ACM RecSys Challenge 2019

Xianzhe Zhang
Stanford University
xianzhez@stanford.edu

Xiao Wang
Stanford University
xiao1105@stanford.edu

Jiaokun Liu
Stanford University
louisliu@stanford.edu

Mentor: Róbert Pálovics
Stanford University
palovics@stanford.edu

Abstract

The theme of the ACM RecSys Challenge 2019 is to develop a session-based and context-aware recommender system using the dataset from Trivago, which is a global hotel search platform. In this task, our goal is to predict which accommodations have been clicked in the search result during the last part of a user session. In this paper we present our approaches to this challenge. We extract features from training data and feed them into binary classification models, like XGBoost, Logistic Regression, etc. Besides basic models. We also try deep learning models like CNN to further explore item meta data. Our full model pipeline is a two-stage model combining binary classification and deep learning, and achieves 0.604 MRR score.

1 Problem Description

RecSys Challenge is an annual data science challenge for the ACM Recommender Systems conference, which gives anyone who's interested the chance to work on real-world data science problems and large data sets.

Trivago provides the dataset for the ACM Recommendation System Challenge 2019[8]. When a user visits the website of Trivago, a list of items will be shown to the user and all of the user's actions (e.g. filter usage, search refinements, item interactions, item searches, item click-outs) will be recorded as a session. Based on given training dataset, we are supposed to train a model using various extracted features to predict which item has been clicked in the given impression list.

2 Related Work

Latent Collaborative Filtering (CF) model Weighted Regularized Matrix Factorization (WRMF) [2] has proven to be an simple and effective model in recommendation system. By reducing the session data and extra users action, it is easy to build WRMF based on click-on information and user ID. In 2018 music recommendation challenge, the top performance team constructed complex vectors based on music information includes vocabulary, style, semantics and other well-chosen feature [4]. Another team utilized neighborhood-based to construct item-item and user-user matrix [3] and fed them directly to neural networks.

3 Data

3.1 Dataset Overview

In this problem, Trivago has released a public dataset of hotel search sessions. The dataset consists of a training set (train.csv) and test set (test.csv), and meta data (item_metadata.csv) for accommodations, i.e., items. The training set contains user actions up to a specified time (split date). The recommendations should be provided for a test set that contains information about sessions after the split date but do not include the information about the accommodations that have been clicked in the last part of the sessions. The required output is a list of maximum 25 items for each click-out ordered by preferences for the specific user. The following Figure 1 illustrates how the dataset was split and how sessions were recorded.

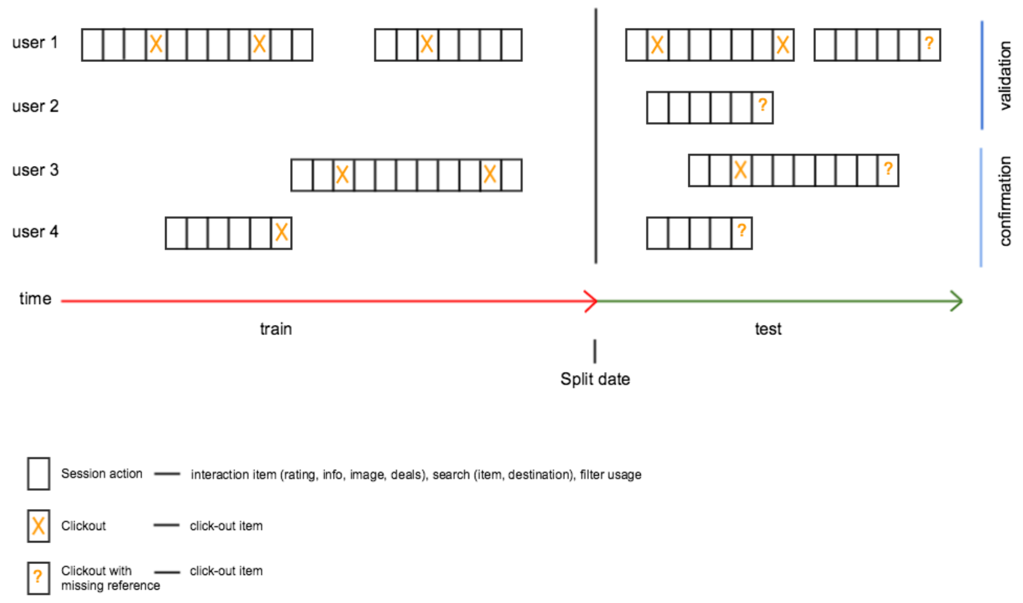

user 1
user 2
user 3
user 4
validation
confirmation
time
train
test
Split date
Session action — interaction item (rating, info, image, deals), search (item, destination), filter usage
Clickout — click-out item
Clickout with missing reference — click-out item

Figure 1: Illustration of the problem setting

Figure 10 (in appendix) displays an entire session action data for a specific user. The action types actually contain more information than the figure shows, including: click out item, interaction item rating, interaction item info, interaction item image, interaction item deals, change of sort order, filter selection, search for item, search for destination, search for point of interest. The training data can be used to build models of user interactions and specifies the type of action that has been performed (filter usage, search refinements, item interactions, item searches, item click-outs) as well as information about impressed items and prices at the time of a clickout item.

3.2 Statistics

In order to generate features, we dug into the dataset to get insights about the details, which could help us better understand the dataset and define our problem. We investigated various statistics of the whole train dataset. Some basic statistics is shown in Table 1. In the dataset, there are more than $15M$ records, $730K$ users and $826K$ sessions during the time span of 6 days. We found that some of the sessions have no clickout item (i.e. the item has been clicked by the user), and some of the sessions have more than one clickout item. Therefore, we plot a pie chart to visualize the distribution of clickout count in a session as shown in Figure 2. There are 60.8% of sessions with only one clickout, 20.0% of sessions with two clickouts, etc. We believe there are some correlations among the clickouts in the same session, especially for the session with more than one clickout. Hence, it would become one of our concentrations to extract features.

Table 1: Basic statistics

Item	Value	Description
User count	730, 803	The number of users.
Session count	826, 842	The number of sessions.
Sessions with clickout	98%	Percentage of sessions with clickout.
Clickout count	910, 683	The number of clickout.
Records	15, 932, 992	The number of records.
Time range	6 days	Time span from 2018.11.01 12:00:08 to 2018.11.06 23:59:59.

We plotted distribution of session time length, which shows heavily tailed effect (see Figure 3). The session time might reflect some information about the behavior of most people when they browse the website. We extracted some useful features about session time in the following steps.

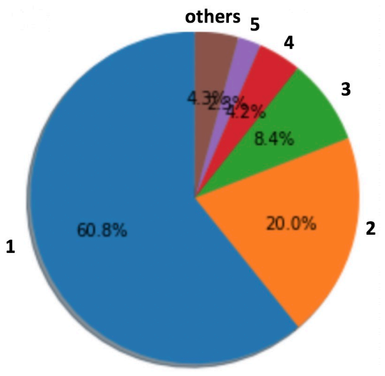


Figure 2: Clickout Number in a Session Distribution

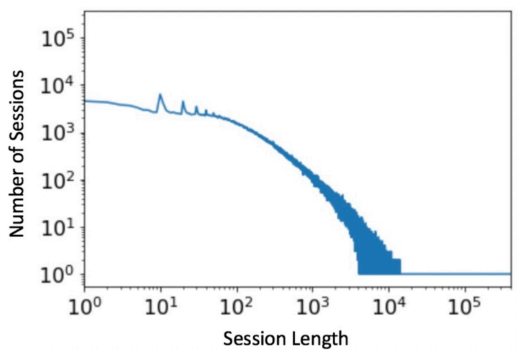


Figure 3: Session Length Distribution

Furthermore, item price is one of the important factors that could affect the choice of a user. We measured the price of clickout item and calculated their relative price in the impression list. The relative price is calculated by min-max normalization $P_{relative} = (P_i - P_{min})/(P_{max} - P_{min})$. The distribution is show in Figure 4. We found that most of the user would choose the cheaper one which make much sense. At the same time, we also found that the position of item in the impression list is also highly related with the clickout probability. Most of the clickout item is in the top 3 of the impression list as shown in Figure 5.

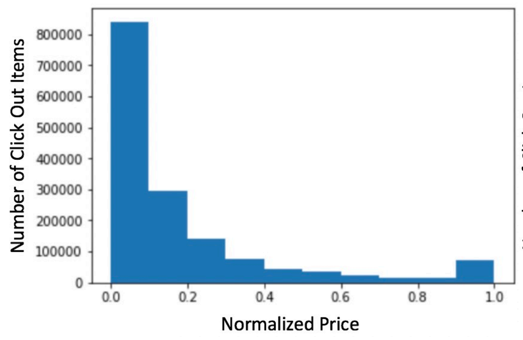


Figure 4: Normalized Price of Clickout Item Distribution

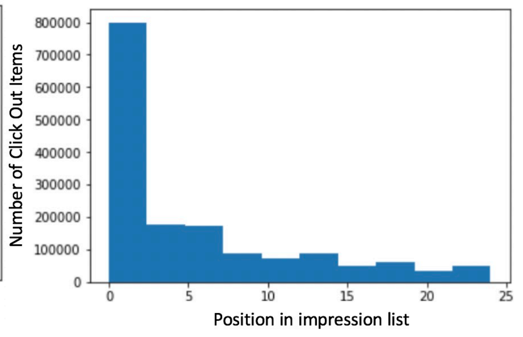


Figure 5: Position of Clickout Items Distribution

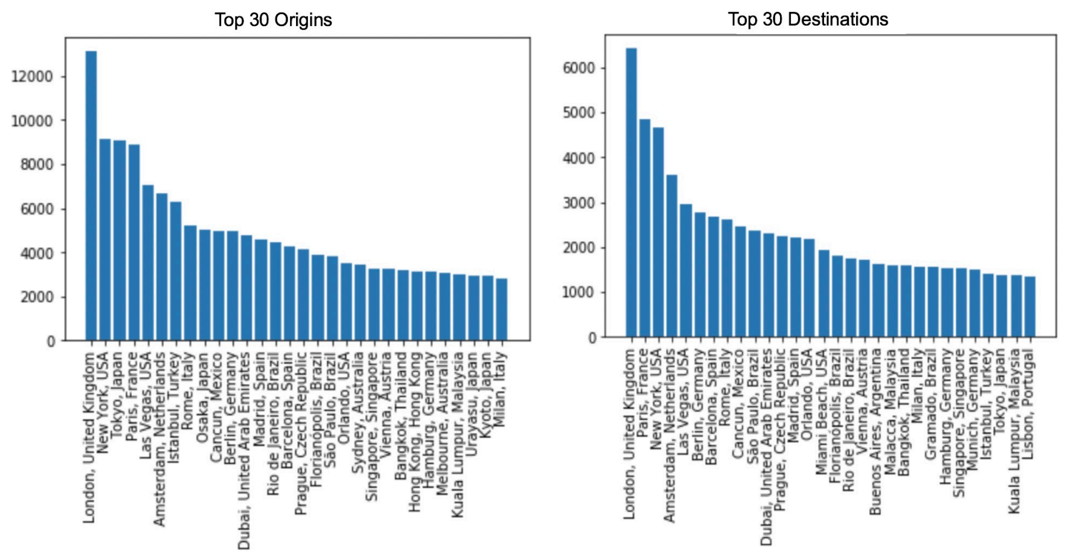


Figure 6: User Origin and Destination Distribution.

We investigated the position of users to explore the popularity of cities all over the world. Specifically, we explored the origin cities which are the cities that users are located and the destination cities that users are interested in. We plotted the top 30 cities separately as shown in Figure 6. London, New York, Tokyo are the top 3 cities that the most users are from. London, Paris, New York are the top 3 cities that users are most interested in.

We also explored lots of other properties and statistics of the dataset, such as the country distribution of users (Figure 11), interaction time and clikout (Figure 12), time distance between interaction and clickout (Figure 13), platfrom distribution (Figure 14), date distribution (Figure 15), etc. You could find more diagrams in appendix.

3.3 Feature set

We believe that an user's behavior is affected by item's own properties, user-item interaction and actions in the session. Therefore, we explore item properties features, session based features, user based features and some global features in this subsection.

3.3.1 Item properties features

We have 927,142 different items in the data set (item_metadata.csv) and each item has many properties. The total number of property categories in item data is 157. The top 10 properties are listed by the number of occurrences and by clickout item probability under each property separately. The clickout item probability under each property formula is listed below. From Table 2, we can see most items have properties like "Satisfactory Rating", "Car Park", "Good Rating" and etc. Under properties like "Water Slide", "5 Star", "Convention Hotel", items are more likely to be clicked out.

$$p(\text{clickout}|\text{property}) = \frac{p(\text{clickout})p(\text{property}|\text{clickout})}{p(\text{property})}$$

3.3.2 Session based features

Another kind of important features are session based features. We list the names and definitions of these features in Table3. Some of the important features here are "interact before", "position in the list" and "first interact". It indicates that whether the user interacts with the item before plays an important role in recommending the item. If the item is in the front of impression list or the item is interacted before, it is more likely to be clicked out.

Table 2: Important Item Properties

The number of occurrences	Property name	Click out item propability under each property	Property name
533286	Satisfactory Rating	0.747851003	Water Slide
487879	Car Park	0.723328658	5 Star
481910	Good Rating	0.695209835	Convention Hotel
467027	WiFi (Rooms)	0.691724437	Hydrotherapy
426875	Shower	0.676814597	Romantic
425953	Television	0.674660272	Hot Stone Massage
399547	WiFi (Public Areas)	0.670235158	Spa (Wellness Facility)
379321	Hotel	0.668779904	Health Retreat
376666	Very Good Rating	0.66874688	Pousada (BR)
353296	Air Conditioning	0.668447815	On-Site Boutique Shopping

Table 3: Session Features

Session based features	Definition
Dwell time (second)	For each item, the total time spent within the session
Interact before (1 or 0)	User interacted with the item before in another session
Interact image (1 or 0)	User interacted with the item image in the session
Interact info (1 or 0)	User interacted with the item info in the session
Search (1 or 0)	User searched for item in the session
Interact deals (1 or 0)	User interacted with the item deals in the session
Interact rating (1 or 0)	User interacted with the item rating in the session
Click out time (second)	For each click out item, click out time deducts session beginning time
Relative price [0,1]	Use min-max normalization for price
First interact (second)	The first interaction time of the item in the session deducts session beginning time
Last interact (second)	The last interaction time of the item in the session deducts session beginning time
Position in list (0,1]	For each item, the inverse of position in impression list in each session
Prices (>= 0)	For each item, the according price in each session
Impression length (>= 0)	The number of items in each impression in each session
Item position (>= 0)	For each item, the position in impression list in each session
Session time (second)	For each session, the session beginning time minus session end time
Num interact	For each session, the number of interactions for each item
Item first interact over session time [0,1]	For each item, first interaction time over session time
Item first interact to session end over session time [0,1]	For each item, session end time deducts item first interaction time over session time
Item first interact over first interact to session end [0,1]	For each item, first interaction time over session end time deducts first interaction time
Item last interact over session time [0,1]	For each item, last interaction time over session time
Item last interact to session end over session time [0,1]	For each item, session end time deducts item last interaction time over session time
Item last interact over last interact to session end [0,1]	For each item, last interaction time over session end time deducts last interaction time
Item position over impression length [0,1]	For each item, item position over impression length
Item position to end over impression length [0,1]	For each item, item position to end over impression length
Item position over position to end [0,1]	For each item, item position over item position to end

3.3.3 User based features

User based features capture the users' preference of price and item position. Some users may like items with high price and in the front, while others may not. For every user, we filter out all the click out item prices and positions in list and calculate the average price and position.

3.3.4 Global features

Besides the features above, we also have global features (see Figure 4, which are not dependent on sessions and users. We can observe the probability of click out under different interaction actions from Figure 7. Most clickout items are distributed in probability interval of [0, 0.2] and [0.9, 1].

Table 4: Global Features

Global features	Definition
AS (0 or 1)	Asian
EU (0 or 1)	Europe
NA (0 or 1)	North America
SA (0 or 1)	South America
OT (0 or 1)	Other areas
Popularity (>=0)	For each item, the number of click out in the dataset
normalized popularity [0,1]	For each click out item, the number of click out over number of occurrence in impression list
probability of click out under interaction item image [0,1]	Number of sessions of click out over number of sessions of interaction item image
probability of click out under interaction item deals [0,1]	Number of sessions of click out over number of sessions of interaction item deals
probability of click out under interaction item info [0,1]	Number of sessions of click out over number of sessions of interaction item info
probability of click out under search for item [0,1]	Number of sessions of click out over number of sessions of search for item

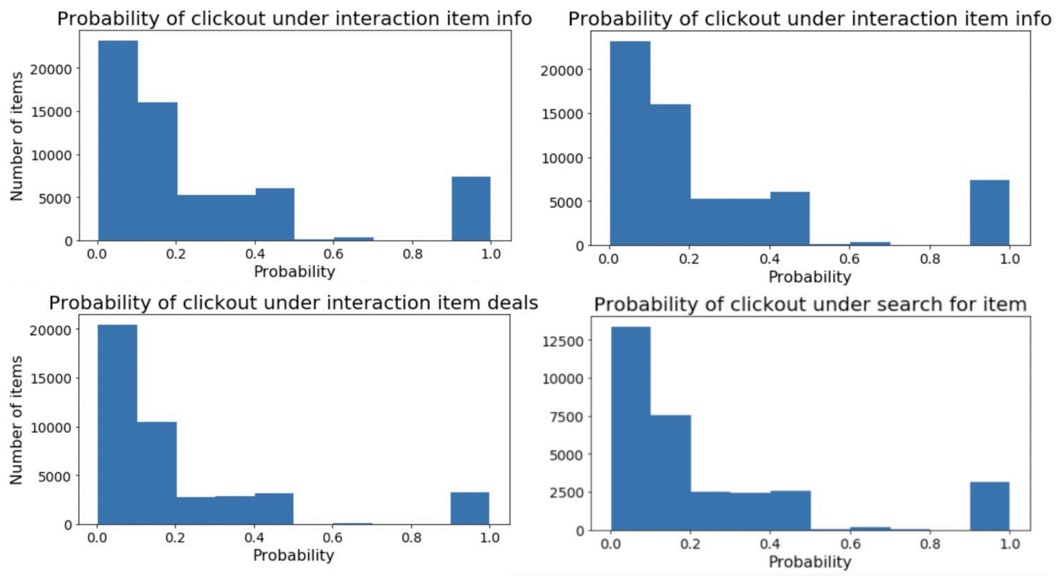


Figure 7: Probability of click out under different interactions

4 Methodology

4.1 Pipeline

We tried to split the train data into train and validation dataset according to the number of sessions we have in train.csv. We choose 90% of the data as train dataset and 10% of the data as validation dataset.

After processing the raw data by exploding the impression list, adding item property features, session features, global features and user features we prepared, which can be fed into our model. Our model will give a value between 0 and 1 which is a probability that a user might click this item. On validation dataset, we could calculate the MRR locally according to our reference in validation set to evaluate the performance of our model. Then, we could format final result to get the submission format.

4.2 Feature Selection

Whenever we calculated a new feature, we use XGBoost to train on new features combined with old features. After fine tuning, if MMR score improves, we classify it as a useful feature. Otherwise, we discard this feature. We also tried different feature set to find the best combination that could help us achieve better performance.

4.3 Models

We tried different models and expect to use these models to explore extracted features in different perspectives. Then we ensembled different models to get our final result. Following subsections explain more about our models.

4.3.1 Baseline model

Before we try any complex model, we will start with simple and effective models to better understand the data and set the baseline model. The baseline model for the problem is based on item popularity. In the train dataset, we firstly get the number of clickouts that each item received, and then recommend each user a list of items according to the number of clickout for each item in the impression list in the test dataset.

4.3.2 Transition model

A Markov chain of order m is defined as:

$$p(X_t = x_t | X_{t-1} = x_{t-1}, \ldots, X_{t-m} = x_{t-m})$$

Therefore, in recommendation system, random variables are defined over I (item set). And each clickout over impression list can be treated as clikout over a bucket (B). As a resuit, an unpersonalized Markov chain can be written as:

$$p(B_t) = p(B_{t-1})$$

In order to predict the future clickout, we treat entire training data as time $t - 1$, and estimation of transition probabilities as clickout probability[5],

$$\text{prob}_{l,i} = \hat{p}(i \in B_t | l \in B_{t-1}) = \frac{\hat{p}(i \in B_t \wedge l \in B_{t-1})}{\hat{p}(l \in B_{t-1})} = \frac{|\{(B_t, B_{t-1} : i \in B_t \wedge B_{t-1}\}|}{|\{(B_t, B_{t-1}) : l \in B_{t-1}\}|}$$

MRR score for this model is 0.3, slightly higher than baseline model.
Next, we extend this model to personalized transition model . Instead of maintain a item-item 2D transition model, we construct a 3D personalized transition model, building separate transition models for different users[6]. In following equation u stands for one user,

$$\text{prob}_{u,l,i} = \hat{p}(i \in B^u_t | l \in B^u_{t-1}) = \frac{\hat{p}(i \in B^u_t \wedge l \in B^u_{t-1})}{\hat{p}(l \in B^u_{t-1})} = \frac{|\{(B^u_t, B^u_{t-1} : i \in B^u_t \wedge B^u_{t-1}\}|}{|\{(B^u_t, B^u_{t-1}) : l \in B^u_{t-1}\}|}$$

MRR score for this model is 0.5, much higher than baseline model and improves unpersonalized transition model.

Another important implementation detail should be noticed is that since we are mining massive dataset, building a full matrix using numpy is not feasible. Here, we use nested dictionary to build sparse matrix to reduce cost on memory.

4.3.3 Binary classification model

We used binary classification method by setting clickout items as positive samples and unclickout items as negative samples. Because of unbalanced training data, we used downsample method to reduce number of negative samples to 90% of training data. We also tried to change the the ratio of negative and positive samples to other values but the AUC result does not change a lot.

Our first model is binary logistic regression (LR) model. The LR model is the baseline for binary classification methods. For each user and each session, we want to classify whether the user click the item or not and give the probability of clicking out for each item. Then, we recommend each user a list of items according to clickout probability.

Another binary classification model is XGBoost. XGBoost is an optimized distributed gradient boosting designed to be highly efficient, flexible and portable. We try to tune hpyer parameters of max depth, learning rate, n estimators and regularization of the XGBoost.

4.3.4 CNN based model

Besides item-user interact information and user information included in the session features, user features and global features, we still have not made full use of item meta data.

There are 157 features in item meta data in total. If we directly feed them into binary classification model, the performance in MRR score will decrease. One potential reason behind this is that there are too many useless features that overwhelmed important features. One way to alleviate this effect is to use Singular Value Decomposition (SVD) to reduce dimension on item properties features. After some experiments, we find out best reduction dimension is 8.

Since we also need to compare items across one session, CNN model can help to explore comparison between item by kernels. Combined with global features, session features and user features, we grouped items information in a session together into a 2D matrix. By feeding this matrix (1 channel * feature length * impression length) into CNN, and using architecture shown in following figure, we can achieve 0.58 MRR score.

Inspired by model architecture from paper [7], we also include personalized item-item transition model into CNN model. By concatenating estimation transition probability into final fully connected layer, we can achieve 0.595 MRR score.

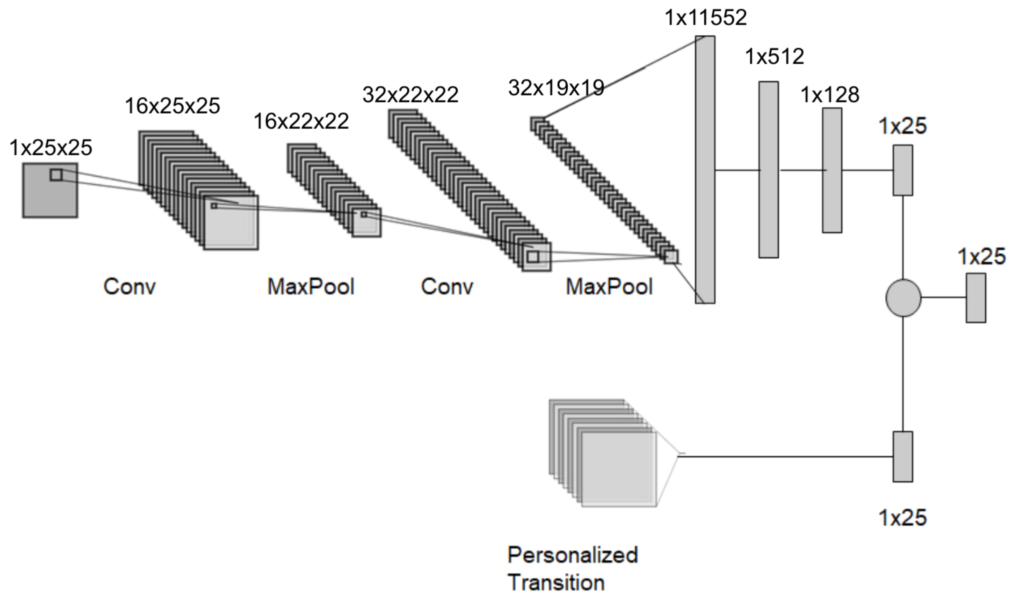


Figure 8: CNN model Framework

4.3.5 Ensemble and full pipeline

To combine these two models, we use a two-stage model architecture similar to paper [9].

At first, data are extracted into important features, combined with item meta data, fed into transition-CNN model. Then, in the second stage session features and global features are directly connected to XGBoost model together with CNN prediction results. Last but not least, we also use different models for sessions without any user actions, which is called cold-start handling in the figure.

5 Experiment

5.1 Evaluation

The main evaluation method we use for our project is the mean reciprocal rank (MRR) which is the official evaluation metric for submissions of Trivago RecSys Challenge 2019. The reciprocal rank of

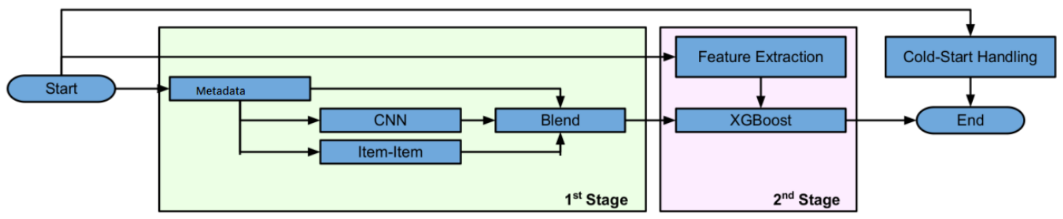


Figure 9: Model Architecture

a query response is the multiplicative inverse of the rank of the first correct answer[1]. This challenge also provides a leaderboard displaying the metric calculated result on the validation dataset. It will recalculate the final score on a different dataset (confirmation group) at the end of the challenge. The challenge also provides a simple baseline algorithm that could help us verify the submission format. The calculation of MRR could be defined as following, where $|Q|$ is the number of queries:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

To evaluate the approaches and models, we will also use other evaluation metrics to help us improve our model. For example, we tried methods such as receiver operating characteristic (ROC) curve and area under the curve (AUC) in binary classification model as the evaluation metrics of the challenge.

5.2 Results

So far, we have achieved different results progressively using different models as shown in Table 5. The best one is that we ensembled these models and get the final MRR as 0.604 on the ACM RecSys 2019 leaderboard.

Table 5: MRR results

Model	**MRR**	**Remark**
Baseline	0.288	
Transition model	0.503	
Logistic regression	0.574	
Decision Tree	0.576	max depth = 100, min samples split = 10, min samples leaf = 20
XGBoost	0.582	max depth = 4, learning rate = 0.1, n estimators' =100
CNN	0.585	filter window size = 5, 3 FC layers
CNN with transition	0.595	filter window size = 5, 3 FC layers
Ensemble	0.604	0.4 CNN with transition + 0.6 XGBoost

6 Brief Conclusion

In this project, we made use of user information, item information, user-item interaction data and build models to understand them. After a lot of tedious work on feature engineering, we find some really important features (interact time, item position, etc.) that can boost performance for simple models, like XGBoost, Logistic Regression, etc. To fully exploit these features, deep learning models and traditional recommendation system are also applied. Our final model architecture is a ensemble of XGBoost, transition model and CNN model.

7 Future work

Our project has served as the first step for 2019 RecSys challenge. We've already achieved quite good results but we need to further explore the dataset and extract more features. More, many classical recommendation models have not been tried like session-KNN, etc. We also plan to use more deep learning models to make full use of features list.

References

[1] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 621–630, New York, NY, USA, 2009. ACM.

[2] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.

[3] Jaehun Kim, Minz Won, Cynthia Liem, and Alan Hanjalic. Towards seed-free music playlist generation: Enhancing collaborative filtering with playlist title information. In *Proceedings of the ACM Recommender Systems Challenge 2018*, page 14. ACM, 2018.

[4] Diego Monti, Enrico Palumbo, Giuseppe Rizzo, Pasquale Lisena, Raphaël Troncy, Michael Fell, Elena Cabrio, and Maurizio Morisio. An ensemble approach of recurrent neural networks using pre-trained embeddings for playlist completion. In *Proceedings of the ACM Recommender Systems Challenge 2018*, page 13. ACM, 2018.

[5] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 811–820, New York, NY, USA, 2010. ACM.

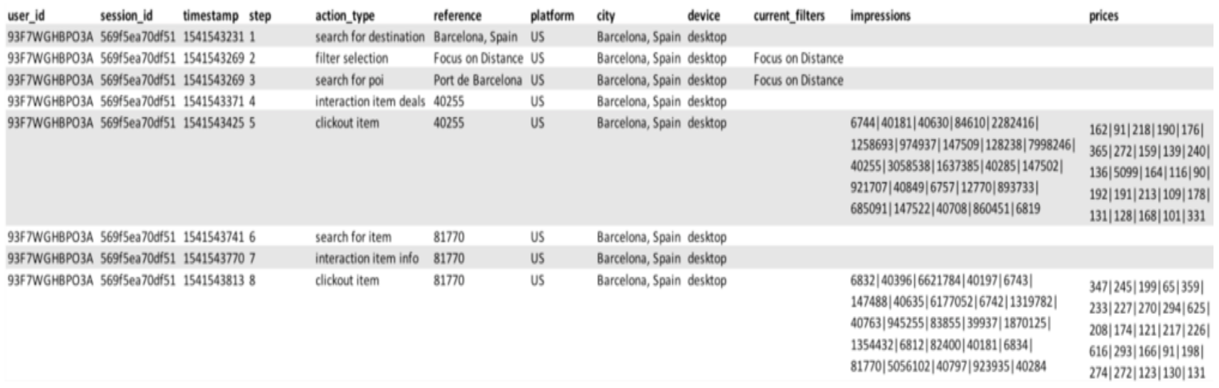
[6] Xiaodan Song, Belle L. Tseng, Ching-Yung Lin, and Ming-Ting Sun. Personalized recommendation driven by information flow. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 509–516, New York, NY, USA, 2006. ACM.

[7] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 565–573, New York, NY, USA, 2018. ACM.

[8] trivago. trivago RecSys Challenge 2019 Dataset.

[9] Maksims Volkovs, Himanshu Rai, Zhaoyue Cheng, Ga Wu, Yichao Lu, and Scott Sanner. Two-stage model for automatic playlist continuation at scale. In *Proceedings of the ACM Recommender Systems Challenge 2018*, page 9. ACM, 2018.

8 Appendix



user_id	session_id	timestamp	step	action_type	reference	platform	city	device	current_filters	impressions	prices
93F7WGHBPO3A	569f5ea70df51	1541543231	1	search for destination	Barcelona, Spain	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543269	2	filter selection	Focus on Distance	US	Barcelona, Spain	desktop	Focus on Distance		
93F7WGHBPO3A	569f5ea70df51	1541543269	3	search for poi	Port de Barcelona	US	Barcelona, Spain	desktop	Focus on Distance		
93F7WGHBPO3A	569f5ea70df51	1541543371	4	interaction item deals	40255	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543425	5	clickout item	40255	US	Barcelona, Spain	desktop		6744|40181|40630|84610|2282416|1258693|974937|147509|128238|7998246|40255|3058538|1637385|40285|147502|921707|40849|6757|12770|893733|685091|147522|40708|860451|6819	162|91|218|190|176|365|272|159|139|240|136|5099|164|116|90|192|191|213|109|178|131|128|168|101|331
93F7WGHBPO3A	569f5ea70df51	1541543741	6	search for item	81770	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543770	7	interaction item info	81770	US	Barcelona, Spain	desktop			
93F7WGHBPO3A	569f5ea70df51	1541543813	8	clickout item	81770	US	Barcelona, Spain	desktop		6832|40396|6621784|40197|6743|147488|40635|6177052|6742|1319782|40763|945255|83855|39937|1870125|1354432|6812|82400|40181|6834|81770|5056102|40797|923935|40284	347|245|199|65|359|233|227|270|294|625|208|174|121|217|226|616|293|166|91|198|274|272|123|130|131

Figure 10: A user's actions in a sample session

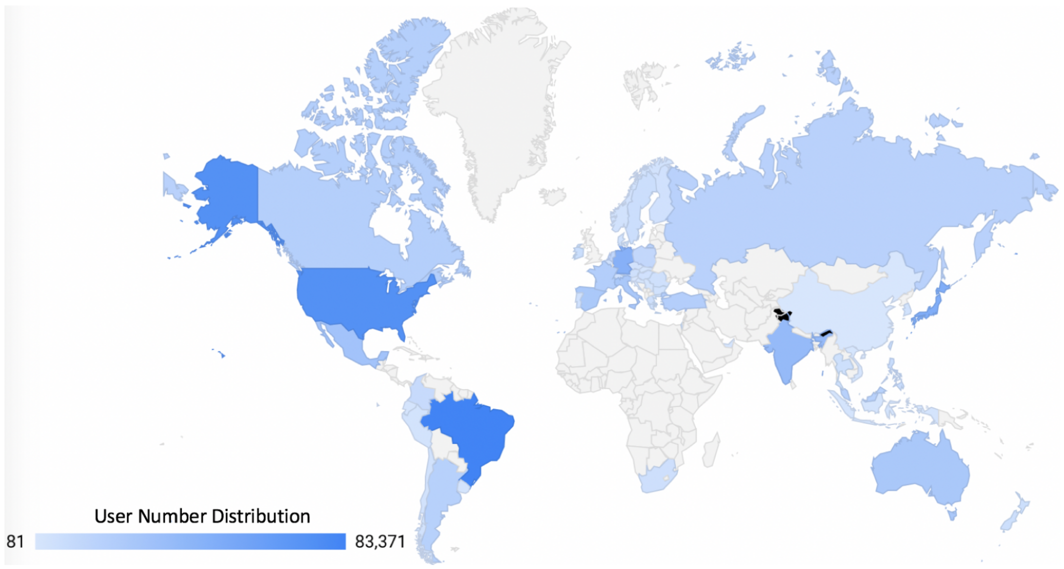


Figure 11: User Country Distribution.

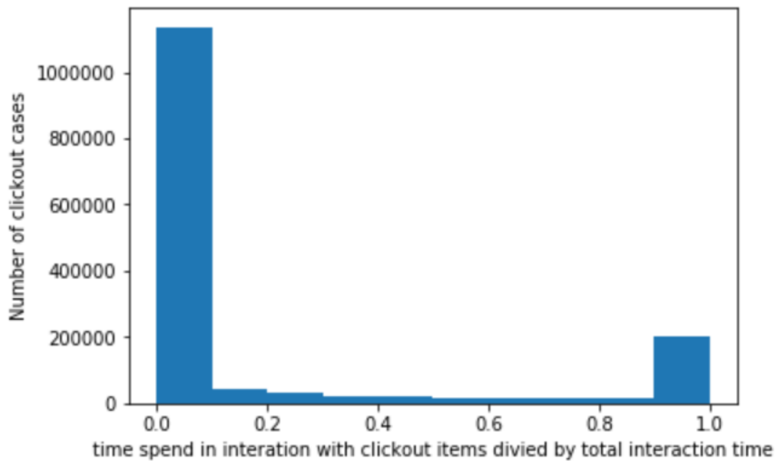


Figure 12: Interaction time and clickout distribution

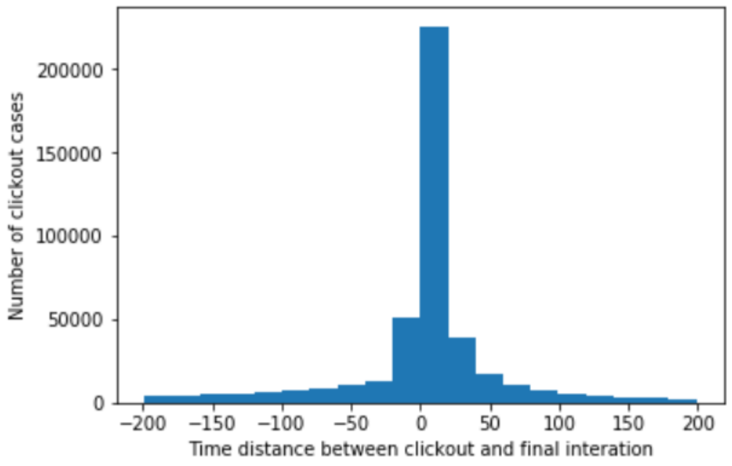


Figure 13: Time distance between clickout and interaction distribution

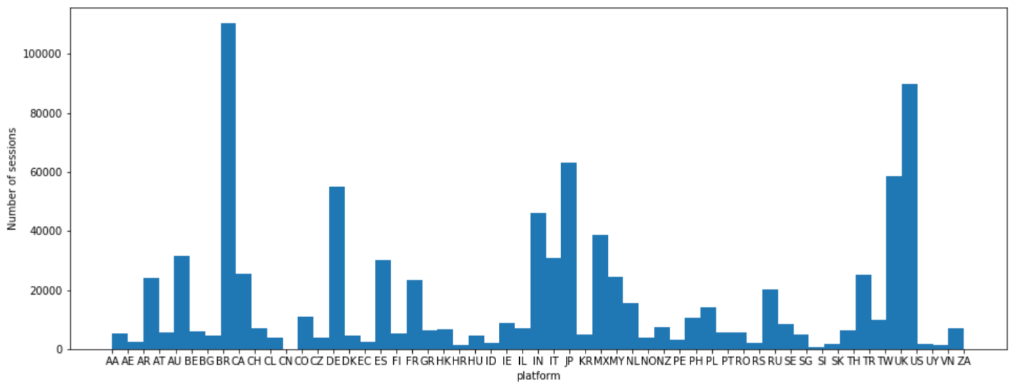


Figure 14: Platform distribution

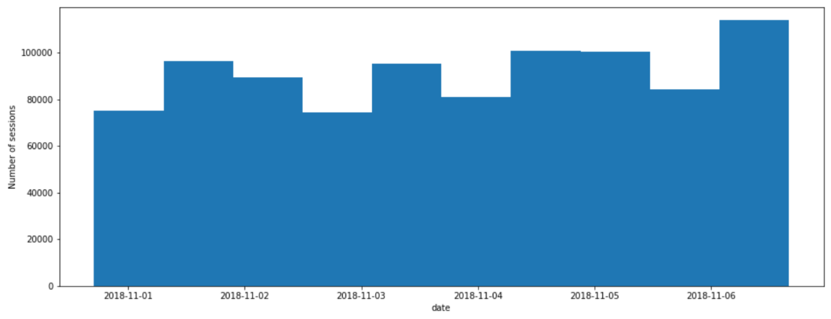


Figure 15: Date distribution