Baldominos, A.; Quintana, D. Data-Driven Interaction Review of an Ed-Tech Application. *Sensors* **2019,** 19, 1910.

DOI: 10.3390/s19081910

sensors

MDPI

Article

Data-Driven Interaction Review of an Ed-Tech Application

Alejandro Baldominos [1,2] and David Quintana [1,*]

[1] Department of the Computer Science, Universidad Carlos III de Madrid, 28911 Leganés, Spain; abaldomi@inf.uc3m.es

[2] Smile and Learn Digital Creations, 28043 Madrid, Spain

* Correspondence: dquintan@inf.uc3m.es; Tel.: +34-91-624-9109

Received: 12 February 2019; Accepted: 18 April 2019; Published: 22 April 2019

Abstract: Smile and Learn is an Ed-Tech company that runs a smart library with more that 100 applications, games and interactive stories, aimed at children aged two to 10 and their families. The platform gathers thousands of data points from the interaction with the system to subsequently offer reports and recommendations. Given the complexity of navigating all the content, the library implements a recommender system. The purpose of this paper is to evaluate two aspects of such system focused on children: the influence of the order of recommendations on user exploratory behavior, and the impact of the choice of the recommendation algorithm on engagement. The assessment, based on data collected between 15 October 2018 and 1 December 2018, required the analysis of the number of clicks performed on the recommendations depending on their ordering, and an A/B/C testing where two standard recommendation algorithms were compared with a random recommendation that served as baseline. The results suggest a direct connection between the order of the recommendation and the interest raised, and the superiority of recommendations based on popularity against other alternatives.

Keywords: educational technologies; recommender systems; artificial intelligence

1. Introduction and Background

Smile and Learn's smart library is an application in the educational technology (Ed-Tech) space which is aimed at children. As of December 2018, the platform features a total of 107 games, which are grouped according to Gardner's theory of multiple intelligences [1].

Contents, which include games, stories and videos, are designed to be used in different devices and rely on a common framework. The application registers thousands of data points as a result of user interaction. Based of these, the system can then generate personalized reports and recommendations relevant to users, parents and educators.

The initial interaction with the application requires choosing among a large set of alternatives that are organized according to broad categories. These can be identified in Figure 1. The different games are grouped in so-called "worlds", with each world corresponding to an intelligence: science (naturalistic), spatial (visual-spatial), multiplayer (group-interpersonal), logic (logical-mathematical), literacy (verbal-linguistic), emotions (emotional-intrapersonal), and arts (artistic). There is one additional world named after the user, which consists on a virtual village where the child interacts with characters to improve her wealth. Here, the dynamics require abilities from all intelligences.

Once a world is chosen, apps are displayed grouped in categories, with one category per line. The children can use the vertical scroll (swipe pattern) to navigate through categories, and the horizontal scroll to navigate between apps within a category. An example screenshot from the Science world is shown in Figure 2.

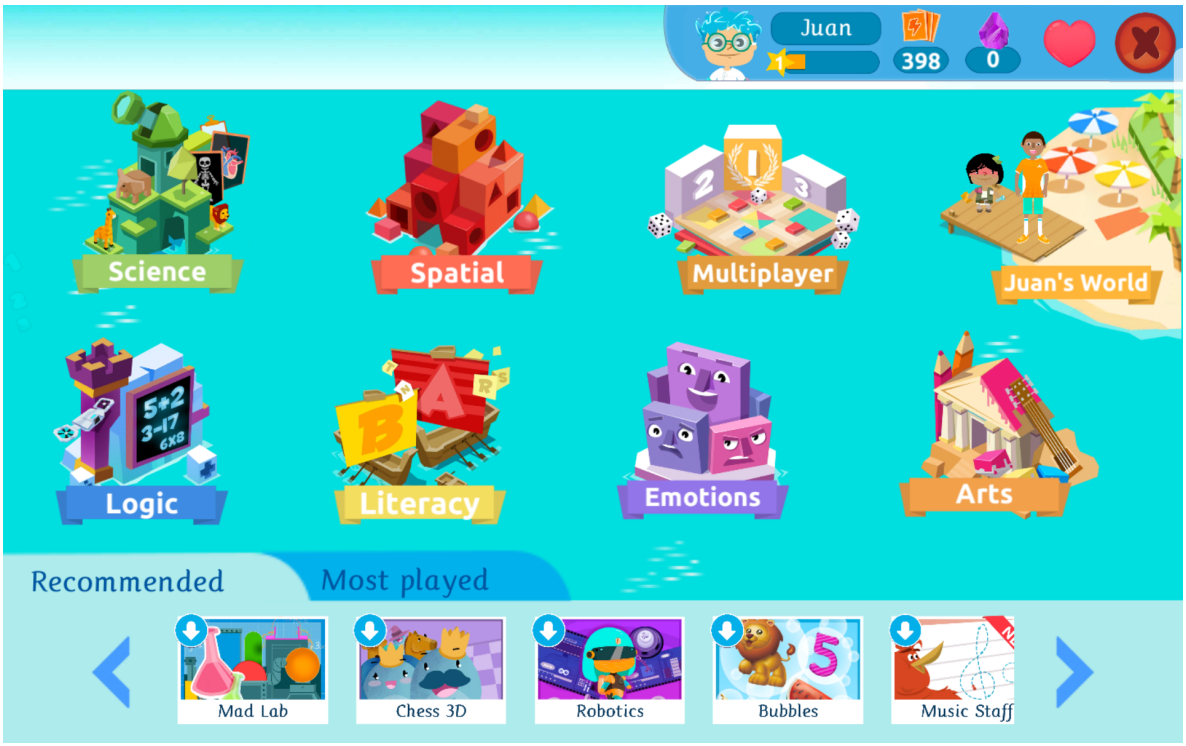

Figure 1. Screenshot of the main menu of Smile and Learn, showing the different worlds available in the application, each world corresponding to an intelligence.

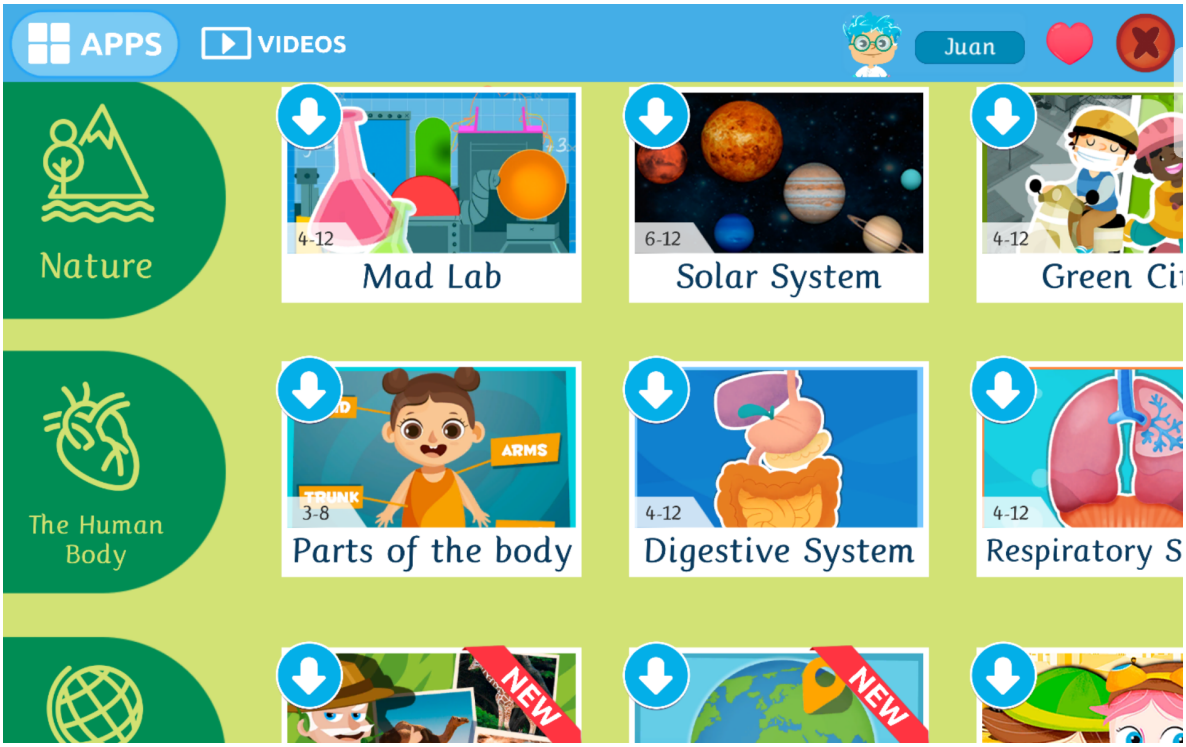

Figure 2. Screenshot of a world menu (in this case, the science world) in Smile and Learn, where apps belonging to the same world (or intelligence) are listed grouped by category.

Due to the increase in the number of games, which is still growing month after month, navigation through the application is becoming more cumbersome, meaning that it can take more time and effort to search for a certain game within the application. In order to alleviate this issue, we have introduced a navigation ribbon at the bottom of the main menu which allows a fast access to some

chosen applications. As it can be seen at the bottom of Figure 1, this navigation pattern consists of two tabs: "recommended" and "most played". The latter displays a list of the five top-played apps for the child using the app, turning into a useful mechanism for providing a fast access to those frequent apps.

Regarding the former tab, it displays a list of up to seven apps recommended to children based on their usage behavior. The design motif behind this recommender system is not only to enable fast access to those apps which might be of interest for a child, but also to enhance exploration: letting children discover games which they might not reach if they had to find them by navigating through the app.

The range of strategies developed to assign users to items is broad. Some well established alternatives would be the recommendation of the most popular items [2]; collaborative filtering [3]; content-based approaches [4] or usage context-based similarity [5]. In addition to the basic possibilities, there is room for hybrid strategies that combine the output of several canonical strategies to generate their output.

Currently, two different strategies are implemented in Smile and Learn:

- Popular: this approach recommends the most popular applications among other users. The rationale behind this approach is that it is likely that some games are specially enjoyable for most children, because of their quality, design or playability, and therefore, they are safe to recommend, as there is a high chance that other children will find them engaging.
- Collaborative filtering: in this approach, the historical records of usage of all children are used in order to generate recommendations. The idea behind collaborative filtering is to find children similar to the one whose recommendations are being generated, so that these recommendations consist of games that have been played by these similar children, but not the child being targeted.

Both these strategies are based on implicit feedback, since explicit feedback given by children can turn out to be unreliable or difficult to interpret. Out of the different implicit parameters that could be taken into account [6], we have used the number of games played by child and app, as well as the duration of such games.

The specifics of the implementation of these algorithms are discussed in relation to a prior version of the system by Ruiz–Iniesta et al. [7].

In the navigation bar displayed at the bottom of the main menu, recommendations from different recommenders could be combined. Also, it is worth noting that while there is a maximum of seven recommended games, in some screen sizes or formats it can occur that less than seven are displayed and then the list becomes scrollable. Based on our experience, most screens are able to show five recommendations, with the last two being hidden and accessible only upon scroll (through a swipe pattern or by pressing the side arrows, an example can be seen in Figure 1).

In this paper, we want to design a sound experiment in order to determine which recommender strategy is working best, i.e., which one is translating into a higher engagement of children with the recommended games. We are also interested in getting insights about the interaction patterns taken by children when dealing with the recommendations. With this information, we could improve the system and enhance user experience.

This work contributes valuable new evidence on the reaction of children to two well-established methods of generating recommendations, and assessing the importance of the order of presentation. Information like this, specifically focused on children in the educational space, is both scarce and very hard to obtain. Given that the related literature on children is very limited, we consider that the results might be very relevant to developers of K-12 Ed-Tech applications.

The rest of the document is structured as follows: in Section 2 we present related works on recommender systems and interaction patterns within the context of an Ed-Tech application. That will be followed by Section 3, where we describe the experimental methodology. The results are then reported in Section 4. Finally, conclusive remarks are presented in Section 5 along with some suggested future lines of work.

2. Related Work

Applications of technology to educational processes (a field known under the term "Ed-Tech") are not particularly new. However, it has only been in the last decade that the availability and ubiquity of technology devices (such as smart-phones or tablets) has reached a point in which these applications can be deployed in large-scale settings.

Additionally, the implementation of data science or machine learning techniques are allowing to extend Ed-Tech far beyond the classical definition of using technology to deliver contents to an audience (e.g., slides, interactive videos, etc). Bhattacharya and Nath [8] discuss how novel development allow tracking the progress of individual users, detecting strengths and weaknesses, and providing a customized experience to enhance the learning process, among other possibilities.

Many examples of such novel applications can be found in the literature in recent years. For example, Charleer et al. [9] presented a learning analytics dashboard aimed at improving the communication in advising sessions, helping to increase students' motivation. Lange et al. [10] also made related contributions in the space of virtual training centers and Käser et al. [11] have proposed an approach to student modelling to represent and predict students' knowledge and skills. Some authors have been working on the prediction of academic performance [12] and others, like Ueno and Miyazawa [13] introduced a system designed to scaffold learning in specific domains such as programming.

An important subset of Ed-Tech applications are those aiming at providing a customized experience by suggesting users a certain topic or learning process. This problem can generally be regarded as a recommendation problem. Recommender systems, the instruments used to tackle the problem, have a three-decade long history and have been the subject of a sizable amount of research. These efforts on recommender systems in general have been surveyed by Bobadilla et al. [14] and, more recently, by Lu et al. [15]. If we focus specifically in education, the literature reviews authored by Drachsler et al. [16] and Erdt et al. [17] are specially insightful and comprehensive.

The design and implementation of recommender systems in the education space poses a number of difficulties, as discussed by Tarus and Niu [18], and so does their evaluation [19]. Among these challenges, we can mention the selection of the right algorithm (or set of algorithms) and the design of the interface. Even though the first aspect is not settled yet, there have been valuable efforts to contribute evidence in this regard like a recent work by Kopeinik et al. [20].

These authors cite among the best-established and computationally inexpensive tag and resource recommendation strategies for technology-enhanced learning: base level learning equation with associative component [21]; collaborative filtering [3]; content-based [4]; most popular [2]; SUSTAIN [22] or usage context-based similarity [5].

In this regard, studies like [23] offer systematic reviews of different algorithmic alternatives, while [24,25] provide specific lists of advantages and disadvantages of the main recommendation techniques for technology-enhanced learning. These are briefly summarized in Table 1, which adapts a very similar one included in the last two mentioned studies, adding popularity-based recommendations to the set of techniques.

Besides the alternatives displayed in the table, an additional category of recommender systems are hybrid approaches. In this case, two or more of these techniques are fused in order to alleviate some of the disadvantages of a particular recommender. This possibility was discussed in detail in a survey on Next Generation of Recommender Systems by Adomavicius and Tuzhilin [26] and, later by Aamir and Bhusry [27].

Table 1. Characteristics of main recommendation techniques suitable for technology-enhanced learning. CF: collaborative filtering. Adapted from Drachsler et al. [24] and Manouselis et al. [25].

Technique	Description	Advantages	Disadvantages
Popularity	Recommends items that are popular among all users.	No new user problem. No content analysis. Domain-independent.	Only popular taste. Insensitive to changes of preferences.
User-based CF	Users who rated the same item similarly probably have the same taste. Based on this assumption, this technique recommends the unseen items already rated by similar users.	No content analysis. Domain-independent. Quality improves. Bottom-up approach. Serendipity.	New user problem. New item problem. Popular taste. Scalability. Sparsity. Cold start problem.
Item-based CF	Focus on items, assuming that the items rated similarly are probably similar. It recommends items with the highest correlation (based on ratings for the items).	No content analysis. Domain-independent. Quality improves. Bottom-up approach. Serendipity.	New item problem. Popular taste. Sparsity. Cold start problem.
Stereotypes or demographics CF	Users with similar attributes are matched, then it recommends items that are preferred by similar users (based on user data instead of ratings).	No cold start problem. Domain-independent. Serendipity.	Obtaining information. Insufficient information. Only popular taste. Obtaining metadata information. Maintenance ontology.
Case-based reasoning	Assumes that if a user likes a certain item, she or he will probably also like similar items. Recommends new but similar items.	No content analysis. Domain-independent. Quality improves	New user problem. Overspecialisation. Sparsity. Cold start problem.
Attribute-based techniques	Recommends items based on the matching of their attributes to the user profile. Attributes could be weighted for their importance to the user.	No cold start problem. No new user/new item problem. Sensitive to changes of preferences. Can include non-item-related features. Can map from user needs to items.	Does not learn. Only works with categories. Ontology modelling and maintenance is required. Overspecialisation.

Among the works that could illustrate this possibility, we could mention one by Rodriguez et al. [28], who built a hybrid recommender combining content-based, collaborative filtering and knowledge-based approaches to develop a recommender system that suggests learning objects extracted from a repository in an educational setting. Another example can be found in the work by Salehi and Kmalabadi [29],

which combines content-based and collaborative filtering recommenders, in this case to suggest appropriate learning materials. Bourkoukou and El Bachari [30] introduce a recommender system for web-based education that adapts its recommendations of learning objects to the learning styles of users. In this case, the hybrid recommender system is based on collaborative filtering and association rule mining. Finally, Nafea et al. [31] recently introduced the ULEARN system, that also selects and sequences learning objects that match the learning styles of the users. To this end, they rely on a hybrid recommendation approach that includes collaborative filtering and content filtering.

The second aspect is specially complicated when it involves children, as their cognitive development makes their interests and capabilities change dramatically over a period of relatively few years [32], and the fact that intuitions by adult designers might not be correct [33]. Even though there have been advances in this aspect, like the contribution of Wu et al. [34] on interface design for children, there is still a significant amount of work to be done.

If we focus on recommendation for children, even though we could mention some relevant works like that by Pera and Ng [35] the volume is still very scarce. To illustrate this, it is worth mentioning that the first specialized workshop, KidRec, took place in 2017 [36]. As Deldjoo et al. [37] explain, recommender systems have been traditionally focused on adults and, when it comes to children, the field is still in its infancy.

3. Materials and Methods

In this study we intend to compare the performance of two popular recommender strategies to suggest potential games of interest to children. As it was described earlier, these are providing recommendations based on apps commonly played by most children, and collaborative filtering, whose recommendations are based on games played by similar users, i.e., those who have a similar record of played games.

In order to get a better understanding of which strategy was working best, we tested the performance of both recommender strategies and, additionally, we compared them against a baseline recommender which made random recommendations. Given that the aim of the work in this regard was descriptive, not prescriptive, an A/B/C testing, a well-established evaluation approach in this context, was performed. This methodology was recently used by Kalloori et al. [38] to evaluate their recommender algorithm in Taobao display advertising platform in production, and so was used by Hidasi and Karatzoglou [39] to evaluate theirs on a large scale online test on an online video portal.

During the whole period in which the experiment was running, each child was assigned one group (either A, B or C) randomly following a uniform distribution. When running the recommender, children received recommendations coming from a different strategy based on their group:

A. Popular strategy was used.
B. The collaborative filtering strategy was used.
C. The random strategy was used.

Instead of generating all the recommendations (a maximum of seven) from the resulting strategy, only the first three recommendations were computed using the recommender corresponding to the child's group, and the remaining recommendations were chosen randomly.

In all cases, some filters were applied in order to opt out of some recommendations which might not be suitable:

- Some games were blacklisted, meaning that they might not have enough quality as to be recommended (e.g., they were in a beta stage).
- Games were filtered out if they were not available in the version of the app owned by the user (e.g., a game was introduced in version 4 and the child is using version 3).
- Games were filtered out if they were not designed for the range of age of the target child (e.g., a game is aimed at children 4–6 years old, but the child is 3 years old).

It is worth mentioning that these filters were applied in all cases, even in the random recommendation strategy.

3.1. Algorithms

As stated earlier, three different recommendation strategies were followed. One of these strategies was entirely random, and was used as a baseline. Its implementation was trivial, as apps to be recommended were randomly sampled from among the whole set.

3.1.1. Popular Recommender

The second strategy was the popular one. For the popular recommender, the algorithm computed a metric of normalized interest, $\overline{I}(a)$, for each app a, which was immediately derived from the number of games ($\#Games(a)$) and the amount of time in days that the app has been available in the library ($Age(a)$), as shown in Equation (1):

$$\overline{I}(a) = \frac{\#Games(a)}{Age(a)}. \quad (1)$$

Once the normalized interest was computed for each app, then all apps were ranked according to this value and the top-k apps were returned by the recommender.

Two design decisions about the popular recommendation system are worth mentioning. First, only the amount of games played for an app was taken into account, while the duration of these games was ignored. This was done on purpose, as each app has its own particularities and some might naturally lead to longer games than others, and we did not want to set these apps higher in the ranking. However, it must be noted that games with a duration of less than five seconds have been ignored. Second, the interest was normalized by the amount of time that the app has been published in the library. While another alternative could have been chosen (for example, considering only the data from the last d days), we have intentionally performed this normalization so that new apps had a higher chance to be chosen by the recommender.

3.1.2. Collaborative Filtering Recommender

The third recommendation strategy was based on collaborative filtering. This strategy relies on implicit rather than explicit feedback. This decision was motivated by the fact that small children have their own interacting patterns, and might be unable to properly give credit or score to a game after playing it [37]. For this reason, a system based on explicit feedback could be unreliable. Instead, we are basing recommendations on implicit patterns of interaction, and in particular in how much and how long a child played with a game.

The idea behind collaborative filtering was to recommend apps to a child which can be of high interest based on the preferences of other children which are deemed similar. Therefore, our implementation of collaborative filtering was done in two stages. In the first stage, the neighborhood was formed by choosing children similar to the one to whom recommendations are being provided. In the second phase, a rating was predicted for each candidate app based on the neighborhood and the top-k apps were selected for the recommendation.

In the neighborhood formation stage, for each target child to whom recommendations were provided, we needed to select similar children on which the recommendations relied. First of all, only children with at least one game of the same application in common were considered for the neighborhood (or equivalently, children whose set of played apps was disjoint with that of the target child are systematically excluded for the neighborhood). Each child's profile included some demographic data including the age and the gender. In this case, the age was used to form the neighborhood; however, the gender was intentionally omitted to prevent unintended biases.

To establish the neighborhood, a similarity metric between two children c_1 and c_2 was defined as shown in Equation (2):

$$Sim(c_1, c_2) = 0.4 \cdot Sim_{age}(c_1, c_2) + 0.6 \cdot \frac{|Apps(c_1) \cap Apps(c_2)|}{|Apps(c_1) \cup Apps(c_2)|}. \quad (2)$$

As it can be seen, this similarity measure relied on a weighted average of two different criteria: the similarity in the age and the similarity of the gaming history of the two children, and returned a value in the range $[0, 1]$. In the case of the former criterion, it was computed following Equation (3):

$$Sim_{age}(c_1, c_2) = \begin{cases} 1 & \text{if } Age(c_1) = Age(c_2), \\ 0.5 & \text{if } |Age(c_1) - Age(c_2)| = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In the previous equation, $Age(c)$ refers to the age of child c in years. The previous function has been designed to fit an educational reality: a difference of more than one year in such an early stage of the educational process (two to 10 years old) was very large, and two children with such an age gap were likely be very different.

The latter part of Equation (2) refers to the similarity in terms of the used apps, which was the main criterion for the implicit feedback collaborative filtering. This was computed as the intersection-over-union of the sets of apps played intensively by the two children. By "intensively" we refer to apps that have a cumulative of at least 10 games and 60 s of game. This filtering has been done to prevent considering apps which might been rarely played by a children while examining or exploring Smile and Learn library.

Neighborhood formation was carried out by choosing the top-100 children with higher similarity. A minimum similarity of 0.5 was enforced in order for a child to be included in the neighborhood.

Finally, for each candidate app a to be recommended we computed the average of the ratings $r(n, a)$ for each children n in the neighborhood N. Since we wanted to deal with implicit feedback, this rating corresponds to the number of games of children c in app a. The average of ratings was weighted by the similarity of the target children and each neighbor. In summary, the interest of children c in app a, $I(c, a)$ was computed as shown in Equation (4):

$$I(c, a) = \frac{\sum_{n \in N} (Sim(c, n) \cdot r(n, a))}{|N|}. \quad (4)$$

Finally, apps were sorted by their interest and the top-k apps were provided as recommendations.

3.2. Dataset

During the period of the experiment, from 15 October 2018 to 1 December 2018, we recorded the following information, which we used to later evaluate the system and report the results:

- The recommender (popular, collaborative filtering, or random) assigned to each child.
- The recommendations generated, and also which recommender provided each of these recommendations.
- The date and time at which each recommendation is generated.
- The games usage per child, including the times at which they play and the duration for each game.

It is worth noting that regarding the second aspect, recommendations might not always be generated using the desired recommender. For example, a child might be assigned the collaborative filtering strategy, but this recommender might not have enough information about usage as to generate useful recommendations, or that all of these recommendations are filtered out. In that case, random recommendations are provided instead.

With this data, we can explore the patterns of engagement of children with the different games depending on whether those games were recommended or not.

3.3. Performance Metrics

The impact of the position in the ribbon of recommendations was be analyzed using click-through frequencies. That way, we will determined whether the first five visible items got more clicks than the last two, and whether the ordering within the visible and invisible ones mattered.

The core assessment of the recommendation algorithms will be made according to two engagement metrics. One based on the number of games and another one on game time.

The first one, the average number of games per user (ANG), is formally defined in Equation (5).

$$ANG = \frac{\sum_{i=1}^{NumUsers_R} Games_{Ri}}{NumUsers_R}, \tag{5}$$

where $Games_{Ri}$ is the number of games played by user i on apps recommended by algorithm R (either collaborative filtering, popular or random, and $NumUsers_R$ is the total number of users who were recommended the apps by algorithm R and acted on it.

The second aspect of engagement to be measured was the average game time (AGT) by users who acted on the recommendations. The expression used to compute the indicator is described in Equation (6).

$$AGT = \frac{\sum_{i=1}^{NumUsers_R} GameTime_{Ri}}{NumUsers_R}, \tag{6}$$

where $GameTime_{Ri}$ is the total time spent by user i playing apps recommended by algorithm R (either collaborative filtering, popular or random, and $NumUsers_R$ is the total number of users who were recommended the apps by algorithm R and used them.

We should note that, for the computation of these metrics, we define "game" as the event where the user interacts with the application for 10 s or more. Given the presence of some outliers, we also filtered out games of more than 3000 s and the instances where a game was played more than 60 times by the same user. These accounted for less than 0.5% of the sample.

In order to provide a more complete picture, in addition to these engagement metrics, we reported four standard performance indicators: accuracy, precision, recall and F1 score.

To compute these metrics we have proceeded as follows: we have divided the pilots period, comprising a total of 45 days, into two different time spans. The first one comprised one month (from 15 October 2018 to 14 November 2018) and the second comprised 15 days (from 15 November 2018 to 1 December 2018). The former was used as the training set while the latter was used for validation purposes.

We have considered the apps suggested by the recommender during the training phase to each child and have then matched this information with the actual apps played by children during the testing period. With this information, we have been able to build a confusion matrix as follows: apps recommended to a child during training and then played during testing constituted a true positive (TP), apps recommended but not played counted as false positives (FP), apps played but not previously recommended were considered false negatives (FN) and apps not recommended and not played constituted the true negatives (TN). All confusion matrices for each child were summed up for each recommendation strategy, and then accuracy, precision, recall and F1 score were computed following their standard definitions, which are shown in Equations (7)–(10).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{7}$$

$$Precision = \frac{TP}{TP + FP}, \tag{8}$$

$$Recall = \frac{TP}{TP + FN}, \tag{9}$$

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (10)$$

4. Results and Discussion

As it was mentioned before, the first part of the study has to do with the analysis of the impact of the position in the recommendation ribbon on the click-through. During the pilots period we have measured a total of 30,516 clicks received by 89 different apps, which have been involved in a total of 472,498 games. The experimental results on this aspect are reported in Table 2. There, we can see the total number of clicks over the relevant period by position for the apps that were recommended in all the possible slots.

Table 2. Accumulated click-through by position in the recommendation ribbon and mean for the visible and invisible positions for the period from 15 October 2018 to 1 December 2018.

Position	Visible					Hidden	
	1	**2**	**3**	**4**	**5**	**6**	**7**
Clicks	6329	5834	4516	2639	3960	3150	2937
Mean	-	-	4830	-	-	3044	-
Rank	1	2	3	7	4	5	6

It is apparent that the first three positions grabbed much more interest than the rest. The difference between the first two and the other five were specially sizable. While it was true that the fourth received fewer clicks than the ones that follow, other than that, the results were consistent with the existence of a direct relationship between the order and the number of times that users acted on the recommendations.

If we consider visibility, the ribbon only showed five recommendations at a time. We expected that to be a relevant factor, as getting to the last two requires a supplementary effort from the user. Interestingly, even though the average number of clicks on the visible slots was 4830, higher than the 3044 average clicks on the hidden ones, the role of friction as an element that drags click-through down could be questioned. If we consider the effect of position, where apps on the left hand side gather more interest that those on the right, and we infer the trend line, the recommendations on the sixth and seventh positions get more clicks than expected.

Regarding engagement, a total of 12,229 games were completed in apps that were being recommended at the moment by a total of 1387 children, summing up a total of 598 gaming hours. From the study of engagement, we find that the recommender algorithm based on popularity was superior across metrics. As we can see in Table 3, it outperformed the other two both in terms of number of games and accumulated use time by user. Unexpectedly, the proposed collaborative filtering algorithm resulted in a slightly lower mean number games vs. the random alternative. The sign of this difference, however, was the opposite for game time.

Table 3. Engagement metrics for game apps by recommender for the period from 15 October 2018 to 1 December 2018.

Engagement Metric		Mean	Median	Variance
Games				
	NG_{Random}	4.79	3	34.22
	$NG_{Popular}$	6.72	4	53.89
	NG_{CF}	4.22	2	24.75
Time				
	GT_{Random}	740.98	405.26	1,084,308.47
	$GT_{Popular}$	1202.76	639.79	3,286,849.96
	GT_{CF}	776.01	295.37	5,020,886.62

The statistical significance of the differences reported in Table 3 was assessed according to the protocol that follows. First, we started testing the normality of the distribution of the engagement metrics with Kolmogorov–Smirnov test with Lilliefors correction. In case normality was rejected, we applied Wilcoxon's test. Otherwise, we tested for the presence homoskedasticity using Levene test and, based on the result, we relied either on Welch test, or the traditional t-test. The results were analogous for the metrics. The superiority of the Popular algorithm over the other two was significant at 1%. Regarding the comparison of the baseline vs the implementation of collaborative filtering, the null hypothesis of equality could not be rejected at the 5% conventional level for neither the number of games nor the total game time.

In regards to the standard performance metrics, we report the main ones (accuracy, precision, recall and F1 Score) in Table 4. As we can see, recommendations based on popularity offered the largest precision and F1 score, while collaborative filtering and random recommendations provided the highest values in terms of accuracy and recall, respectively.

Table 4. Performance metrics for game apps by recommender algorithm. Train: 15 October 2018 to 14 November 2018. Test: 15 November 2018 to 1 December 2018.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Col. Filtering	94.92	7.40	1.28	2.18
Popular	94.73	12.93	3.30	5.25
Random	93.51	6.64	3.57	4.64

These results are influenced by the nature of the algorithms and the variety or recommendations that they generate. At the ends of the spectrum we would have random recommendation and recommendations based on popularity. The former approach is less constrained than the other two and, as a result, it foments exploration to a higher extent. Among the remainder, even though both are more stable, collaborative filtering provides a richer range of alternatives, as unguided exploration by peers is likely to result in new app recommendations.

A second aspect to be considered regarding the performance indicators is that, unlike the engagement ones, they only provide information on whether the user felt compelled to try new apps, not whether the user liked them. We are more interested in discovering this latter extent, which we find more important in the scope of this study. For this reason, we find that the average number of games and average time spent playing the recommended apps are the key indicators to consider, whereas standard performance metrics can be useful supplemental information to gain a better understanding of the situation.

To summarize the results, the two elements that constituted the subject of study of the recommender system were the influence of the order of recommendation on user exploratory behavior and the impact of the choice among two well-established recommendation algorithms, pre-selected by the company, on engagement. In order to evaluate such aspects, we acquired data of real interaction with the application between 15 October 2018 and 1 December 2018, running an A/B/C test with three different implementations of a recommender system (including a random baseline) and measuring clicks made on recommendations.

We found a direct association between the order in the recommendation ribbon and the number of clicks. Apps on the left gather more interest that those on the right hand side. The friction introduced by the fact that reaching the last recommendations requires either swiping or pressing on the arrows on the side does not seem to have a negative impact.

The A/B/C testing analysis used to compare the two recommendation approaches, one that recommends apps based on popularity and an implementation of collaborative filtering, vs the random recommender used as baseline offered two main results in regards to engagement: The first one is that the popular algorithm beats the other two both in terms of number of games and the total time spent playing the recommended contents. The second is that the current implementation of collaborative

filtering does not seem to add value, as it offers the same performance as the baseline. A likely explanation to this outcome is that there are some games that due to their nature are very likeable, and for this reason they end up being "the popular ones". These games will end up being suggested by the recommender based on popularity, and if children start to play them, in most cases they will quickly become engaged. This process could be adding some bias in the computed engagement metrics in favor of the popular recommendation strategy. If we consider performance indicators, the approach based on popularity seems to be the best alternative in terms of precision and $F1$ score, but collaborative filtering offers the highest accuracy.

5. Summary and Conclusions

In this paper we have described and evaluated the behavior of a recommender system in the scope of an Ed-Tech application aimed at children aged 2–10 and their families. The smart library gathers thousands of data points based on user interaction and uses that information to generate tailored reports and recommendations. The latter aspect is managed by a recommender system designed for the purpose of easing navigation through the library and enhancing exploration. We evaluated two aspects of the system: the influence of the order of recommendations on user exploratory behavior, and the impact of the choice of the recommendation algorithm on engagement.

The analysis led to the following conclusions: first, the order in the recommendation ribbon matter. Apps in the left positions got more clicks, and the effort to access hidden recommendations did not seem to have a negative impact. The second one is that the popular algorithm resulted in more engagement than the other two. Finally, the current implementation of collaborative filtering does not seem to add value. Future works would include thorough studies on alternatives to the current implementation of collaborative filtering or the optimization of its parameters; studying new possibilities based on the interface to drive the attention of the user to the recommended applications; or the implementation of recommendation strategies based on competences aimed at fostering the development of the user according to predefined preferences established by parents and educators.

Author Contributions: D.Q. surveyed the state of the art; A.B. and D.Q. conceived and designed the experiments; A.B. performed the experiments, A.B. and D.Q. analyzed the data; A.B. and D.Q. wrote the paper.

Funding: This research has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 756826.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gardner, H. *Multiple Intelligences*; Basic Books: New York, NY, USA, 1983.
2. Jäschke, R.; Marinho, L.; Hotho, A.; Schmidt-Thieme, L.; Stumme, G. Tag Recommendations in Folksonomies. In *Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, 17–21 September 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 506–514.
3. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative Filtering Recommender Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324.
4. Basilico, J.; Hofmann, T. Unifying Collaborative and Content-based Filtering. In Proceedings of the Twenty-first International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 9.
5. Niemann, K.; Wolpers, M. Usage Context-Boosted Filtering for Recommender Systems in TEL. In *Scaling up Learning for Sustained Impact: 8th European Conference, on Technology Enhanced Learning, EC-TEL 2013, Paphos, Cyprus, 17–21 September 2013.*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8095, pp. 246–259.
6. Rolando Núñez-Valdez, E.; Cueva Lovelle, J.M.; Infante Hernández, G.; Juan Fuente, A.; Labra-Gayo, J.E. Creating recommendations on electronic books: A collaborative learning implicit approach. *Comput. Hum. Behav.* **2015**, *51*, 1320–1330. [CrossRef]

7. Ruiz-Iniesta, A.; Melgar, L.; Baldominos, A.; Quintana, D. Improving Children's Experience on a Mobile EdTech Platform through a Recommender System. *Mob. Inf. Syst.* **2018**, *2018*, 1–8. [CrossRef]
8. Bhattacharya, S.; Nath, S. Intelligent e-Learning Systems: An Educational Paradigm Shift. *Int. J. Interact. Multimed. Artif. Intell.* **2016**, *4*, 83–88. [CrossRef]
9. Charleer, S.; Moere, A.V.; Klerkx, J.; Verbert, K.; De Laet, T. Learning Analytics Dashboards to Support Adviser-Student Dialogue. *IEEE Trans. Learn. Technol.* **2017**, *11*, 389–399. [CrossRef]
10. de Lange, P.; Neumann, A.T.; Nicolaescu, P.; Klamma, R. An Integrated Learning Analytics Approach for Virtual Vocational Training Centers. *Int. J. Interact. Multimed. Artif. Intell.* **2018**, *5*, 32–38. [CrossRef]
11. Käser, T.; Klingler, S.; Schwing, A.G.; Gross, M. Dynamic Bayesian Networks for Student Modeling. *IEEE Trans. Learn. Technol.* **2017**, *10*, 450–462. [CrossRef]
12. Hamoud, A.K.; Hashim, A.S.; Awadh, W.A. Predicting Student Performance in Higher Education Institutions Using Decision Tree Analysis. *Int. J. Interact. Multimed. Artif. Intell.* **2018**, *5*, 26–31. [CrossRef]
13. Ueno, M.; Miyazawa, Y. IRT-based adaptive hints to scaffold learning in programming. *IEEE Trans. Learn. Technol.* **2017**. [CrossRef]
14. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender Systems Survey. *Knowl.-Based Syst.* **2013**, *46*, 109–132. [CrossRef]
15. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2015**, *74*, 12–32. [CrossRef]
16. Drachsler, H.; Verbert, K.; Santos, O.C.; Manouselis, N., Panorama of Recommender Systems to Support Learning. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015; pp. 421–451.
17. Erdt, M.; Fernández, A.; Rensing, C. Evaluating Recommender Systems for Technology Enhanced Learning: A Quantitative Survey. *IEEE Trans. Learn. Technol.* **2015**, *8*, 326–344. [CrossRef]
18. Tarus, J.; Niu, Z. A survey of learner and researcher related challenges in e-learning recommender systems. *Commun. Comput. Inf. Sci.* **2017**, *734*, 122–132.
19. Lombardi, M.; Marani, A. A comparative framework to evaluate recommender systems in technology enhanced learning: A case study. *Lecture Notes Comput. Sci.* **2015**, *9414*, 155–170.
20. Kopeinik, S.; Kowald, D.; Lex, E. Which Algorithms Suit Which Learning Environments? A Comparative Study of Recommender Systems in TEL. In *Adaptive and Adaptable Learning*; Springer: Berlin, Germany, 2016; Volume 9891, pp. 124–138.
21. Kowald, D.; Kopeinik, S.; Seitlinger, P.; Ley, T.; Albert, D.; Trattner, C. Refining Frequency-Based Tag Reuse Predictions by Means of Time and Semantic Context. In *Mining, Modeling, and Recommending 'Things' in Social Media: 4th International Workshops, Prague, Czech Republic, 23 September 2013, and MSM 2013, Paris, France, 1 May 2013, Revised Selected Papers*; Springer International Publishing: Cham, Switzerland, 2015; Volume 8940, pp. 55–74.
22. Love, B.; Medin, D.; Gureckis, T. SUSTAIN: A Network Model of Category Learning. *Psychol. Rev.* **2004**, *111*, 309–332. [CrossRef]
23. Portugal, I.; Alencar, P.; Cowan, D. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Syst. Appl.* **2018**, *97*, 205–227. [CrossRef]
24. Drachsler, H.; Hummel, H.G.K.; Koper, R. Personal Recommender Systems for Learners in Lifelong Learning Networks: the Requirements, Techniques and Model. *Int. J. Learn. Technol.* **2008**, *3*, 404–423. [CrossRef]
25. Manouselis, N.; Drachsler, H.; Vuorikari, R.; Hummel, H.; Koper, R., Recommender Systems in Technology Enhanced Learning. In *Recommender Systems Handbook*; Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., Eds.; Springer: Boston, MA, USA, 2011; pp. 387–415.
26. Adomavicius, G.; 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.* **2005**, *17*, 734–749. [CrossRef]
27. Aamir, M.; Bhusry, M. Recommendation System: State of the Art Approach. *Int. J. Comput. Appl.* **2015**, *120*, 25–32. [CrossRef]
28. Rodríguez, P.A.; Ovalle, D.A.; Duque, N.D. A Student-Centered Hybrid Recommender System to Provide Relevant Learning Objects from Repositories. In *Learning and Collaboration Technologies*; Zaphiris, P., Ioannou, A., Eds.; Springer: Berlin, Germany, 2015; Volume 9192, pp. 291–300.
29. Salehi, M.; Kmalabadi, I.N. A Hybrid Attribute—Based Recommender System for E-learning Material Recommendation. *IERI Procedia* **2012**, *2*, 565–570. [CrossRef]

30. Bourkoukou, O.; El Bachari, E. Toward a Hybrid Recommender System for E-learning Personnalization Based on Data Mining Techniques. *JOIV Int. J. Inform. Vis.* **2018**, *2*. [CrossRef]
31. Nafea, S.; Siewe, F.; He, Y. ULEARN: Personalized Course Learning Objects Based on Hybrid Recommendation Approach. *Int. J. Inf. Educ. Technol.* **2018**, *8*, 842–847. [CrossRef]
32. Sears, A.; Jacko, J.A. *Human-Computer Interaction: Designing for Diverse Users And Domains*; CRC Press: Boca Raton, FL, USA, 2009.
33. Bjorklund, D.; Causey, K. *Children's Thinking: Cognitive Development and Individual Differences*; SAGE Publications: Newcastle upon Tyne, UK, 2017.
34. Wu, K.C.; Tang, Y.M.; Tsai, C.Y. Graphical interface design for children seeking information in a digital library. *Vis. Eng.* **2014**, *2*, 5. [CrossRef]
35. Pera, M.S.; Ng, Y.K. Automating Readers' Advisory to Make Book Recommendations for K-12 Readers. In Proceedings of the 8th ACM Conference on Recommender Systems, Silicon Valley, CA, USA, 6–10 October 2014; pp. 9–16.
36. Pera, M.S.; Fails, J.A.; Gelsomini, M.; Garzotto, F. Building Community: Report on KidRec Workshop on Children and Recommender Systems at RecSys 2017. *SIGIR Forum* **2018**, *52*, 153–161. [CrossRef]
37. Deldjoo, Y.; Frà, C.; Valla, M.; Paladini, A.; Anghileri, D.; Tuncil, M.A.; Garzotta, F.; Cremonesi, P. Enhancing Children's Experience with Recommendation Systems. In Proceedings of the 11th ACM Conference of Recommender Systems, Workshop on Children and Recommender Systems (KidRec'17), Como, Italy, 27–31 August 2017.
38. Kalloori, S.; Ricci, F.; Gennari, R. Eliciting Pairwise Preferences in Recommender Systems. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2 October 2018; pp. 329–337.
39. Hidasi, B.; Karatzoglou, A. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 843–852.