Data Mining Techniques
Project 2
Group 115

Julia Sudnik[1 [2715115]], Meifang Li[1 [2719570]], and Rafael Cárdenas-Heredia[1 [2718909]]

Vrije Universiteit Amsterdam

Introduction

Today's e-commerce systems must constantly offer customization in the presentation of their content. In this sense, recommendation systems make personalised suggestions and provide information about what is available in the system. "You may like to purchase...", "People who bought this article also bought ..." or "You may be interested in..." are phrases that are increasingly present in the daily activity of users who consume products in online services. Data mining techniques can be used to provide customization in such systems as they pursue the automatic discovery of knowledge contained in large databases.

The main objective of this type of approach is to use existing data on user preferences applying machine learning techniques to develop a recommendation system, suggesting new elements adapted to user tastes based on the user or product profiles generated for this purpose. Through these computational methods the identification of the user's browsing pattern and their profile in real time allows an immediate and satisfactory response to the needs of the consumer.

In this report, the use of the information collected by a hotel reservation portal will be presented. The assignment is done within the framework of the Data Mining course of the Vrije Universiteit Amsterdam, with the aim of demonstrating the handling of the concepts to which the students have been exposed during the classes. In order to demonstrate mastery of the basic application of data mining techniques, a task corresponding to the domain of recommendation systems is approached.

Task 1: Business Understanding and Related Work

The proposed assignment corresponds to a scenario inspired by the competition "Personalize Expedia Hotel Searches - ICDM 2013" hosted by Expedia in 2013 on the Kaggle platform. Both in the latter and in the activity that is now being addressed, the establishment of a system for predicting hotel preferences is sought from a database of user interactions with the Expedia website. The database is made up of 4958347 entries related to 54 features that offer information on various aspects of this interaction, both corresponding to user characteristics and

their interaction with the web interface as well as corresponding to the characteristics of the hotel properties to which the records are related. The practical goal of obtaining predictions about the possible future preferences of the users would respond to the objective of, eventually, suggesting the hotel properties to the user, thus hoping to facilitate a consequent booking based on the knowledge that has been previously about the user in question.

During the literature search for the approach to this task, multiple sources corresponding to data mining approaches to the original challenge posed in Kaggle by Expedia were found. It is pertinent to clarify that none of these sources correspond to peer-reviewed documents, however this is not surprising, since the effective approach to this class of competences does not always necessarily lead to a scientific or similar publication.

We evaluated presentations of the three best predictions in the original challenge published by the authors of the challenge [1]. In the pre-processing of the data all three strategies had some similarities. For instance, all winners used normalisation of numerical values by *search_id*. Owen Zhang (1st Place) decided to additionally use *prop_id* and *destination_id* as normalisation keys, while Jun Wang and Alexandros Kalousis (2nd Place) chose even a larger list of normalisation keys consisting of *search_id, prop_id, month, srch_booking_window, srch_destination_id, prop_country_id* .

Intrestingly, we found different approaches to handling missing values - Owen Zhung (1st Place) imputed missing values with a negative value, while Jun Wang and Alexandros Kalousis (2nd Place) replaced them with 0. Liu et al. (3rd Place) on the other hand, used the first quartile to represent the missing data.

All of the winning strategies involved additional feature engineering. Owen Zhang examined the difference between hotel price and recent price, and order of the price within the same *search_id*. Jun Wang and Alexandros Kalousis evaluated the differences in user's history data (i.e. difference between past rankings and hotel's ranking, and difference between past price and hotel's price) and hotel quality (estimated by the overall probability of being clicked on or being booked from the whole data-set). Moreover, Owen Zhang decided to estimate position of the hotel in the test-set, as this feature did not exist there.

In Table 1 we present which models were evaluated by some of the best participants of the original challenge (methods that achieved the highest results are in bold). The best models appear to be GBM and LambdaMART.

Interestingly, for the 1st place winner the most relevant feature was the one created by the winner using feature engineering, i.e. *estimated position*. Further, price was found as one the most predictors for both Owen Zhang and Liu et al. (see Table 2 for more feature relevance).

[1] https://www.dropbox.com/sh/5kedakjizgrog0y/LEDF CA7J/ICDM2013 All references in this paper to any of the three winners relate to information found in this dropbox published by the authors of the Kaggle challenge. It consists of slides with the introduction to the challenge, data description and three presentations created by each of the winning teams about their strategies

Table 1. Models used for Expedia challenge

Authors	Models Used
Zhang	**GBM**
Wang, Kalousis	**LambdaMART**, Regression, SVM-Rank
Liu et al.	**Random Forest**, **GBM**, **LambdaMart**, Deep Neural Networks, Linear Model Factorization Machine

Table 2. Feature Importance (in Order of Importance)

Authors	Most Important Features
Owen Zhang	position, price, prop_location_score2
Liu et al.	price_usd, prop_location_score1, prop_location_score2

Task 2: Data Understanding

General Analysis

The set of resources provided for the resolution of the task in Kaggle includes two databases and a sample that exemplifies the type of file that must be delivered as a solution. The training and testing data sets comprise, respectively, 4958347 and 4959183 entries. In the case of the former, this information is arranged in 54 columns, which offer a diverse array of information about the characteristics of each search, the hotels related to the searches, the interaction of the user in particular with the platform and the interrelationship between the Expedia platform and its competitors. Also, this information can be found in the training data, with the exception of some features: *position, click_bool, booking_bool, gross_booking*.

The objective of this EDA is to obtain valuable information about the data from which the training of the models will be carried out for the eventual prediction from the testing set. For this reason, the EDA will only be carried out based on the information provided in the training set, with the intention of avoiding information biases.

The dataset provided, as mentioned before, is comprised in 54 columns, all of which posses fully numerical information with the particular exception of the one comprising the date of the search. Once imported, it can be seen that the data types are distributed as observed in Fig. 1. This is, however, not a faithful portrayal of the function of the different data in the dataset. While most of the information provided in the dataset is, indeed, quantitative in nature, there exist several (6, to be exact) categorical columns that, while represented with numbers, don't intend to portray magnitudes of any kind, but numerical ID's. The same consideration has to be made for the only "object" element in the columns since, while initially categorized in the table simply as "object", contains chronological information regarding dates and hours.

Although the dataset is comprised in 54 columns, theres a set of features around which the analysis will pivot, when it comes to portraying valuable information. These are those regarding the location of the agents of the search,

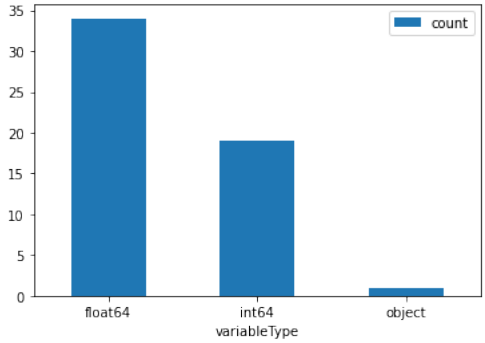

Fig. 1. Data by type

as well as the time frame in which they are comprised. These key elements are summarized in Table 3.

Table 3. Key features summarized

Feature	Size
Search entry ID	199795
Website ID	210
User location ID	34
Property ID	129113
Property location ID	172
Search entry target destination ID	18127
Payment received through booking USD	0 - 159292.38
Time frame	2012-11-01 00:08:29 - 2013-06-30 23:58:24

Missing data

As can be seen in Figure 2, the data provided for the training phase is widely sparse in some categories. This is especially evident in a couple of features related to the history of active interaction between the user and the platform (*visitor_hist_starrating, visitor_hist_adr_usd*) and the whole set of features related Expedia's competitors. It can be observed that these information gaps are diverse in their extension, comprising percentages from 21.9% to 98.1%.

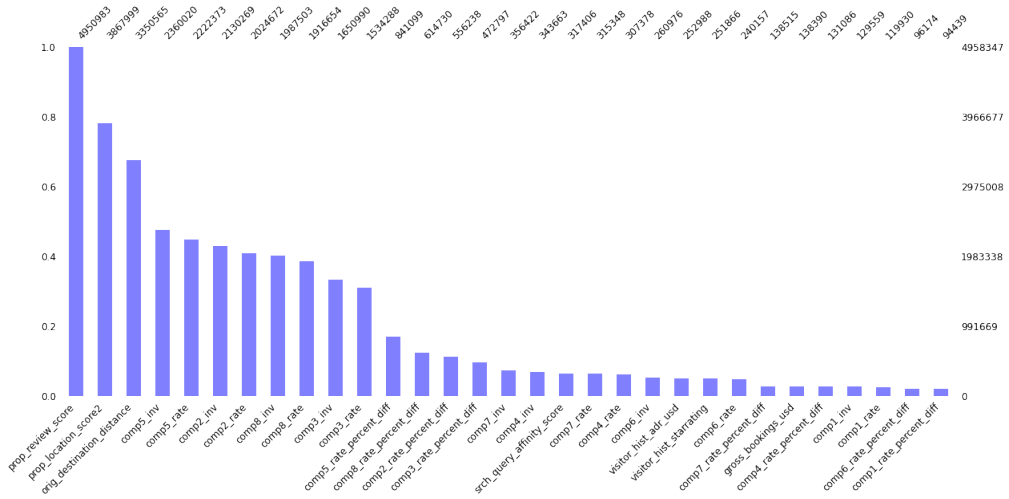

Fig. 2. Percentage of missing data in incomplete features

Visitor Features Analysis

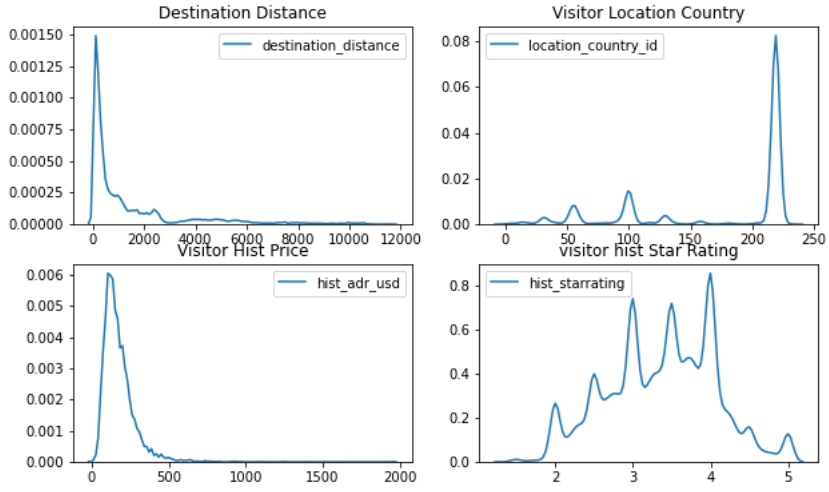

Fig. 3. Distribution of features about Visitors

With the intent to visualize the distribution of the information regarding visitor features, four plots were generated, corresponding to the metrics of distance between the search where the location was generated and the property in the entry, the location of the users, the price per night of the room, and the rating provided by the users to the hotels. These can be observed in Fig. 3.

Various assessments can be made from the graphs generated. First off, there is a clear skewness in the distance data towards shorted distances, in contrast to farther travel destinations, being most of the information comprised in the interval between 0 and 2000; this helps visualize that among the population of users that use this platform there is a clear tendency towards closer destinations. It can be said, also, that the user base of Expedia is highly diverse, since, although there's a noticeable few very specific country ID's with a higher density, the highest one has a density of around 0,08. Skewness can also be observed in the historical information of prices paid by users; this shows a clear tendency towards rooms priced under 500USD per night, with a observable preference for rooms under the price if 200USD. Finally, when it comes to the rating information, it can be said that there's a tendency towards more moderate ratings, between 3 and 4, being extremely poor or extremely high ratings much rarer.

Search Features Analysis

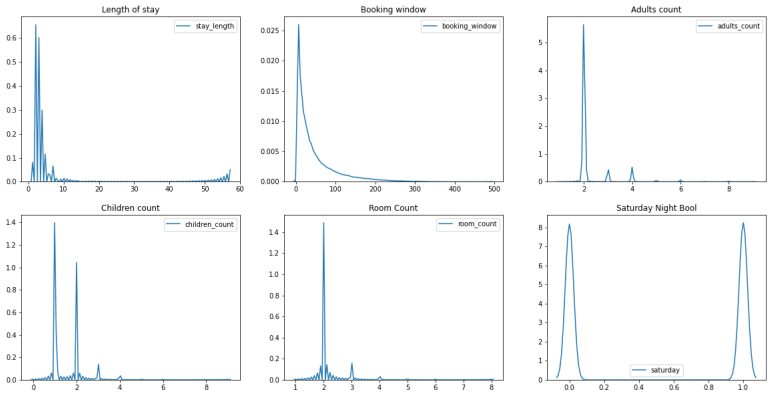

Fig. 4. Distribution of features about search

From observing the graphed search features, it can be assessed that users that use the search platform tend to have a preference towards shorter stays, in contrast to lengthier ones; however, there is a small but noticeable amount of users that use the platform with the intend to book very lengthy stays. There

is also an observable tendency towards wider booking windows, which conveys information about a user base that has a tendency towards planned trips, in contrast to impromptu ones. When it comes to kids, it can be observed that most families that travel with children show a tendency towards traveling with fewer of them, since there is a noticeable increase in density for 1 and 2 children. When it comes to adults the tendency continues, being noticeable that adults tend to travel in sets of two, in clear contrast with larger groups. The graph illustrating the room count in the entries shows a marked preference for searches of 2 rooms. Finally, it can be observed from the Saturday Night Bool graph that the frequency with which people search for rooms Saturday night is comparable to that of people searching for rooms during the rest of the week summed together, which shows a marked preference of Saturday as a day during which time is dedicated to this sort of searches.

Hotel Features Analysis

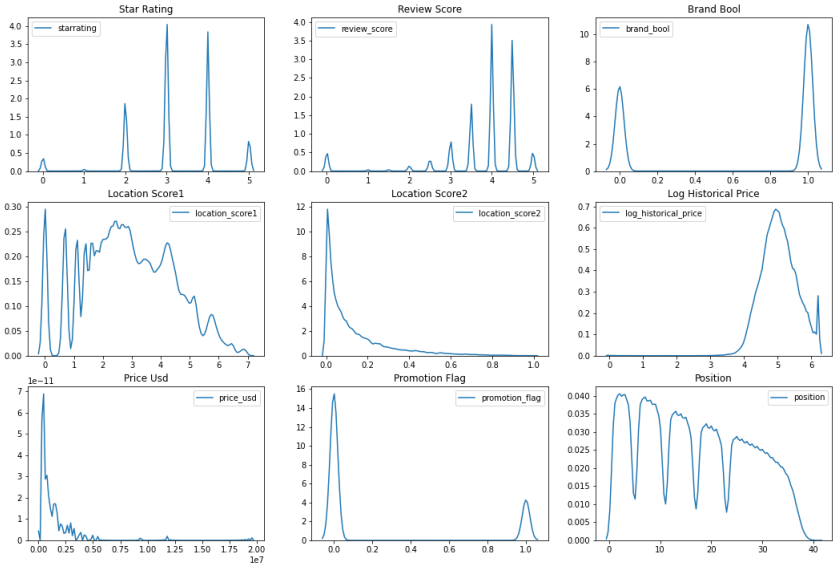

Fig. 5. Distribution of features about hotel

When it comes to hotel features, it can be observed a tendency towards more moderate star ratings rather than higher or lower ones; being, however, the 5

star ratings substantially higher in contrast to very low ratings of 0 and 1. This same tendency can be observed in the review score, showing a clear increase of density in scores between 3,5 and 4,5 and substantially lower scores towards the extremes. When it comes to brand preference, it's evident that there's a skewness towards hotel chains, in contrast to smaller independent hotels. The graphs portraying the data regarding location scores show a great amount of difference between each other, showing a much clearer gradient in the change of desirability scores for the Location Score 2 metric, in contrast to a much more disorganized distribution in Location Score 1. When it comes to the USD Price information, there can be observed a marked skewness towards lower prices displayed in searches for hotels, being most observable values positioned between 0,00 and 0,25. From the promotion graph, it can be observed that it was much more frequent for a hotel to be displayed without a promotional flag during searches.

Performance with Respect to Competitors

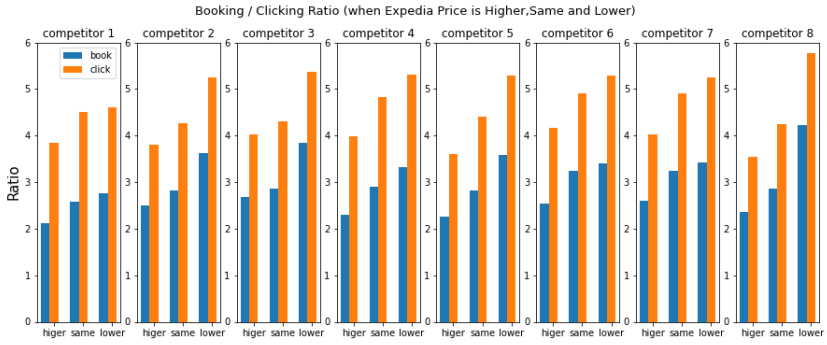

Fig. 6. Booking and clicking ratio of competitors

After plotting the ratio with which expedia hotels are clicked and book in scenarios of higher, same an lower prices with respect to competitors, a homogenous tendency is observed for all competitors almost equally: when prices are lower, the ratio of clicking tends to remain in the vicinity of 5 while, for most cases, the book ratio tends to be significantly lower by nearly 2 in magnitude; this same book/click relationship is observed for "same" and "higher" values, with lower ratios being displayed for all competitors as the price disadvantage for Expedia increases. Noticeable, but small, diferences can be observed in the case of competitor, being the click ratio clearly lower for "lower" prices, while the opposite behaviour is observable for this category for competitor 8.

Task 3: Data preparation and Feature Engineering

Data preparation

The biggest weakness of the given data-set was found in the large amount of missing values, half of the features has more then 50% values missing. For this reason a threshold of 90% was established, and features with more than 90% missing values were dropped. To address the rest of incomplete data, an approach used by Jun Wang, the 2nd place winner of the original Kaggle competition [2] was taken and all NA were replaced with '0'. This decision is based on the intuition that users hesitate to book hotels with lacking information, for instance about the star rating. Further, four features existing in the training set *position, click_bool, booking_bool, gross_booking* do not exist in the test set. Those features were dropped as well. Furthermore, due to using feature engineering on competitors information it has been decided to drop all their pre-existing features, assuming that the new features will contain similar data. Similarly, column *date_time* was replaced with the new features mentioned in the previous section. Moreover, *search_id* column due to its randomness is believed to not bring any quality information to the model and is also not used for predicting the target value (although used for grouping the rankings).

Feature Engineering

A number of feature engineering has been applied to the data-set. Firstly, from the complex column *date_time* two new simple features were extracted: *month* and *day_of_the_week*. Secondly, it's been decided to average out the competitors information which resulted in features *comp_rate_avg*, *comp_inv_avg*, *comp_diff_avg*. Moreover, we discovered that users tend to book hotels with similar star ratings and price over time [2]. For this reason, two more features comparing the star ratings and price of the hotels booked in the past and hotels searched were created, i.e. *hist_star_rating_diff* and *hist_price_diff*.

Furthermore, a number of feature normalisation was applied. We followed an intuition that some features can be perceived as high or low depending on values of other hotels within each individual search. Additionally, we assumed that prices might differ depending on the country or month (e.g. prices might be higher during summer) and that star ratings might also be distributed differently in different countries. Due to mentioned reasoning a total of 9 new features was created using normalisation on existing features as seen in Table 2.

Lastly, we found it necessary to create the target feature that will be later on predicted in the model. We found using only the information from the *booking_bool* to be not sufficient as it does not give any clue on how likely is the user to book the hotels with value 0. Thus a new feature *target* has been created assigning a value 2: to a hotel that was booked; 1: to a hotel that was clicked on; 0: to a hotel that has not been clicked on.

Below is the final list of the features used in the model:

Table 4. Feature Normalization

Normalized Key	Normalized Feature	Normalized Key	Normalized Feature
srch_id	price_usd	prop_id	price_usd
srch_id	prop_starrating	month	price_usd
prop_country_id	prop_starrating	prop_country_id	price_usd
srch_id	prop_location_score2	srch_id	prop_location_score1
srch_id	prop_review_score	-	-

```
['site_id', 'visitor_location_country_id', 'visitor_hist_starrating',
 'visitor_hist_adr_usd', 'prop_country_id', 'prop_starrating',
 'prop_review_score', 'prop_brand_bool', 'prop_location_score1',
 'prop_location_score2', 'prop_log_historical_price', 'price_usd',
 'promotion_flag', 'srch_destination_id', 'srch_length_of_stay',
 'srch_booking_window', 'srch_adults_count', 'srch_children_count',
 'srch_room_count', 'srch_saturday_night_bool', 'srch_query_affinity_score',
 'orig_destination_distance', 'month', 'dayofweek', 'comp_rate_avg',
 'comp_inv_avg', 'comp_diff_avg', 'hist_star_rating_diff', 'hist_price_diff',
 'price_usd_norm_by_srch_id', 'price_usd_norm_by_prop_id',
 'price_usd_norm_by_month', 'price_usd_norm_by_prop_country_id',
 'prop_starrating_norm_by_srch_id',
 'prop_starrating_norm_by_prop_country_id',
 'prop_location_score2_norm_by_srch_id',
 'prop_location_score1_norm_by_srch_id',
 'prop_review_score_norm_by_srch_id']
```

Task 4: Modelling

The main goal of this competition is learning to rank and the existing algorithms for ranking problem could be categorized into three groups: the pointwise, pairwise, and listwise approach[1]. Listwise approaches usually perform better than pairwise approaches and pointwise approaches in practice. Thus we first implement the well-known algorithm that incorporates listwise approaches such as LambdaMART but come into failure due to the extremely time-consuming process. Then we specially choose the Microsoft's LightGBM model because it not only supports for ranking(LambdaRank) but also is extremely efficient and relatively accurate. LightGBM is a gradient boosting framework using decision trees to increases the efficiency of the model. It splits the tree leaf-wise and chooses the leaf with maximum delta loss to grow so this method has lower loss compared to the level-wise algorithm. Hence, leaf-wise tree growth might achieve higher accuracy and it is especially suitable for large dataset. In addition, the speed is shocking compared to other algorithms.

Parameters Tuning

Given the LightGBM model we choose, We also have to tune the parameters in the model. As shown in Table 5, the objective is *lambdarank* since the task is to rank and metric is specified for *ndcg* for LGBMRanker. The number of boosted trees we have tested is 500/1000/1500 but the larger number improves the result only a little with much longer time spent. Thus we think 500 is enough to ensure relatively high accuracy and save our time. Then the learning rate we choose is 0.12, a little larger than the default 0.1, since a suitable learning rate could let the objective function converge to the local minimum in a suitable time. Next, max position optimizes NDCG at this position so the value we determine is 5. Also, the number of seed we choose is 69. Additionally, we tried two boosting types: 'dart' for Dropouts meet Multiple Additive Regression Trees and 'rf' for Random Forest. However, we find that the 'rf' would experience early stopping when we stop training if validation scores don't improve for 200 rounds and the best score of 'rf' is lower than that of 'dart' when other parameters are the same. Thus we choose 'dart' for the final model concerning better accuracy. Finally, the categorical features we used in training model is *day, month, prop_country_id, site_id, visitor_location_country_id*.

Table 5. Parameters of LightGBM

Parameter	Value	Function
objective	lambdarank	Specify learning objective to be used
metric	ndcg	'ndcg' for LGBMRanker
boosting_type	dart	Dropouts meet Multiple Additive Regression Trees
n_estimators	500	Number of boosted trees to fit
learning_rate	0.12	Boosting learning rate
max_position	5	optimizes NDCG at this position
random_state/seed	69	Random number seed
categorical_features	5	Specify the categorical features for model training

Evaluation

We split the training dataset into two datasets: train and validation(150000) and fit into the final model with parameters mentioned above. The result score of ndcg for training and validation is 0.556522 and 0.504604 respectively. Then the prediction process is implemented in the test dataset by following the same data processing of training set. The final score of accuracy is 0.38867 as shown in Kaggle and we rank 46 out of 283 teams(till 22pm 21/05).

The feature importance could indicate which features are important in the predicting process. As shown in Figure 7, we can say that the top three important features are *prop_country_id, visitor_location_country_id, price_usd_norm_by_srch_id* which means the countries that the hotel and customer is located in is very essential to the predictions, also the displaced price after normalized by the search ID.

These features are surprisingly not the same as we expected since the features other people find important are all different as we mentioned in Task 1.

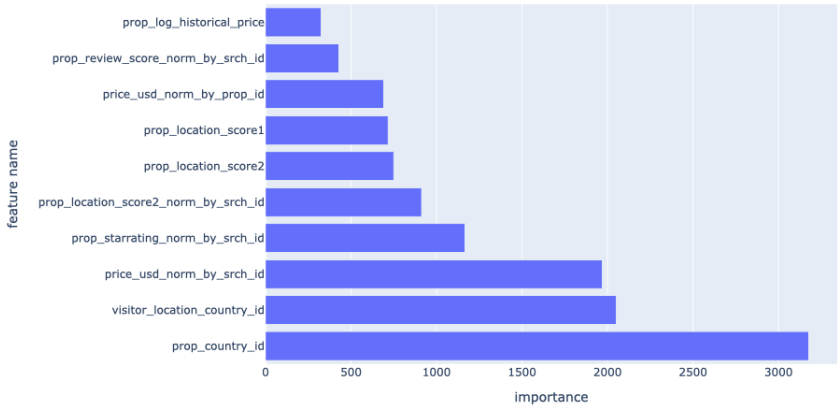

Fig. 7. Feature importance of final model

We have gained various knowledge in DMT techniques and a good understanding of LTR models. This is the first time we step into the data analysis in the real world and the amount of this dataset is as large as expected. We have to deal with the missing values and explore the relationship between features, then add some new features and drop some unimportant features. During this feature engineering process, there are chances that we omit something which could significantly influence the results so we try different combinations when normalizing the features. And the large dataset and a large number of features make it difficult to manage the choices since the process of data takes most of the time. Regarding the models, it is also struggling to select a suitable model and certain parameters to use in the model. Sometimes it is very disappointing to find some parameters don't improve the results much when we have wasted so much time tuning them. However, the final score proves our efforts and better understanding of this task as we expected. It is worthy to dive deeper into this field and learn a lot about DMT techniques that could be of great help in the future.

References

1. Liu, T.Y.: Learning to rank for information retrieval (2011)
2. Wang, J.: Icdm$_2$013, $https://www.dropbox.com/sh/5kedakjizgrog0y/_LE_DFCA7J/ICDM_2013?preview = 4_jun_wang.pdf$