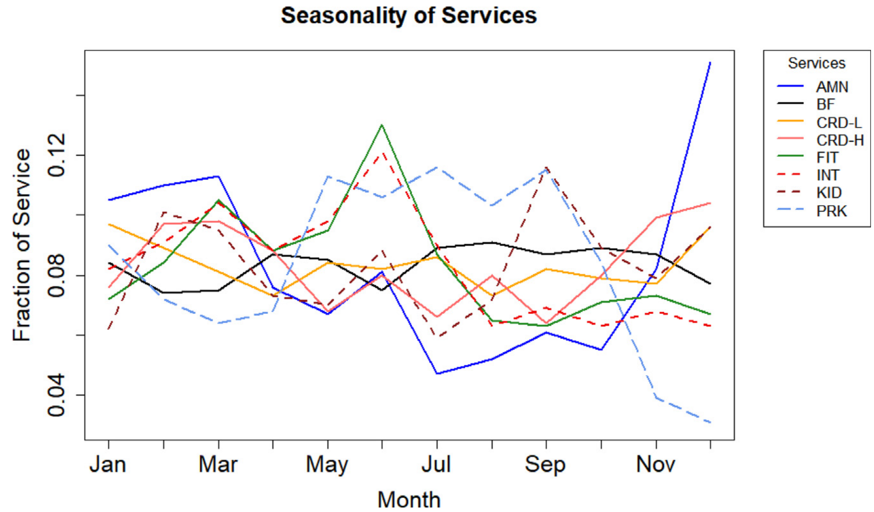



**Fig. 1.** Monthly normalized frequency of services bought.

persons and number of children per service purchased. On average, there are two persons per reservation, with a difference in the amenities and parking. The number of children attending kids club might not be recorded accurately since the mean is 0.

**Table 3**
Evaluation metrics.

| Name | Top-N | Per service |
|---|---|---|
| Accuracy | Yes | Yes |
| AUC | No | Yes |
| Revenue | Yes | No |

### 3.3. *Methodology*

A brief overview is provided of the models and evaluation techniques used during the empirical use case. The applied techniques are not novel and therefore the focus is on providing a guide to implementing a RS. The model selection process prioritizes practical implementation considerations rather than providing an extensive overview of all available techniques. This distinction underscores the focus on guiding practitioners in the hospitality industry towards effective RS implementation.

It is worth noting that the task is set up as a multi-label classification problem. There are five models compared: two intuitive benchmark strategies and three statistical/machine learning models, which are used as model-based collaborative filtering techniques. Every model takes the guest features $X$ as input. The output is an ordered list of items as output, where the top-$N$ items are presented to the user.

#### 3.3.1. *Models*

Two naive strategies benchmark the statistical and machine learning techniques. A benchmark provides the ability to compare advanced techniques to naive strategies. These naive strategies are common practice for touchpoints within the hospitality industry due to the challenges of slow implementation [33]. The first baseline is *Random Rank*, which offers a top-$N$ list of items randomly to each user. The second baseline is *Most Popular*, which offers a top-$N$ list of items based on the most frequently sold items.

Multi-label Logistic Regression [34] is applied to recommend the top-$N$ list of items to a user. The logistic regression calculates a probability for each item, and by sorting these probabilities in descending order, the top-$N$ items are presented to the user. Multi-label Logistic Regression is implemented with the *sklearn* package [35] in Python.

An eXtreme Gradient Boosting (XGBoost) classifier [36] is used, with a binary logistic objective, to recommend the top-$N$ list of items to a user. XGBoost uses gradient boosting to calculate class probabilities. This entails the creation of multiple weak decision tree predictors and combining them into a single model. These weak decision trees are created using a gradient descent algorithm by minimizing an objective function. The XGBoost-classifier is implemented with the *xgboost* package [36] in Python.

A Dynamically Expandable Network (DEN) is a neural network with a dynamically expandable component. The network is capable of increasing or decreasing nodes while maintaining the previously learned weights [37], where nodes represent the items. A DEN is designed with the purpose of incremental learning; the model is capable of continuous learning on incoming data when the number of items is changing. DEN is implemented with the *PyTorch* package [38] in Python.

A challenge within hospitality is the change in item catalogue due to the season or guest needs. Unlike DEN, Multi-label Logistic Regression and XGBoost-classifier are unable to adapt to a change in item catalogue once the model is trained. To overcome this challenge, an exploration technique, such as the epsilon-greedy strategy, can explore newly introduced items. This strategy balances exploration and exploitation by choosing between exploration and exploitation randomly, with a $\epsilon$ probability. After a number of instances, the model can be retrained including the new item to increase overall performance.

#### 3.3.2. *Evaluation metrics*

The assessment metrics used for the recommender system have been tailored to align with a multi-label classification model. Three primary metrics, namely accuracy, AUC (Area Under the Curve), and revenue, are utilized to gauge the system's performance (refer to Table 3). The top-$N$ class evaluates the RS when there are fewer placeholders or slots present than items. The *Per Service* calculates the performance for each service separately. In essence, recommending a service is a binary classification problem where the top-$N$ highest probabilities from a model are the binary decisions. A data point is classified in one of two buckets: correct or incorrect. Given the recommended services and chosen services, there are four combinations of classifications: true positives TP (correct positives), true negatives TN (correct negatives), false positives FP (incorrect positives), and false negatives FN (incorrect negatives) [39,40]. These four classifications form the basis of the confusion matrix, which is utilized to assess the system's performance in service recommendations within the context of multi-label classification.

Accuracy($y_i$) defines the accuracy of service $i$ (see Eq. (1)). This metric indicates the fraction of guests that choose a service $i$, and the

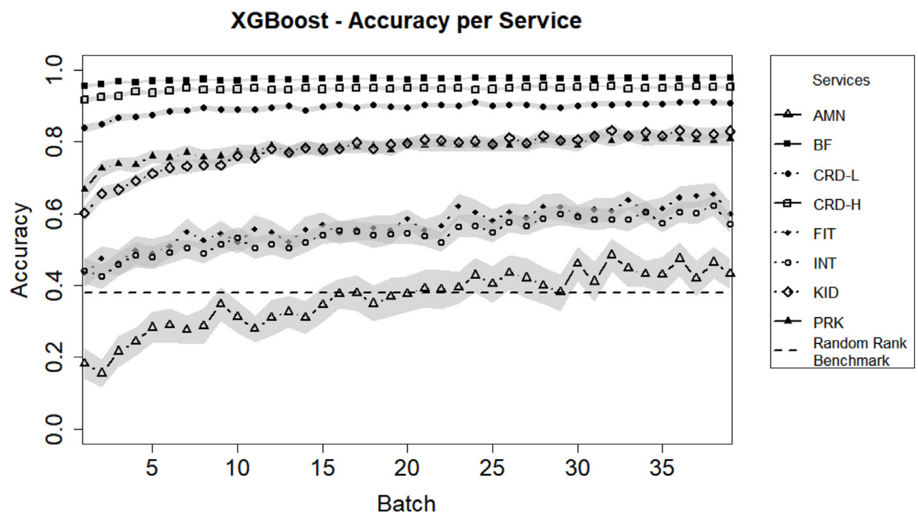



**Fig. 2.** XGBoost - accuracy per service.

**Table 4**
Revenue per service.

| Service | Revenue |
|---|---|
| AMN | 500 |
| BF | 25 |
| CRD-L | 50 |
| CRD-H | 125 |
| FIT | 40 |
| INT | 30 |
| KID | 120 |
| PRK | 40 |

fact that the service was presented via a recommendation.

$$y_i = \frac{\mathrm{TP}(y_i)}{\mathrm{TP}(y_i) + \mathrm{FN}(y_i)} \tag{1}$$

The area under the curve of receiver operating characteristics (AUC-roc or AUC) indicates the predictability of a model. AUC is favoured over precision for evaluating a recommender engine because it balances the trade-off between recall and precision, ensuring that relevant items are not missed while allowing flexibility in recommendations. It is top-$N$ independent and analyses the probability outcomes on true positive and false positive rates. The AUC is created by evaluating multiple model outputs to see how the number of true positives and false positives occurs for different thresholds. The AUC score is a number between 0 and 1, where a higher score indicates higher predictability of a model [41].

Total revenue is the sum of true positive recommendations. Table 4 shows the corresponding revenue of a service. If the overall business objective is to increase revenue, an estimation of revenue is preferred over the accuracy or the predictability of the model.

### 3.4. Results

To obtain the results of the case study, the booking process of guests is simulated with the empirical data set. All the services described in Table 1 are available during this booking process, and no exploration technique was applied. There are 3 placeholders active during this simulation, which sets the top-3 for the evaluation. The comparison between machine learning (ML) models and strategies are obtained when guests can only choose from the top 3 recommendations. If a chosen service from the empirical data set is part of the recommendation, the service is chosen. Since the interest in services is unknown, only services from the empirical data set, which are part of the recommendation, can be chosen.

A rolling window technique simulates the behaviour of a RS in the hospitality industry. Such a technique divides the data into $L$ batches with $p$ reservations per batch. The ML models are trained on a fixed number of batches to make recommendations for the next batch, defined as the evaluation batch. In this way. the ML models move sequentially through the data this way. For the simulation of the empirical case study data set, there are 40 batches with 500 reservations per batch. The shuffled data obtains stable results by eliminating seasonality. The simulation runs 50 times, and the average is taken from the final evaluation batch. Additionally, a Shapiro–Wilk test is executed to assess the distribution of the data equally between the batches. The combination of three evaluation metrics, as described in Section 3.3.2, formulates the performance of the strategies and ML models.

Table 5 shows the average accuracy taken from the last evaluation batch of the simulation, where the highest evaluations are in bold. The results of the last batch provides a clear snapshot of the final performance of each model. XGBoost scores the highest weighted accuracy, defined as the accuracy by the frequency of the services. However, XGBoost does not outperform all other models on an individual service level. Logistic regression outperforms XGBoost for breakfast and parking. For example, fitness is a service where XGBoost clearly outperforms all the models. The Random Rank strategy stands out in accuracy for the service amenities.

Fig. 2 shows how the accuracy per model improves when additional data becomes available for training purposes. This visualization depicts trends over batches, providing insights into the evolving performance throughout the simulation. The accuracy per service is plotted with a 95% confidence interval obtained from the 50 simulations. Each service has an accuracy above the weighted Random Rank threshold, shown in the figure as the horizontal black striped line without confidence interval.

Table 6 shows the average AUC taken from the last batch of the simulation, where the highest evaluations are marked in bold. XGBoost outperformed the other models in terms of the weighted and service-specific AUC. The baselines have zero predictability because the AUC is equal to 0.5. XGBoost has a weighted AUC score of 0.908, which translates to high predictability of the trained model [42].

Table 7 shows the obtained revenue for each model. As for the accuracy and AUC, XGBoost outperforms the other models by obtaining 0.856 of the maximum obtainable revenue. In addition, XGBoost obtains 22.4% more revenue than the most popular strategy and 120.4% more revenue than the Random Rank strategy.

**Table 5**
Accuracy - Final evaluation batch.

| Accuracy | AMN | BF | CRD-L | CRD-H | FIT | INT | KID | PRK | Weighted |
|---|---|---|---|---|---|---|---|---|---|
| Most Popular | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.793 |
| Random Rank | **0.438** | 0.377 | 0.386 | 0.377 | 0.382 | 0.369 | 0.375 | 0.382 | 0.380 |
| Logistic Regression | 0.164 | **0.993** | 0.803 | 0.948 | 0.496 | 0.450 | 0.651 | **0.847** | 0.868 |
| DEN | 0.384 | 0.976 | 0.865 | 0.938 | 0.431 | 0.446 | 0.740 | 0.771 | 0.874 |
| XGBoost | 0.432 | 0.978 | **0.907** | **0.952** | **0.600** | **0.572** | **0.830** | 0.809 | **0.905** |

**Table 6**
AUC - Final evaluation batch.

| AUC | AMN | BF | CRD-L | CRD-H | FIT | INT | KID | PRK | Weighted |
|---|---|---|---|---|---|---|---|---|---|
| Most Popular | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| Random Rank | 0.517 | 0.500 | 0.507 | 0.501 | 0.508 | 0.494 | 0.501 | 0.503 | 0.502 |
| Logistic Regression | 0.766 | 0.803 | 0.760 | 0.894 | 0.942 | 0.918 | 0.777 | 0.892 | 0.834 |
| DEN | 0.803 | 0.840 | 0.810 | 0.901 | 0.921 | 0.898 | 0.829 | 0.875 | 0.857 |
| XGBoost | **0.836** | **0.890** | **0.886** | **0.932** | **0.957** | **0.942** | **0.913** | **0.922** | **0.908** |

**Table 7**
Revenue - Final evaluation batch.

| Models | Revenue | Fraction of always right |
|---|---|---|
| Always Right | 46,361 | 1.0 |
| Most Popular | 32,399 | 0.699 |
| Random Rank | 17,999 | 0.388 |
| Logistic Regression | 35,384 | 0.763 |
| DEN | 37,760 | 0.814 |
| XGBoost | 39,662 | 0.856 |

## 4. Exposure effect

This section explains how to deal with an exposure effect within a RS. The experimental setup provides a description of the model, experimental data, simulation framework, methodology, and results. The results of the simulation environment show a comparison between a trained algorithm based on the best performing algorithm of the empirical case study (see Section 3) and the novel methodology based on reinforcement learning. The objective is to maximize the total number of services chosen by guests.

### 4.1. Model

A RS impacts guests decision-making by recommending some services and not recommending others. This impact is incorporated via the exposure effect, which relates to the service awareness of guests. This awareness of services influences the guests' interest in the available services before recommendations are provided [13]. Within this model, guests can choose services without them being initially recommended. Notably, unlike the results in Section 3.4, guests have the ability to choose services which are not part of the recommendation.

An assumption is made that the awareness is a factor, denoted as $\rho \in [0,1]$, which is service-specific and not guest-specific. Section 6 describes how this assumption can be relaxed. Services with a high awareness factor are more likely to be chosen if the guest is interested in the service and it is not recommended. If the awareness factor is equal to 0, the service will never be chosen if it is not recommended. If the awareness factor is equal to 1, it does not make a difference if the service is recommended or not. This implies that the same factor is applied for a guest who is not interested compared to a guest who is interested in the services when it is not included in the recommendation.

The RS implementation is modelled as a part of the booking process within the guest journey. A guest wants to book a hotel room and enters the booking process, e.g., at the website of the hotel. For example, the guest may see the additional services after selecting their stay dates and room type. This implies that there is a feature set available to the guest before services are recommended. The process of generating data for recommending services to guests is described in Section 4.2. The set of available services the guest can choose from is denoted as $U = \{u_1, \dots, u_K\}$. Within this model, the total number of available services is $K = 12$. The set of services $U$ is divided into the recommended set of services, denoted as $U_1$ where $|U_1| = N$, and the set of services that is not recommended, denoted as $U_0$. The guest sees $N = 3$ recommendations and has the ability to view other services, and therefore the possibility to choose from all services.

A guest choosing service $u_i$, denoted as $c_i \in \{0,1\}$, is modelled as a Bernoulli random variable $y_i$, with probability $\theta_i$ choosing the service $u_i$ and probability $1-\theta_i$ not choosing the service $u_i$. The set of chosen services is defined as $C = (c_1, \dots, c_K)$.

The decision-making of a guest is dependent on the interest in a service and whether it is recommended or not recommended. If a service $u_i \in U_1$, the probability of choosing is equal to the interest $y_i \in [0,1]$. If a service $u_i \in U_0$, the probability of choosing is equal to the interest $y_i \cdot \rho_i$, where $\rho_i$ represents the awareness factor of service $u_i$. The set of awareness factors is defined as $\boldsymbol{\rho} = (\rho_1, \dots, \rho_K)$.

### 4.2. Guest data simulation

As outlined in Section 3, guest features play a pivotal role in shaping personalized recommendations. To evaluate the system's performance and simulate the potential gain of the exposure effect, new data points are generated. These data points represent individual guests, denoted as $g$, engaged in booking processes, encompassing guest-specific features and service interests. This combination allows for the simulation of decisions that would be impractical or impossible to derive solely from the empirical dataset.

The interest in service $u_i$, the dependent variable $y_i$, is based on a set of features, the independent variables $X$. The dependent variable is modelled as a linear model. The set of independent variables is denoted as $X$, where each independent variable $x$ follows a normal distribution with mean 0 and standard deviation 1. An error $\epsilon$ is added as noise with mean 0 and standard deviation 1. Eq. (2) represents how the interest in service $u_i$ is compiled. A sigmoid function is applied to scale the value between 0 and 1. For simplicity, each service interest $y_i$ depends on a single independent variable $x_i$. The underlying dependencies between features and services can easily be extended with this setup.

$$y_i = \frac{1}{1 + e^{-(x_i+\epsilon)}} \tag{2}$$

The set of $K$ interests in the services for a single guest is denoted as $Y = y_1, \dots, y_K$. The combination of $X$ and $Y$ represents a single guest.

### 4.3. Simulation framework

In the empirical case study, XGBoost outperformed the other approaches (Section 3). Therefore, the base recommender in this simulation framework is a XGBoost model. The outcome of the base

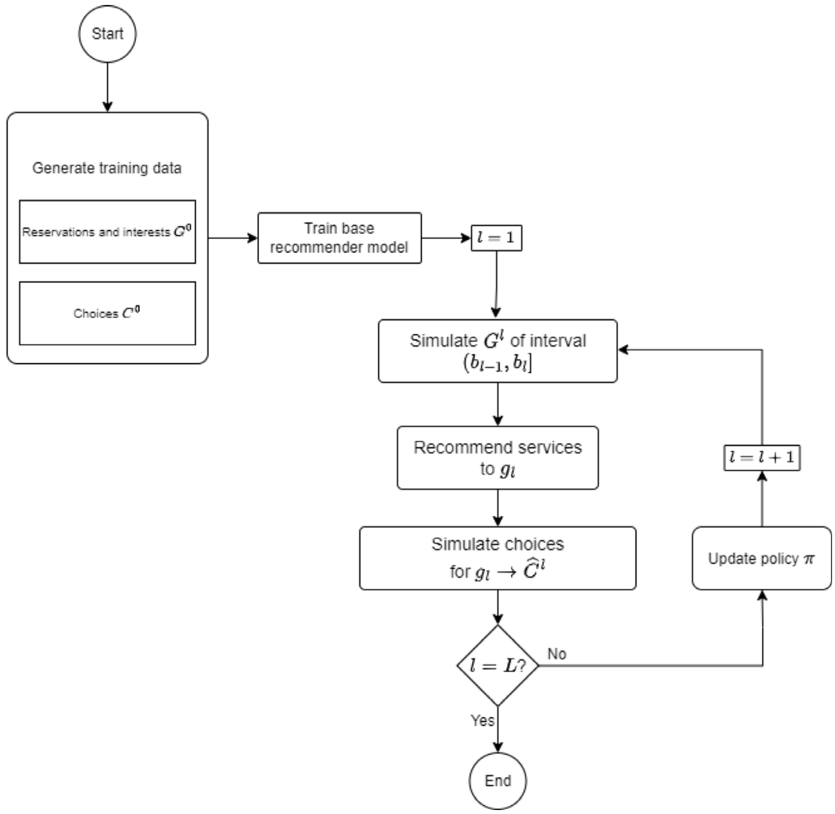



**Fig. 3.** Simulation environment.

**Table 8**
Awareness factor per service.

| | Service | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| $\rho$ | 0.78 | 0.38 | 0.13 | 0.06 | 0.62 | 0.57 | 0.29 | 0.09 | 0.15 | 0.22 | 0.42 | 0.10 |

recommender, defined as $\hat{Y}$, is used as input for RL policy to calculate the expected reward. This XGBoost model is trained on 100,000 simulated guests $g$, consisting of reservations, defined as $G^0$, and corresponding interest. Random recommendations are given to these guests and services are chosen by guests, defined as $P^0$, according to the corresponding interest. The base recommender is trained on $G^0$ and $P^0$. The randomness in recommendations ensures that the base recommender has the ability to learn the relationships between the independent and dependent variables from the simulated data. The trained base recommender model is stored and used in each simulation use case to ensure that the base is a stable component in the simulation framework.

To mimic the empirical case study, the recommendations are not saved for the initial $G^0$ reservations. Since the recommendations are not saved, the exposure effect needs to be learned over time through RL policy. For each arriving guest in the booking process, the base recommender provides $\hat{Y}$ based on a set of features $X$, which results in the estimated interest for each service $u_i$.

As mentioned in Section 4.1, guest decision-making is impacted via a service-specific awareness factor. Table 8 shows the awareness factor per service used in the simulation. A higher awareness factor represents a service which is widely known, such as breakfast or late check-out.

Fig. 3 summarizes the simulation framework, as described in this section. The simulation is executed in batches, where batch $b_l$ denotes the $l$th batch containing $g$ guests. This is a common approach in practice because reinforcement learning policies cannot usually be updated after each instance due to limitation with resources, e.g., computational power or running costs. To understand the stability within the results, the simulation framework is simulated for 50 use cases, each containing $l = 20$ batches of 500 guests. Table 9 provides a summary of symbols and notations.

### 4.4. Reinforcement learning policy

A reinforcement learning (RL) policy is dedicated to training an agent to operate in an environment to maximize the agent's utility in the pursuit of a goal [43]. In this setting, a RL policy, as part of a RS, is incorporated into a touchpoint within the guest journey, such as the booking process or kiosk check-in. During a touchpoint, guests are provided the option to choose additional services to enhance their hotel stay. Usually, the offer of additional services is part of a flow. We assume that at most a single guest arrives during a time step $t$, where $t = 1, \ldots, T$. The goal is to maximize the number of services being chosen per guest.

The policy, denoted as $\pi$, dictates the action $a_t$ the agent takes based on a given state $s_t$ at time $t$. Within this set up, it is the services that are recommended from a set of services $U$ to a guest. The action space is discrete and denoted as $\mathcal{A} = \{a_1, \ldots, a_j\}$, where $j = \binom{K}{N}$ where $K$ represents the number of services and $N$ the number of recommendations. Each action $a \in \mathcal{A}$ is a binary vector with length $K$ and the sum equal to $N$. The $i$th index of action $a$, denoted as $a^i \in \{0, 1\}$, represents if a service is not recommended, denoted as 0, or if a service is recommended, denoted as 1.

The state $s$ is the information an agent needs to make a decision. The information is an estimated interest, denoted as $\hat{y}_i$, of a guest in a service $u_i$. The interest within a service is unknown, and therefore an estimate is provided by a recommender model. The recommender model

**Table 9**
Summary of symbols and notations.

| $K$ | Number of services |
|---|---|
| $N$ | Number of recommendations, where N < K |
| $G$ | Full set of reservations and corresponding interests of guests |
| $U$ | Full set of services, where $U = \{u_1, \dots, u_K\}$, and $U = U_1 \cup U_0$ |
| $U_1$ | A set of recommended services, where $U_1 \subset U$, $U_1 \cap U_0 = \emptyset$, and $\lvert U_1 \rvert = N$ |
| $U_0$ | A set of services that are not recommended, where $U_0 \subset U$, $U_1 \cap U_0 = \emptyset$, and $\lvert U_2 \rvert = K - N$ |
| $\rho$ | Full set of awareness factors, where $\rho = (\rho_1, \dots, \rho_K)$ |
| $L$ | Number of the simulated time intervals |
| $b_0$ | Starting point of the simulation |
| $b_l$ | End point of the $l$th batch |
| $g_l$ | List of the guests who make choices in $b_l$ |
| $(b_{l-1}, b_l]$ | Interval between $b_{l-1}$ (excluded), $b_l$ (included) |
| $G^0$ | Matrix of the reservations and corresponding interests in the data until $b_0$ |
| $C$ | Binary $\lvert G \rvert \times \lvert K \rvert$ matrix of the observed choices in the data |
| $C^0$ | Matrix of the observed choices in the data until $b_0$ |
| $\hat{C}^l$ | Matrix of the simulated choices in $(b_{l-1}, b_l]'$ |

is referred to as a base recommender, which does not takes into account the exposure effect in predicting the interest. This recommender model requires features $X$ as input, which represent guests and are known or gathered during the touchpoint. The evaluation originating from the base recommender provides an estimate of interest in a service $u_i$, denoted as $\hat{y}_i$.

Given the state $s$, the policy determines the action, defined as $\pi(s) \rightarrow a$. In this set up, the reward $r_t$ is immediate since the guest finishes the flow of the touchpoint at time $t$. The agent observes the reward for the selected action, and nothing else. The reward is defined as the sum of the chosen services at time $t$, denoted as $C_t$ (see Eq. (3)). The probability of a guest choosing a service depends on the interest in a service and if a service is part of the recommend set of services; therefore, the reward is dependent on the action.

$$r_t = \sum_{i=1}^{K} c_i \tag{3}$$

Per action, an expected reward is calculated based on the predicted interest in each service for the guest given a set of features $X$, and the estimated awareness factors. The agent chooses the action, representing the services that are recommended and not recommended, with the highest expected reward. An $\epsilon$-greedy approach is added to the policy to trade off exploration and exploitation. The aim of this addition is to avoid ending up in a local optimal solution. The $\epsilon$-greedy approach selects the greedy action with a fixed probability 1-$\epsilon$, $0 < \epsilon < 1$ and takes a random action with probability $\epsilon$.

The expected reward of an action, denoted as $\hat{\theta}(\hat{Y}, \hat{\rho})$ , is defined as a function of the estimated interest in services $\hat{Y}$ and the estimated awareness factors $\hat{\rho}$. The expected reward is the sum of expected rewards per service, defined as $\hat{\theta}(\hat{Y}, \hat{\rho}) = \sum_{i=1}^{K} \hat{\theta}_i(\hat{y}_i, \hat{\rho}_i)$. The expected reward for service $u_i$ differs for recommending or not recommending the service due to the exposure effect. The expected reward is separated and constructed as a linear model (see Eq. (4)).

$$\hat{\theta}_i(\hat{y}_i, \hat{\rho}_i) = \begin{cases} \hat{\theta}_i^1(\hat{y}_i, \hat{\rho}_i) = \hat{y}_i & \text{if } u_i \in U_1 \\ \hat{\theta}_i^0(\hat{y}_i, \hat{\rho}_i) = \hat{\rho}_i \cdot \hat{y}_i & \text{if } u_i \in U_0 \end{cases} \tag{4}$$

An assumption is made that showing the service has an equal or higher expected reward than not showing the service, given the same interest. This results in a constraint where $\hat{\theta}_i^1(\hat{y}_i, \hat{\rho}_i) \geq \hat{\theta}_i^0(\hat{y}_i, \hat{\rho}_i)$.

The initialization of $\hat{\rho}_i$ is set to 0, which creates a policy which recommends the services with the highest interest. To update the estimated awareness factor per service, new instances need to be available where guests make a decision based on recommendations. The estimated awareness factors $\hat{\rho}$ in the expected reward $\hat{\theta}(\hat{Y}, \hat{\rho})$ are fitted given the available combination of three data points: (1) if a service was or was not recommended, (2) the interest in a service, and (3) if the service was chosen. The link between these three data points is important, otherwise the policy $\pi$ cannot be updated. The update of the policy $\pi$ can be triggered after a single instance or a batch of instances.

**Table 10**
Comparison of evaluation metrics based on 50 simulations.

| | Accuracy | Weighted AUC |
|---|---|---|
| Base Recommender | **0.867** | **0.788** |
| RL policy | 0.745 | 0.704 |

The agent learns by estimating an action-value function, denoted as $Q(s, a)$. The action-value function $Q_t(s, a)$ represents the expected reward for taking an action $a$ in state $s$ at time $t$ (see Eq. (5)). The action $a$, estimated interests $\hat{Y}$, and guest decision are required to update the estimated awareness factors $\hat{\rho}$ of the expected reward. When $Q^*$ is learned, the optimal policy $\pi^*$ can be constructed by selecting the action with the highest expected reward for each state, i.e., $\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a)$

$$Q_t(s, a) = \hat{\theta}(\hat{Y}, \hat{\rho}) \tag{5}$$

### 4.5. Results

This section discusses the results of the exposure effect in the simulation framework by providing a comparison between a base recommender model and a RL policy, introduced in Section 4.4. The objective is to understand if there is a possibility to leverage the exposure effect to maximize the number of services chosen by guests.

The accuracy and weighted AUC (see Section 3.3.2) are the selected metrics to evaluate the RS at a first glance this evaluation metrics are standard evaluation. The accuracy of the base recommender is 0.862, which is similar to results for the empirical case study. The weighted AUC is 0.792, which is lower than the results for the empirical case study. The evaluation of the base recommender was obtained from 10,000 out-of-sample simulated guests to ensure the predictability of the trained model.

Table 10 provides an overview of the standard evaluation metrics based on 50 simulations. The base recommender outperforms the RL policy in terms of accuracy and weighted AUC. The base recommender does not contain an exploration strategy; it will always decide to recommend the top-N services with the highest estimated interest. The reason why the RL policy scores lower on the standard evaluation metrics is because this strategy deliberately does not recommend services when guests are highly interested in these when the awareness factor is high. The objective is not to score higher in terms of accuracy and AUC, but in the number of services chosen.

Fig. 4 shows the average reward over the batches for the RL policy (solid black line) and the base recommender (dotted grey line), for both approaches a confidence interval is added based on 50 use cases. The objective of the RL policy is to increase the number of services that a guest chooses compared to the base recommender. The RL policy outperforms the base recommender by 7.60%, because the average

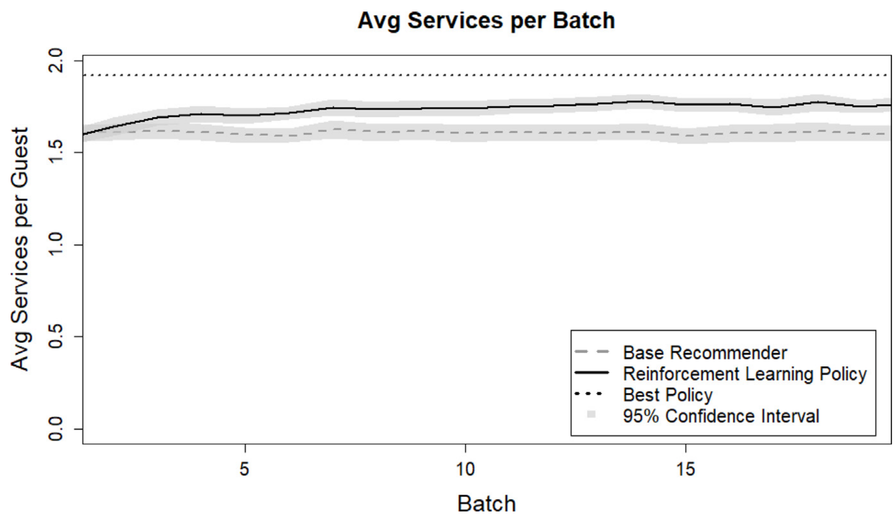



**Fig. 4.** Result of 50 simulated use cases.

**Table 11**
Awareness factor estimates extracted from the RL policy $\pi$.

| | Service | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Average |
| $\rho$ | 0.78 | 0.38 | 0.13 | 0.06 | 0.62 | 0.57 | 0.29 | 0.09 | 0.15 | 0.22 | 0.42 | 0.10 | – |
| $\hat{\rho}$ | 0.73 | 0.46 | 0.11 | 0.08 | 0.64 | 0.62 | 0.40 | 0.08 | 0.22 | 0.26 | 0.47 | 0.13 | – |
| AE | 0.05 | 0.08 | 0.02 | 0.02 | 0.02 | 0.05 | 0.11 | 0.01 | 0.07 | 0.04 | 0.05 | 0.03 | 0.05 |
| PE | 6.00 | 21.30 | 13.10 | 30.00 | 2.90 | 8.80 | 36.90 | 12.20 | 46.00 | 19.50 | 12.60 | 29.00 | 19.86 |

number of services chosen per guest is 1.73 compared to 1.61 for the base recommender. The results have a confidence interval with a width of 0.1 due to the added noise when simulating services interest (see Eq. (2) in Section 4.2). Fig. 4 shows the learning process of the awareness factors during batch 1 through 5 since the average rewards increases before it stabilizes. This increase can be explained by the initialization of awareness factors, which at the start of each simulation is set to 0 and updated after each batch.

The *Best Policy* is present (dotted black lines) within Fig. 4. The Best Policy is defined if the interests in services $Y$ and the awareness factors $\rho$ are known to the agent. The difference between the agent of the RL policy and the Best Policy is known as regret. The value of the Best Policy is the average of the chosen services by guests over 50 simulations. The gap between the Best Policy and RL policy is explained by the error of the estimate in interests of services and the error of estimated awareness factors. This error is explained by the noise that is added to the simulated guests features.

The mean estimated awareness factor for each service over 50 simulation is presented in Table 11. The absolute error (AE) and percentage error (PE) are added as accuracy measurements. On average, the absolute error is 0.05 between the actual and estimated awareness factor. Fig. 5 displays a visual summary of the estimated awareness factor for each service.

With the RL policy, guests with a high estimated interest in a service are exposed to recommendations where this service is deliberately not recommended. The policy selects an another action $a$, where this service is not included in the top-$N$ recommended services, because it has a higher estimated awareness factor. The overlap between actions of the base recommender and RL policy is on average 8.74% over 50 simulations with 10,000 instances each. The recommended services play an important role in explaining why the RL policy outperforms the base recommender. Services with a high awareness factor, e.g., service 1, 5, and 6, are recommended less when applying the RL policy compared to the base recommender. On average, the RL policy recommends services 1, 5, and 6 as 4.52%, 7.42%, and 7.87% respectively, while on average the base recommender includes the same services 36.65%, 43.26%, and 39.61% of the time with the initial recommendation.

## 5. Theoretical & practical implications

The success of implementing a RS in the hospitality industry largely depends on its ability to overcome various challenges that can decrease its contribution. This raises questions about the theoretical and practical implications of such challenges and how they can be addressed. Overcoming the challenges that RSs encounter in the hospitality industry can unleash their potential and result in more effective implementations. It is recommended that future research investigate different techniques and approaches to surmount these challenges and enhance the performance of RSs in hospitality.

In practice, challenges for a RS in hospitality may decrease the contribution, which endangers the success of the implementation and ultimately of the business strategy [22]. One of the challenges is the change in the availability of additional services, due to seasonality or rotation of services. For example, a hotel in the mountains receives winter sport tourism in the winter, but hikers in the summer, and both these seasons require different services. Another challenge is the fluid nature of guest bases in a globalizing world [44]. For example, the number of Asian tourists in Europe has increased significantly in recent years [45]. People from different cultures also have other preferences and expectations for services [46]. These factors lead to an ever-changing catalogue of available services.

The additional reinforcement learning policy can only be applied when recommendations and choices are linked. The results of Section 4.5 show the potential gain when these data points are stored. The gain is two-fold since not only is additional revenue generated by the additional chosen services, but the hotel stay is more complete than it is without the additional services. An investment in the process and storage of these data points is required to unlock this potential.

It is important to note that the modelling of the exposure effect assumes a service-specific awareness factor instead of user-specific. However, in practice, there is a user-specific awareness set since recurring guests might have more information about the available services, which affects the awareness factor. The awareness set can also differ for a season within a year due to a catalogue change. The awareness factor

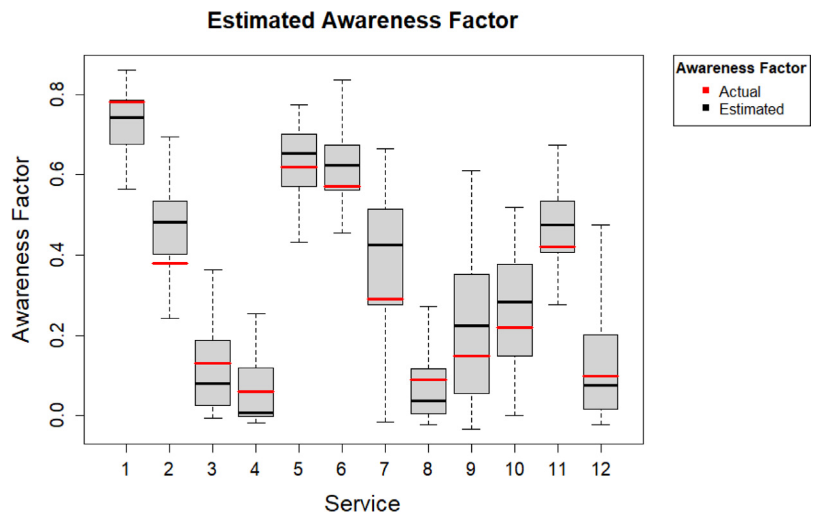



**Fig. 5.** Awareness factor estimates per service of 50 simulations.

**Table A.1**
Overview of the features of the empirical case study.

| Features | Data type | Description |
|---|---|---|
| Length of stay | Integer | Duration of stay |
| Lead time days | Integer | Number of days between booking Date and arrival date |
| Number of persons | Integer | Number of guests |
| Number of children | Integer | Number of children |
| Rate plan average daily rate | Float | Total costs per stay day |
| Rate plan total | Float | Total costs of the reservations |
| Weekend stay | Binary | Stay during the weekend only (Fri - Sun) |
| Weekday stay | Binary | Stay during the week days (Mon-Thu) |
| Group | Binary | If the reservation is part of a group |
| Company | Binary | If the reservation is associated to a company |
| Repeat | Binary | If the primary guest is a recurring guest |
| Arrival month (x12) | Binary | Month of the year the stay occurs |
| Arrival day (x7) | Binary | Day of the week the stay occurs |
| Booking day (x7) | Binary | Day of the week the reservation is made |
| Unit type (x4) | Binary | Selected room type |

can be extended as a function of additional features, e.g., binary features to indicate a recurring guest. Further research can be conducted in relaxing the assumption on the awareness set.

In Section 4.5, the results of the exposure effect are generated given a trained base recommender model, which is not updated when additional data becomes available. The exposure effect is isolated by keeping the base recommender fixed over the batches. However, the base recommender needs to be retrained over time when dealing with changing guest needs.

The exploration method used in the research was simple and static, but more advanced exploration methods, such as reward-based epsilon greedy, can be implemented to understand if the learning process is increased. Incremental learning can also be explored as a highly effective technique compared to epsilon greedy, although it may affect the effectiveness of recommending other services.

## 6. Conclusion

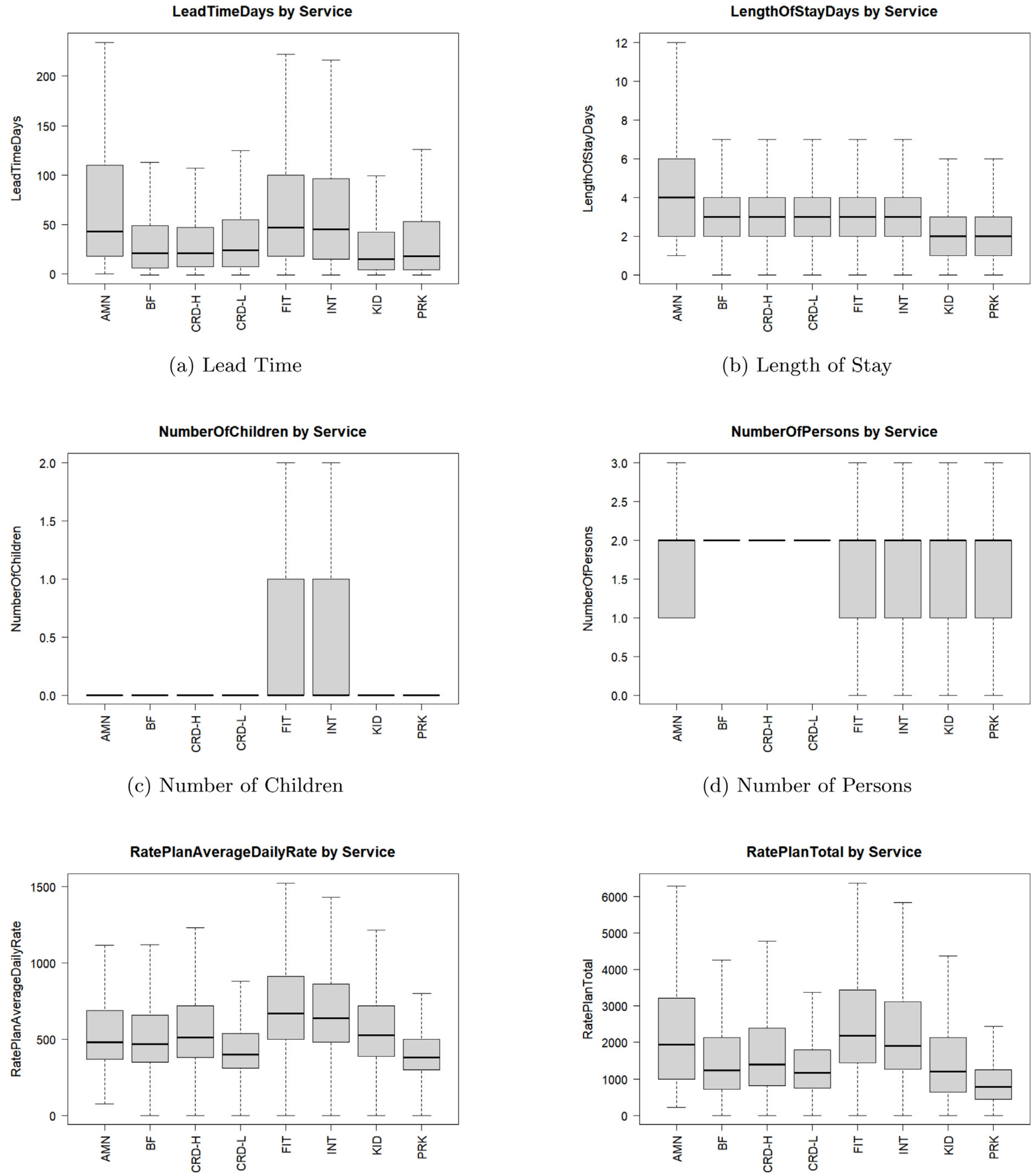
This study provides a guideline on how hospitality can incorporate a recommender system within the guest journey to increase the number of services chosen by guests. This study also highlights a recommender system effect, the exposure effect, and a way of turning this into an advantage to boost the number of chosen services. The main focus was to maximize the number of services chosen by guests to accommodate an enhanced guest experience.

The empirical case study shows the applicability of a recommender system and the required data points during a booking process, which is part of the guest journey. This includes a total of 41 guest features from the input to estimate the interest in services. The accuracy and AUC of XGBoost outperformed other machine learning methods and naive strategies on an overall and service level. The evaluation metrics prove the added value of a machine learning approach compared to a naive baseline approach, e.g., recommending the most popular services.

Given the findings of the empirical case study, an offline setting enabled the policy on dealing with the exposure effect. The exposure effect is the merely added value of recommending services compared to not recommending services. This effect relates to the awareness factor per service, which influences the interest in a service. The additional policy is a RL approach since the exposure effects are unknown from the start of the simulation and require exploration. These effects are unknown since historic recommendations were unavailable during the initiation of the RL policy, which is similar to the empirical case study. This policy, which uses estimated interest in services and the estimated awareness factors, resulted in an increase of 7.60% in the average number of services chosen per guest compared to a trained base recommender model. The driver of this increase is deliberately not recommending services with high interest. The RL policy increases the average number of services chosen while the evaluation metrics, accuracy and AUC, decrease compared to a ML model.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.




(a) Lead Time (b) Length of Stay

(c) Number of Children (d) Number of Persons

(e) Rate Plan - Average Daily Rate (f) Rate Plan - Total

**Fig. B.1.** Boxplots of numerical features.

## Data availability

The authors do not have permission to share data.

## Appendix A. Available data

See Table A.1.

## Appendix B. Numerical features

See Fig. B.1.

## Appendix C. Categorical feature correlation

See Table C.1.

**Table C.1**
Cramér V association.

| Features | Amenity | Breakfast | F&B Credit High | F&B Credit Low | Fitness | Internet | Kids | Parking |
|---|---|---|---|---|---|---|---|---|
| Weekend stay | 0.00 | 0.02 | 0.03 | 0.03 | 0.00 | 0.05 | 0.00 | 0.01 |
| Weekday stay | 0.00 | 0.00 | 0.02 | 0.02 | 0.01 | 0.02 | 0.01 | 0.00 |
| Group | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Company | 0.48 | 0.03 | 0.32 | 0.23 | 0.39 | 0.25 | 0.05 | 0.20 |
| Repeat stay | 0.04 | 0.00 | 0.00 | 0.02 | 0.02 | 0.02 | 0.02 | 0.01 |
| Unit type 1 | 0.17 | 0.36 | 0.14 | 0.25 | 0.53 | 0.22 | 0.07 | 0.23 |
| Unit type 2 | 0.21 | 0.29 | 0.12 | 0.08 | 0.44 | 0.06 | 0.06 | 0.22 |
| Unit type 3 | 0.08 | 0.12 | 0.04 | 0.33 | 0.17 | 0.30 | 0.01 | 0.01 |
| Unit type 4 | 0.02 | 0.04 | 0.01 | 0.09 | 0.06 | 0.08 | 0.00 | 0.02 |
| Arrival Monday | 0.00 | 0.00 | 0.02 | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 |
| Arrival Tuesday | 0.00 | 0.01 | 0.01 | 0.00 | 0.00 | 0.01 | 0.03 | 0.01 |
| Arrival Wednesday | 0.02 | 0.01 | 0.03 | 0.02 | 0.01 | 0.00 | 0.00 | 0.01 |
| Arrival Thursday | 0.00 | 0.01 | 0.04 | 0.00 | 0.02 | 0.02 | 0.01 | 0.01 |
| Arrival Friday | 0.01 | 0.00 | 0.04 | 0.00 | 0.02 | 0.03 | 0.02 | 0.01 |
| Arrival Saturday | 0.02 | 0.01 | 0.05 | 0.03 | 0.02 | 0.03 | 0.01 | 0.04 |
| Arrival Sunday | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 |
| Arrival January | 0.00 | 0.02 | 0.00 | 0.01 | 0.03 | 0.00 | 0.01 | 0.02 |
| Arrival February | 0.03 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.01 | 0.03 |
| Arrival March | 0.01 | 0.01 | 0.03 | 0.02 | 0.03 | 0.02 | 0.02 | 0.01 |
| Arrival April | 0.00 | 0.01 | 0.00 | 0.01 | 0.02 | 0.01 | 0.00 | 0.02 |
| Arrival May | 0.04 | 0.00 | 0.05 | 0.01 | 0.05 | 0.02 | 0.01 | 0.04 |
| Arrival June | 0.07 | 0.00 | 0.04 | 0.05 | 0.02 | 0.04 | 0.00 | 0.01 |
| Arrival July | 0.05 | 0.02 | 0.07 | 0.01 | 0.05 | 0.02 | 0.02 | 0.04 |
| Arrival August | 0.00 | 0.01 | 0.05 | 0.01 | 0.00 | 0.01 | 0.02 | 0.01 |
| Arrival September | 0.01 | 0.00 | 0.05 | 0.01 | 0.04 | 0.00 | 0.01 | 0.05 |
| Arrival October | 0.01 | 0.00 | 0.01 | 0.00 | 0.00 | 0.02 | 0.02 | 0.02 |
| Arrival November | 0.02 | 0.01 | 0.06 | 0.01 | 0.05 | 0.01 | 0.00 | 0.00 |
| Arrival December | 0.04 | 0.01 | 0.10 | 0.03 | 0.04 | 0.04 | 0.04 | 0.02 |
| Booking Monday | 0.00 | 0.00 | 0.03 | 0.00 | 0.01 | 0.00 | 0.00 | 0.02 |
| Booking Tuesday | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 |
| Booking Wednesday | 0.01 | 0.01 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 |
| Booking Thursday | 0.00 | 0.01 | 0.01 | 0.00 | 0.02 | 0.00 | 0.03 | 0.01 |
| Booking Friday | 0.01 | 0.02 | 0.00 | 0.00 | 0.01 | 0.00 | 0.01 | 0.01 |
| Booking Saturday | 0.00 | 0.01 | 0.01 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 |
| Booking Sunday | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.04 | 0.00 |

## References


[1] F. Ricci, L. Rokach, B. Shapira, Recommender systems: Introduction and challenges, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer US, Boston, MA, 2015, pp. 1–34, http://dx.doi.org/10.1007/978-1-4899-7637-6.

[2] L. Gamidullaeva, A. Finogeev, M. Kataev, L. Bulysheva, A design concept for a tourism recommender system for regional development, Algorithms 16 (1) (2023) http://dx.doi.org/10.3390/a16010058.

[3] X. Chen, Z. Owen, C. Pixton, D. Simchi-Levi, A statistical learning approach to personalization in revenue management, Manage. Sci. 68 (3) (2021) 1923–1937, http://dx.doi.org/10.1287/mnsc.2020.3772.

[4] C.A. Gomez-Uribe, N. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Trans. Manag. Inf. Syst. 6 (4) (2016) 1–19, http://dx.doi.org/10.1145/2843948.

[5] E. Brynjolfsson, Y.J. Hu, M.D. Smith, Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers, Manage. Sci. 49 (11) (2003) 1580–1596, http://dx.doi.org/10.1287/mnsc.49.11.1580.20580.

[6] J.L. Sarkar, A. Majumder, C.R. Panigrahi, S. Roy, B. Pati, Tourism recommendation system: a survey and future research directions, Multimedia Tools Appl. 82 (6) (2023) 8983–9027, http://dx.doi.org/10.1007/s11042-022-12167-w.

[7] D. Jannach, P. Pu, F. Ricci, M. Zanker, Recommender systems: Trends and frontiers, AI Mag. 43 (2) (2022) 145–150, http://dx.doi.org/10.1002/aaai.12050.

[8] A.L. Karn, R.K. Karna, B.R. Kondamudi, G. Bagale, D.A. Pustokhin, I.V. Pustokhina, S. Sengan, Customer centric hybrid recommendation system for E-commerce applications by integrating hybrid sentiment analysis, Electron. Commer. Res. 23 (1) (2023) 279–314, http://dx.doi.org/10.1007/s10660-022-09630-z.

[9] A. Beerli-Palacio, J.D. Martín-Santana, C. Román-Montoya, Complementary services at hotels in accordance with their pricing strategy and the price sensitivity of tourists, Int. J. Hosp. Manag. 87 (2020) 102458, http://dx.doi.org/10.1016/j.ijhm.2020.102458.

[10] M. Ottenbacher, J. Gnoth, How to develop successful hospitality innovation, Cornell Hotel Restaur. Adm. Q. 46 (2) (2005) 205–222, http://dx.doi.org/10.1177/0010880404271097.

[11] M. Vusumuzi, M. Mfowabo, Fifteen years of recommender systems research in higher education: Current trends and future direction, Appl. Artif. Intell. 37 (1) (2023) 3109–3127, http://dx.doi.org/10.1080/08839514.2023.2175106.

[12] Al-Rousan, M. Ramzi, Badaruddin Mohamed, Customer loyalty and the impacts of service quality:the case of five star hotels in jordan, Int. J. Econ. Manag. Eng. 4 (7) (2010) http://dx.doi.org/10.5281/zenodo.1331937.

[13] N. Hazrati, F. Ricci, Recommender systems effect on the evolution of users' choices distribution, Inf. Process. Manage. 59 (1) (2022) 102766, http://dx.doi.org/10.1016/j.ipm.2021.102766.

[14] D. Fleder, K. Hosanagar, Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity, Manage. Sci. 55 (5) (2009) 697–712, http://dx.doi.org/10.1287/mnsc.1080.0974.

[15] M. Hong, J.J. Jung, Multi-criteria tensor model for tourism recommender systems, Expert Syst. Appl. 170 (2021) 114537, http://dx.doi.org/10.1016/j.eswa.2020.114537.

[16] G. Adomavicius, Tuzhilin A., Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749, http://dx.doi.org/10.1109/TKDE.2005.99.

[17] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI '98, Morgan Kaufmann Publishers Inc., 1998, pp. 43–52, http://dx.doi.org/10.5555/2074094.2074100.

[18] F. McCarey, M.O. Cinneide, N. Kushmerick, A recommender agent for software libraries: An evaluation of memory-based and model-based collaborative filtering, in: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT '06, IEEE Computer Society, 2006, pp. 154–162, http://dx.doi.org/10.1109/IAT.2006.23.

[19] P.H. Aditya, I. Budi, Q. Munajat, A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT x, in: 2016 International Conference on Advanced Computer Science and Information Systems, ICACSIS, 2016, pp. 303–308, http://dx.doi.org/10.1109/ICACSIS.2016.7872755.

[20] D. Yashar, J. Dietmar, B. Alejandro, D. Alessandro, Dario Z., Fairness in recommender systems: research landscape and future directions, in: User Modeling and User-Adapted Interaction, 2023, http://dx.doi.org/10.1007/s11257-023-09364-z.

[21] Q. Li, S.H. Myaeng, B.M. Kim, A probabilistic music recommender considering user opinions and audio features, Inf. Process. Manage. 43 (2) (2007) 473–487, http://dx.doi.org/10.1016/j.ipm.2006.07.005.

[22] A. Van Looy, A. Shafagatova, Business process performance measurement: a structured literature review of indicators, measures and metrics, Springerplus 5 (1) (2016) 1797, http://dx.doi.org/10.1186/s40064-016-3498-1.

[23] G.L. Ciampaglia, A. Nematzadeh, F. Menczer, A. Flammini, How algorithmic popularity bias hinders or promotes quality, Sci. Rep. 8 (1) (2018) 15951, http://dx.doi.org/10.1038/s41598-018-34203-2.

[24] A. Oulasvirta, J.P. Hukkinen, B. Schwartz, When more is less: The paradox of choice in search engine use, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, Association for Computing Machinery, 2009, pp. 516–523, http://dx.doi.org/10.1145/1571941.1572030.

[25] Y. Wang, W. Ma, M. Zhang, Y. Liu, S. Ma, A survey on the fairness of recommender systems, Assoc. Comput. Mach. 41 (3) (2023) http://dx.doi.org/10.1145/3547333.

[26] A. Kumar, K. Hosanagar, Measuring the value of recommendation links on product demand, Inf. Syst. Res. 30 (3) (2019) 819–838, http://dx.doi.org/10.1287/isre.2018.0833.

[27] K. Krauth, Y. Wang, M.I. Jordan, Breaking feedback loops in recommender systems with causal inference, 2022, http://dx.doi.org/10.48550/ARXIV.2207.01616, arXiv.

[28] A. Khenissi, O. Nasraoui, Modeling and counteracting exposure bias in recommender systems, in: Electronic Theses and Dissertations, 2020, http://dx.doi.org/10.18297/etd/3182.

[29] R. Jiang, S. Chiappa, T. Lattimore, A. György, P. Kohli, Degenerate feedback loops in recommender systems, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, ACM, 2019, pp. 383–390, http://dx.doi.org/10.1145/3306618.3314288.

[30] J. Suleri, R. Meijer, E. Tarus, Exploring hotel identity by focusing on customer experience analysis, Res. Hosp. Manag. 11 (2) (2021) 113–120, http://dx.doi.org/10.1080/22243534.2021.1917178.

[31] H. Cramér, Mathematical Methods of Statistics (PMS-9), Princeton University Press, 1999, http://dx.doi.org/10.2307/j.ctt1bpm9r4.

[32] L. Zientek, Z. Yetkiner, S. Özel, J. Allen, Reporting confidence intervals and effect sizes: Collecting the evidence, Career Tech. Edu. Res. J. 37 (2) (2012) 277–295, http://dx.doi.org/10.5328/cter37.3.277.

[33] A.H. Hon, S.S. Lui, Employee creativity and innovation in organizations, Int. J. Contemp. Hosp. Manag. 28 (5) (2016) 862–885, http://dx.doi.org/10.1108/IJCHM-09-2014-0454.

[34] H.F. Yu, F.L. Huang, C.J. Lin, Dual coordinate descent methods for logistic regression and maximum entropy models, Mach. Learn. 85 (1) (2011) 41–75, http://dx.doi.org/10.1007/s10994-010-5221-8.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830, http://dx.doi.org/10.5555/1953048.2078195.

[36] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[37] J. Yoon, E. Yang, J. Lee, S. Hwang, Lifelong learning with dynamically expandable networks, 2017, http://dx.doi.org/10.48550/arXiv.1708.01547, CoRR.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035, http://dx.doi.org/10.5555/3454287.3455008.

[39] S. Maya, K. Ueno, T. Nishikawa, dLSTM: a new approach for anomaly detection using deep learning with delayed prediction, Int. J. Data Sci. Anal. 8 (2) (2019) 137–164, http://dx.doi.org/10.1007/s41060-019-00186-0.

[40] M. Casillo, F. Colace, D. Conte, M. Lombardi, D. Santaniello, C. Valentino, Context-aware recommender systems and cultural heritage: a survey, J. Ambient Intell. Humaniz. Comput. 14 (4) (2023) 3109–3127, http://dx.doi.org/10.1007/s12652-021-03438-9.

[41] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874, http://dx.doi.org/10.1016/j.patrec.2005.10.010.

[42] C.E. Metz, Basic principles of ROC analysis, Semin. Nucl. Med. 8 (4) (1978) 283–298, http://dx.doi.org/10.1016/s0001-2998(78)80014-2.

[43] R. Crites, A. Barto, Improving elevator performance using reinforcement learning, in: Advances in Neural Information Processing Systems, 8, MIT Press, 1995.

[44] P. Niewiadomski, Towards an economic-geographical approach to the globalisation of the hotel industry, Tour. Geograph. 16 (1) (2014) 48–67, http://dx.doi.org/10.1080/14616688.2013.867528.

[45] H.T. Bui, A. Trupp, The development and diversity of Asian tourism in Europe: The case of Vienna, Int. J. Tour. Sci. 14 (2) (2014) 1–17, http://dx.doi.org/10.1080/15980634.2014.11434689.

[46] L.P. Mey, A.K. Akbar, D.Y.G. Fie, Measuring service quality and customer satisfaction of the hotels in Malaysia: Malaysian, Asian and non-Asian hotel guests, J. Hosp. Tour. Manag. 13 (2) (2006) 144–160, http://dx.doi.org/10.1375/jhtm.13.2.144.