



HAL
open science

Deep Recurrent Neural Networks for OYO Hotels Recommendation

Anshul Rankawat, Rahul Kumar, Arun Kumar

► **To cite this version:**

Anshul Rankawat, Rahul Kumar, Arun Kumar. Deep Recurrent Neural Networks for OYO Hotels Recommendation. 18th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2022, Hersonissos, Greece. pp.245-256, 10.1007/978-3-031-08333-4_20 . hal-04317174

HAL Id: hal-04317174

<https://inria.hal.science/hal-04317174v1>

Submitted on 1 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Deep Recurrent Neural Networks for OYO Hotels Recommendation

Anshul Rankawat^{1*}, Rahul Kumar¹, and Arun Kumar²

OYO HOTELS AND HOMES PVT LTD, Gurgaon, India

^{1*}anshul.rankawat@oyorooms.com, ¹rahul.kumar26@oyorooms.com,

²arun.kumar12@oyorooms.com

Abstract. Recommendation Systems at OYO solve a complex personalization problem with scale and sophistication. The authors have focused on the development of the best-in-class recommendation system in the hospitality industry using a deep learning based model. The objective of the work is to develop a recommendation model which uses sequences of user interactions with contents derived from user interactions. The hybrid model described in the paper is a deep recurrent neural network based architecture split into two components: first, an embedding generation model and then a deep prediction and ranking model. The models have shown significant performance improvement both online and offline over existing collaborative filtering based models across geographies irrespective of traffic density and hotel supply density. The success of the deep learning based hybrid recommendation model at OYO across different geographies indicates immense potential of such recommender systems in industries such as travel, hospitality etc.

Keywords : Neural Network, Deep Learning, RNN, GRU, LSTM, BiLSTM, Graph-based Model, CTR, Conversion, Realization, CxR, Embeddings, Hit Ratio, Mean Reciprocal Rank

1 Introduction

While owning a large inventory is a great benefit for OYO's customers as it allows OYO to offer plenty of opportunities for finding a perfect place to stay, this presents a great challenge for customers by overburdening them with choices. Search and Ranking system at OYO partially solves the problem by listing properties for searched locations along with filters available but it still lacks in serving the most relevant and personalized properties to the users which results in significant drop-offs in the funnel.

Recommendation system is one of the most important traffic driving widgets at OYO App which drives traffic on the home page directly to the detail pages of the most relevant properties, customers may be the most interested in and thereby reducing friction points significantly by short-circuiting the funnel.

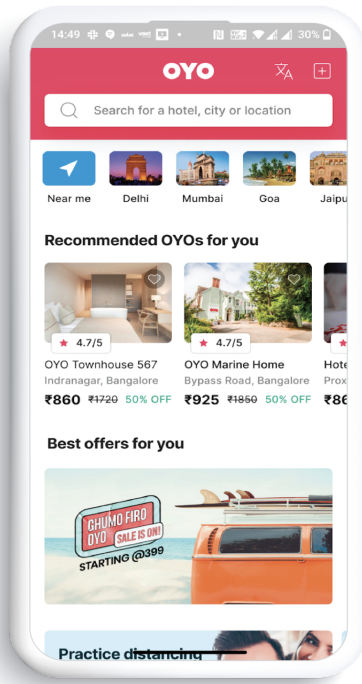


Figure 1: Recommendation widget displayed on OYO mobile App home page

There are various algorithms used in the development of recommender systems each with certain advantages over others. The authors in this paper have explored numerous variants of deep learning based models which use hybrid methods of content and collaborative filtering. The approach, implemented on the OYO App now, uses deep learning models on huge amounts of data of customer & property interactions to generate implicit embedding of properties and user interests.

2 Literature Review

Recommendation models are predominantly categorized into collaborative filtering recommender systems, content-based recommender systems and hybrid recommender systems based on the different types of input data [1]. Collaborative filtering generates personalized recommendations about a user's preference based upon the collective preferences of many other users by learning from past user-item interactions. Among numerous collaborative filtering methods, Matrix Factorization is widely used across industries, which projects users and items into a common vector space and tries to capture a user's preference on an item by the inner product between their vectors [13, 14, 15]. Content-based recommendations work by creating the user-item profiles using explicit (e.g. user's previous ratings) or implicit data (e.g. browsing history). The content here refers to the attributes of the item in which a user has shown interest in the past. Content-based recommendation systems do not take into account the behavior of the other users in the system. Hybrid recommendation systems appropriately make

use of both of the above types of recommendations [11]. Recently, a number of findings in traditional recommender systems have been presented. For example, Su et al. [7] showcased a systematic review on collaborative filtering techniques; Burke et al. [8] demonstrate a comprehensive survey on hybrid recommender system; Fernandez-Tobias et al. [9] and Khan et al. [10] reviewed the cross-domain recommendation models. However, researchers have started using deep learning for recommendation systems since 2015, which seems pretty recent but it's a long time in the purview of current AI development.

The demand and acceptance of deep learning models have been increasing exponentially in the online industry due to their ability to solve complex problems with state-of-the-art results on a high volume of real-time data [2]. Contrary to linear models, a deep learning model based recommendation model can capture the non-linear user-item relationships and can generate high quality recommendations for users. Conventional methods such as matrix factorization models the user-item interaction based on the linear combination of user and item latent factors [12]. Another advantage of deep learning based recommendation systems is that it significantly reduces the efforts in creating explicit tailor-made features by automated feature learning from raw data in an unsupervised or supervised approach. Furthermore, deep learning models like RNN also facilitate the mining of temporal behavior of user-item interactions which plays an vital role in generating session based sequential recommendation in the hospitality industry. Unfortunately, none of the conventional methods are for sequential recommendation since they do not tend to capture the inherent order in a user's browsing behavior.

There are many recent examples which substantiate the pervasiveness of recommendation systems in improving user experience and promoting sales for many websites and mobile applications. Researchers have shown that 80 percent of movies watched on Netflix came from recommendations [3] and 60 percent of video clicks on Youtube came from homepage recommendations [4]. Recently, RNN and its variants, Long Short-Term Memory (LSTM) [16] and Gated Recurrent Unit (GRU) [17] have revolutionized the recommender systems by modeling the user behavior sequences [18, 19, 20, 21, 22, 23, 24]. The basic purpose of these methods is to represent the user's temporal interactions as a vector with multiple recurrent architectures. By doing that, it can pass the relevant information down to the deep chain of sequences to generate predictions. Cheng et al. [5] presented an App recommender system for Google Play with a deep and wide model. Shumpei et al. [6] proposed a RNN based news recommender system for Yahoo News. All of these models have been passed through successful experimentation and have outperformed the traditional recommendation models which corroborates the ubiquity of deep learning based recommender systems as industrial applications. In the hospitality industry, Garboviv et al. [26] used the skip gram model to learn embedding for vacation rental properties which captures the aspects of user interest, hotel geographic information, hotel attributes, and so on, as latent variables. However, their model does not use any information apart from the click data. To address this shortcoming, we proposed adding more explicit information about the hotel which is derived from aggregated customer behavior towards the hotel in terms of the hotel average monthly rating and the hotel realization values.

3 Methodology

A recommendation system starts with some relevant information about every user, their interactions with our App and uses the information to identify his/her interests. Then it merges this information with the collective behavior of all other consumers who have similar preferences to recommend stuff that the consumer may like.

OYO App has been serving personalized recommendations to its visitors through a collaborative filtering based robust recommendation system which is optimized on browsing data as user's feedback. However, the newly developed model is a hybrid model which uses sequences of user interactions with contents derived from user interactions. Hence, the modeling framework is divided into two major components - i) Embedding Generation Model, ii) Prediction and Ranking Model. Figure 2 illustrates this hybrid architecture of the Embedding Generation Model, and the Prediction and Ranking Model. For every user who had visited the OYO app, the sequence of their browsed hotels and embeddings (generated by Embedding Generation Model) along with the rating tokens and the realization tokens for those hotels are passed into the Prediction and Ranking Model to identify top N hotel recommendations which are subsequently, shown on the OYO app for the users who are viewing the recommendation widget.

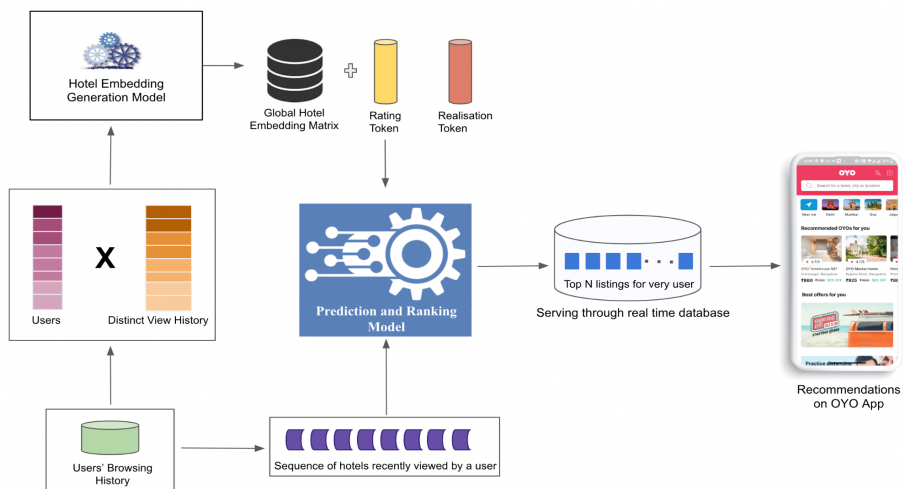


Figure 2: Recommendation Model Schema demonstrating generation of personalized ranking for every user on OYO App

3.1 Embedding Generation Model

The content can be derived either in the form of explicit feedback (e.g. users' previous ratings) or implicit feedback (e.g. browsing history). The explicit feedback is generated by explicitly asking the consumers about their interest in terms of rating. This feedback is then used to build up a profile of those consumers on various

interests. However, the problem with this approach is that it requires extra effort from the consumers. Moreover, not everyone is bothered to leave a rating on everything they see on the platform. Hence this data tends to be very sparse which generally leads to low quality recommendations in conventional methods. Another problem with explicit feedback is that the human perception to rate things differs from individual to individual. The same hotel might be rated differently because customers' preference of parameters differs. If a customer is very particular about cleanliness, a hotel not so clean but very good at other parameters like locations, other amenities etc will still be rated low.

On the other hand, the implicit feedback is generated directly by capturing consumers' behavior through browsing data and/or purchases history. It tends to be more robust because one need not rely on consumers' interaction with the platform for a voluntarily given explicit rating. The predecessor to the recommendation model described in this work was a graph-based model which is optimized on browsing data as user's feedback. Browsing data solves the problem of data sparsity as it contains plenty of information about the consumers but they aren't always reliable in terms of consumers interests because consumers can often click on platforms by accident. Browsing data is also highly susceptible to fraud because there are a lot of bots present on the internet to perform redundant clicks and hamper the quality of data. Hence, using purchases as the implicit positive feedback can be more effective in generating higher quality recommendations for a consumer.

In this work, we derived implicit features using the Embedding Generation Model which is an unsupervised recurrent neural network type of deep learning model. These latent features, along with the hotel average monthly rating and the hotel realization values, act as a precursor to the hotel prediction and ranking model as shown in Figure3.

3.2 Prediction and Ranking Model

To understand customers' interactions based on the collective preference of customers, we used hotels in the sequence of visits by a user along with embeddings i.e. implicit features of those hotels. The predecessor model implemented at OYO was a graph-based collaborative filtering model optimized for Click-Through-Rate (CTR) whereas the proposed recommendation model is optimized for realized bookings (Conversion along with Realization of bookings).

We implemented four sequential deep learning architectures (akin to collaborative filtering) i) basic RNN, ii) GRU, iii) LSTM and iv) BiLSTM to identify the most effective model and compare with the predecessor model. Recurrent Neural Networks (RNN) are the type of neural networks that were designed to work with sequential data. In the hospitality industry, the sequential data can be in the form of user reviews, image sequences, browsing history etc. However, during back propagation, RNNs can suffer from having a short-term memory due to vanishing gradient problems. Gradients are values that are used to update weights of the layers within a neural network. The vanishing gradient problem signifies that the gradient value shrinks as it back-propagates through time. If a gradient value becomes extremely small, it does not help in too much learning. Hence, the layers that get a small gradient update tend to stop learning and the model forgets the information on the longer sequences.

Researchers have developed GRUs [17], LSTMs [16] and BiLSTMs [25] as the solution to short-term memory. These networks have internal mechanisms called gates that can regulate the flow of information. These gates can learn which data in a sequence is important to keep or throw away. By doing that, it allows relevant information to pass through the long chain of sequences to generate predictions. The layers in GRUs and LSTMs try to preserve the information of the past because the only inputs they have seen are from the past whereas the layers in bidirectional LSTMs (BiLSTM) set up use the the two hidden states combined in any point of time and preserves the information from both past and future.

To train the model, we created a training set of one million users where the sequence of their browsed hotels within a click session acts as an input. Formally, a click session is defined as a sequence of hotels clicked on by a user during a defined window of time or visit. The input sequence is then padded and limited to 15 hotels because the weights in the sequential models can blow up quickly if larger numbers of hotels are taken. Since the neural network has to back propagate through time, an upper bound is required on how many time steps it needs to back propagate to. In this way, we truncate our backpropagation through time, which helps in reducing the time complexity of the model significantly. Our aim was to identify the affinity of a user towards realized booking at high rated hotels in the form of probability which we get as an output from the model. We identified the top N hotels as a list of recommendations for the user based on their probability of realized booking for high rated hotels.

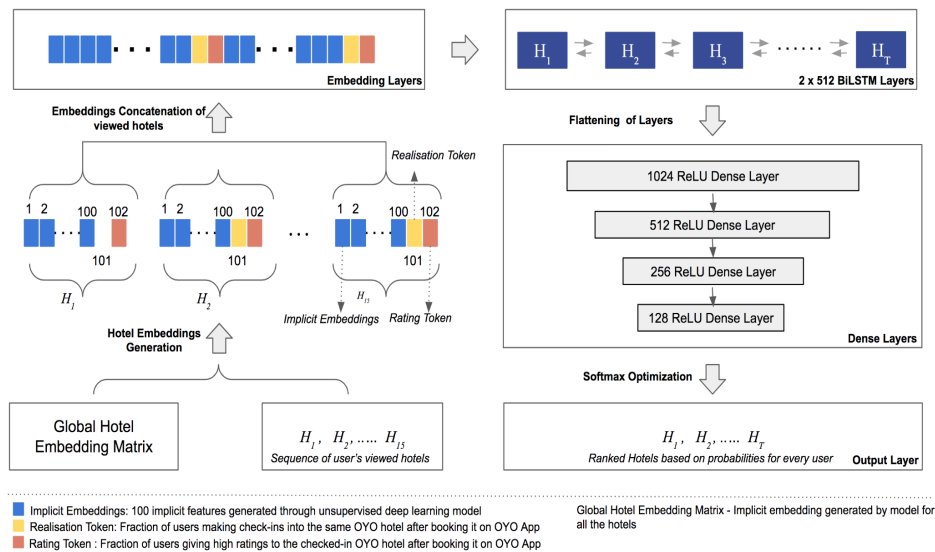


Figure 3: Deep RNN based Prediction and Ranking Model Architecture

Figure 3 illustrates the proposed architecture for an enriched prediction and ranking model. As we can see, each aspect of the hotel is embedded separately, and these representations are later concatenated and further compressed through BiLSTM layers followed by four dense layers before being used for hotel predictions.

4 Performance Evaluation

We first attempted to evaluate performance of embeddings generated from unsupervised learning and then evaluated the ranking list of hotels as our final output generated from supervised learning.

4.1 Embedding Generation Model Evaluation

There is no direct quantitative method to evaluate embeddings holistically. The embedding generation model here provides the vector of length 100 as implicit features for each hotel. We used a cosine similarity index across parameters to measure similarity between hotels and identified the list of top 10 similar hotels corresponding to every hotel in the modeling dataset. Once the list of top 10 similar hotels is generated, we employ four accuracy metrics based on parameters such as location, distance, price and ratings of hotel to evaluate the performance of the embeddings quantitatively. Location, distance, price and ratings are the most important selection criteria for hotel bookings. The four custom metrics are generalized as follows:

$$\text{Sim_Index}@x = \frac{\sum_{i=1}^H (\text{Sim}_{x,i})}{H}$$

where x = Price, Location, Ratings and Distance; H = Total number of hotels; $\text{Sim}_{x,i}$ = fraction of top 10 hotels similar to the target hotel which fulfill criteria of similarity across parameters x . For example - $\text{Sim}_{\text{Price},2}$ means a fraction of the top 10 similar hotels identified for the second target hotel which are within +/- 15% of the price of that target hotel. The criteria for other parameters are as follows: For Distance, it is less than 20 km from the target hotel; for Ratings, it is +/-1 rating from the target hotel's rating; and for location, the hotels should be in the same city of the target hotel.

Table 1: Parameter based Similarity Indices for Embedding Generation Models.

Embedding Models	Learning Epochs	Learning Rate	Number of Embeddings	Sim_Index @ Ratings	Sim_Index @ Price	Sim_Index @ Distance	Sim_Index @ Location
Embedding_v1	50	0.01	50	7.78	7.34	7.50	8.25
Embedding_v2	100	0.005	100	7.83	7.43	7.55	8.23
Embedding_v3	50	0.01	100	7.84	7.47	7.56	8.25
Embedding_v4	150	0.01	100	7.87	7.47	7.60	8.30
Embedding_v5	100	0.01	100	7.87	7.48	7.59	8.28
Embedding_v6	100	0.01	50	7.08	7.49	6.81	7.60
Embedding_v7	100	0.005	100	7.20	6.67	6.91	7.66
Embedding_v8	100	0.01	100	7.26	6.69	6.97	7.72

Based on these similarity indices, we compared the quality of eight variants of the embedding generation models and identified the best model with a sim index for ratings, price, distance and location standing at 7.87, 7.48, 7.59 and 8.28 respectively as shown in Table 1.

Furthermore, we visually evaluated top 10 similar hotels for random 30 hotels selected and observed explicit features like price, locations, ratings, brands etc. The embedding models successfully listed top 10 similar hotels in most of the cases. For example - In Figure 4, similar hotels generated for hotel A (in extreme left) have similar prices, locations, brands and ratings.

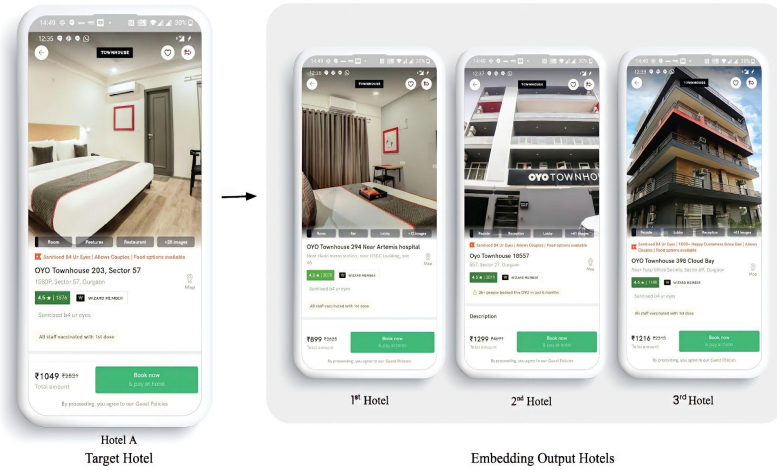


Figure 4: Visual evaluation of top 3 similar hotels identified for a target hotel using embeddings

4.2 Prediction and Ranking Model Evaluation

Recommendation model provides output in the form of a ranking list of hotels regardless of different modeling frameworks. We thoroughly evaluated the output offline as well as in real-time production setup.

4.2.1 Offline Evaluation

To evaluate the ranking list of hotels, it is always a better approach to directly evaluate the ranking quality instead of the other proxy metrics like mean squared error, etc. In this work, we report two accuracy metrics to evaluate our Prediction and Ranking Model performance.

4.2.1.1 Precision@k or Hit Ratio

In our recommendation setting, precision or hit ratio is defined as the fraction of users for which the booked hotel is included in the list of ordered recommendations having length k.

$$\text{Precision@k} = \frac{U_{\text{hit}}^k}{U_{\text{all}}}$$

where U_{hit}^k is the number of users for which the booked hotel is included in the top k recommendation list, U_{all} is the total number of users who booked a hotel in the test dataset.

4.2.1.2 Mean Reciprocal Rank (MRR)

For a given user, the Reciprocal Rank (RR) is defined as the reciprocal of the rank at which the first booked hotel is identified in the list of ordered recommendations. RR is 1 if a relevant hotel is identified at first position, it is 0.5 if the relevant hotel is identified at rank 2 and so on. When the average is determined across all the users, the measure is called the Mean Reciprocal Rank (MRR).

$$RR(u) = \sum_{i < k} \frac{relevance_i}{rank_i}$$

$$MRR = \frac{\sum_{u=1}^U (RR(u))}{U}$$

where RR is the reciprocal rank of user u which is the sum of the relevance score of top k hotels weighted by reciprocal rank and MRR is the mean of RR for all the users.

Table 2: Accuracy Metrics for Hotel Prediction and Ranking Model Variants

Models	Batch size	Embedding Training	#Dense Layers	Nodes per ReLu Dense Layers	Dense Epochs	Precision @20	MRR
Basic RNN v1	64	FALSE	2	512 :: 256	80	85.6%	72.0%
Basic RNN v2	128	TRUE	3	512 :: 256 :: 128	100	87.0%	72.4%
Basic RNN v3	256	TRUE	4	1024 :: 512 :: 256 :: 128	100	88.2%	72.8%
GRU v1	128	FALSE	2	256 :: 128	80	89.2%	72.4%
GRU v2	256	TRUE	2	512 :: 256	100	82.9%	75.2%
GRU v3	256	FALSE	3	512 :: 256 :: 128	60	71.6%	69.8%
GRU v4	512	TRUE	4	1024 :: 512 :: 256 :: 128	60	75.8%	73.9%
LSTM v1	128	FALSE	2	256 :: 128	80	68.0%	74.5%
LSTM v2	256	TRUE	2	512 :: 256	100	82.4%	75.1%
LSTM v3	256	TRUE	3	512 :: 256 :: 128	80	89.9%	74.6%
LSTM v4	512	TRUE	4	1024 :: 512 :: 256 :: 128	80	84.5%	75.7%
BiLSTM v1	128	FALSE	2	256 :: 128	80	84.9%	75.7%
BiLSTM v2	256	TRUE	2	512 :: 256	100	85.3%	75.7%
BiLSTM v3	256	TRUE	3	512 :: 256 :: 128	80	86.6%	75.0%
BiLSTM v4	512	TRUE	4	1024 :: 512 :: 256 :: 128	100	88.0%	75.8%

Based on Precision@20 and MRR, we compared variants of four sequential neural network architectures for sequential models as shown in Table 2 and identified the best model from each category i.e basic RNN v3, GRU v1, LSTM v3 and BiLSTM v4. Further, we performed the out of time validation with precision@2, 5, 10, 20 and MRR as evaluation metrics for these four best in category models along with the predecessor model, i.e. a graph-based model, to evaluate their robustness on a dataset that is more closely aligned with the real-time environment. The results suggested that the variant of BiLSTM model i.e BiLSTM v4 has outperformed all the other models with a precision of 62.5%, 55.4%, 48.8%, 41.5% at top 20, 10, 5 and 2 positions of recommendation respectively and with a MRR of 43.1% as shown in Table 3.

Table 3: Out of time model validation for top 4 model variants

Models	Precision@20	Precision@10	Precision@5	Precision@2	MRR
Graph Based Model	46.6%	41.2%	35.4%	27.4%	30.5%
Basic RNN v3	60.4%	53.0%	46.4%	39.8%	42.0%
GRU v1	59.7%	52.4%	45.9%	39.4%	41.7%
LSTM v3	62.6%	55.4%	48.6%	41.2%	43.1%
BiLSTM v4	62.5%	55.4%	48.8%	41.5%	43.1%

4.2.2 Online Evaluation

During the development phase, we made extensive use of offline metrics (precision@k, MRR etc) to improve the efficacy of our algorithm iteratively. However, for the final vetting of the effectiveness of an algorithm, we relied on real-time experiments via A/B testing because these results are not always correlated with offline evaluation. In a real time experiment, we can measure subtle differences in metrics like conversion, realization etc. that measure user affinity and satisfaction towards the hotel. In our experiment, OYO hotels were selected in a manner that ensured no other experiments were live on those properties during the same time period. User behavior was measured in terms of realized bookings or CxR (CxR is multiplication of booking conversion and realization of bookings) at high rated hotels. Lifts of 3% to 6% were observed in different geographies when compared with existing graph-based collaborative filtering models tested in A/B set ups.

5 Conclusion

With paradigm shift in machine learning algorithms, huge customer interactions on OYO App and improvement in computing power of machines, the recommendation systems on OYO App needed a change of approach for recommendation systems. The authors have tried multiple variants of deep learning based hybrid models to make the recommendation system more effective.

The hybrid model described in the paper is a deep recurrent neural network based architecture split into two components: i) Embedding Generation Model and ii) Hotel Prediction and Ranking Model. Evaluated in A/B testing and offline simulations, the deep learning based hybrid models have shown significant performance improvement in realized bookings and revenue over then implemented graph based collaborative filtering models. The consistent lifts of 3% to 6% in realized hotel bookings compared with previous recommendation models across geographies with different daily active users and hotel density shows robustness of the approach with hybrid deep learning based models.

The intermediate outputs like global hotel embedding matrix can be used in many other personalization solutions like similar hotels widgets on hotel details page, personalized campaign for similar hotels for the hotels viewed but not booked, personalized campaign for similar hotels for the hotels viewed but were sold out etc. The success of the deep learning based hybrid recommendation algorithms across different geographies with different traffic density and supply density demonstrate a great promise to be used in many other industries like Travel, Restaurant and other ecommerce industries.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17, 6 (2005), 734–749.
2. Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Recsys*. 191–198.
3. Carlos A Gomez-Uribe and Neil Hunt. 2016. The Netflix recommender system: Algorithms, business value, and innovation. *TMIS* 6, 4 (2016), 13.
4. James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The YouTube Video Recommendation System. In *Recsys*.
5. Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, and others. 2016. Wide & deep learning for recommender systems. In *Recsys*. 7–10.
6. Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based News Recommendation for Millions of Users. In *SIGKDD*.
7. Xiaoyuan Su and Taghi M Khoshgoor. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence 2009* (2009), 4.
8. Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.
9. Ignacio Fernandez-Tobias, Ivan Cantador, Marius Kaminskis, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*. 24.
10. Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross Domain Recommender Systems: A Systematic Literature Review. *ACM Comput. Surv.* 50, 3.
11. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. Recommender systems: An Introduction.
12. Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

13. Yehuda Koren and Robert Bell. 2011. Advances in Collaborative Filtering. In the Recommender Systems Handbook. Springer US, Boston, MA, 145–186.
14. Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37.
15. Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In Proceedings of NIPS. Curran Associates Inc., USA, 1257–1264.
16. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (Nov. 1997), 1735–1780.
17. Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of EMNLP. Association for Computational Linguistics, 1724–1734.
18. Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In Proceedings of RecSys. ACM, New York, NY, USA, 152–160.
19. Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In Proceedings of CIKM. ACM, New York, NY, USA, 843–852.
20. Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In Proceedings of ICLR.
21. Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In Proceedings of CIKM. ACM, New York, NY, USA, 1419–1428.
22. Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In Proceedings of RecSys. ACM, New York, NY, USA, 130–137.
23. Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks. In Proceedings of WSDM. ACM, New York, NY, USA, 495–503.
24. Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In Proceedings of SIGIR. ACM, New York, NY, USA, 729–732.
25. Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE transaction on signal processing* 45, 11 (1997), 2673–2681.
26. Grbovic, C. 2018. Real-time personalization using embeddings for search ranking at airbnb. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 311–320. KDD.