Recommending the Duration of Stay in Personalized Travel Recommender Systems

Abhishek Agarwal[1,2], Linus W. Dietz[1,2]

[1]Department of Informatics, Technical University of Munich, Garching, 85748, Germany
[2]Both authors equally contributed to the paper

Abstract
The main focus of recommender systems research has been recommending fitting items to the users. However, in some domains, not only which item but also the quantity the target user should consume could be part of the recommendation. In this work, we tackle the under-researched problem of recommending the duration of stay in the domain of destination recommendation. Using two data sets, one based on hotel bookings and the other on mobility derived from geotagged Tweets, we perform extensive feature engineering with unsupervised learning to discover types of users and graph embeddings of the cities. In our experiments, we compare the performance of supervised learning algorithms with varying features to statistical baselines for predicting the duration of stay at a destination. The results underline the task's difficulty: we obtain the best results for the hotel bookings data set using personalized mobility embeddings with CatBoost. At the same time, the simple strategy of recommending the mode duration of all users is competitive in the noisy Twitter data set.

Keywords
destination recommendation, duration of stay, feature engineering, graph embeddings, offline evaluation,

1. Introduction

The mainstream of recommender systems research has been to use sparse matrices of ratings to predict the suitability of items for a given user with specific needs. Since the early beginnings of Collaborative Filtering, recommendation algorithms have reached tremendous maturity [1] to the extent that it is unclear whether novel approaches can outperform long-established algorithms in terms of ranking accuracy [2]. Current recommendation challenges include ensuring fairness [3, 4], multi-sided markets [5], and sequential recommendations [6].

Unlike these recommendation research problems, which are concerned with choosing suitable items and determining rankings, we analyze the problem of recommending the quantity in which a given user should consume an item. Concretely, our goal is to evaluate algorithms that compute the optimal duration of stay at a touristic destination based on hotel booking data and traveler mobility from location-based social networks. In tourism recommendations, this is a significant challenge as determining the duration of stay at specific locations is a cornerstone of online tour planners [7, 8]. Note that this problem differs from the question of how many

RecSys Workshop on Recommenders in Tourism (RecTour 2022), September 22th, 2022, co-located with the 16th ACM Conference on Recommender Systems, Seattle, WA, USA

abhishek.agarwal@tum.de (A. Agarwal); linus.dietz@tum.de (L. W. Dietz)

items we should recommend, e.g., Knapsack optimization approaches for resolving the Tourist Trip Design Problem [8, 9], or recommending items multiple times within a travel package [10].

To compute a personalized duration of stay at a travel destination, Dietz and Wörndl suggest an algorithm that relies on a statistical analysis of a user's previous trip duration relative to other travelers [11]. Our work has a similar goal; however, we present a systematic experimental design based on two different data sets. We compare the predictive performance of multiple statistical baselines – including the method proposed in [11] – to classic supervised machine learning algorithms with different features that we derive from unsupervised machine learning. This feature engineering process is specific to this domain and gives further insights into essential factors for algorithmic travel planning. Framing the recommendation problem as a quantitative prediction task instead of using a survey to evaluate the algorithms makes the results reproducible and comparable for future research.

The structure of the paper is the following: In the upcoming Section 2, we survey the limited prior work on this open research problem. Section 3 is devoted to the two data sets and the pre-processing steps employed. In Section 4, we describe our features engineering efforts, which we will use in our experiments described in Section 5. We discuss our results in Section 6 and finally draw our conclusions in Section 7.

2. Related Work

Actively recommending the duration of item consumption received limited attention from the scientific community. The expected duration of item consumption plays a role in sequence-aware recommendation scenarios, such as video recommendation [12] or content recommendation on social media to predict dwell times [13]. It is distinct from our approach since the problem in our paper is to recommend the duration of stay, not which destination to visit. The approaches mentioned above use the predicted item consumption duration to optimize business metrics such as the overall time spent on the platform [14].

Most literature mentioning recommending the amount of item consumption can be found in the area of travel recommendation, where it is often resolved in an ad-hoc manner [9, 15] or regarded as future work [16, 17]. For example, in a region recommender system, the travel regions are first scored based on the user preferences and combined into a composite trip [9]. The authors' proposal to determine the duration of stay is to apply a gradual decrease in the score of the travel regions by 5-10% per week, concluding the stay at the current region as soon as the score of the following region exceeds the score of the first.

The duration of stay is also necessary when planning a stay in points of interest. However, the approaches we found in the literature also do not systematically address this question. For example, the typical duration of stays for tourists in different point-of-interest categories has been analyzed using a Foursquare data set to improve context-aware services [18], but this approach is not personalized. Other systems incorporate average durations of stay to recommend trips within a city [15, 19] but only report the durations to the user as part of the overall trip duration. These durations can stem from commercial services such as Google Maps, which showcase information based on human mobility derived from mobile phones. Such information is not personalized, and due to the aggregation, we can not use it to compute recommendations for individual users.

It leaves us with the approach proposed by Dietz and Wörndl [11], which analyzes past user trips from Foursquare[1], a location-based social network, to understand their pace of travel. The mean percentile duration of the previously visited cities quantifies a given user's pace. To find the corresponding period of stay for the target destination, the mean percentile is computed using the duration distribution for all the users in that destination. In our work, we aim to systematically evaluate this approach with our novel machine learning efforts on two data sets described in the upcoming section.

3. Data Sets and Preprocessing

Currently, there are few suitable data sets for determining the duration of stay at a destination on a global level. We conduct experiments on two real-world data sets, one released by Booking.com and a self-collected data set of trips derived from geotagged Tweets.

3.1. Booking.com

The Booking.com Multi-Destination Trips Dataset [17] was made available as a part of the Booking.com WSDM WebTour 2021 Challenge[2]. The original challenge involved the sequential recommendation problem of predicting the last destination of a four-destination trip; however, the data set also includes the duration of stay at the individual destinations. Overall, the data set consists of over a million anonymized hotel bookings making it a suitable data set for predicting the duration of stays. Unfortunately, the anonymization masks the real destinations, which prevented us from gaining additional insights. Another artifact of the release of this data set is that the users originate from only five countries but have traveled to 107 different countries in total. Since this data set includes trips comprising exactly four destinations, it is a *clean* data set with a rather specific portion of the reality of international travel.

3.2. Twitter

To complement the Booking.com data set, we used the Twitter API[3] to query user timelines who have enabled sharing their geolocation in their Tweets. Using the `tripmining` library[4], we segmented the users' mobility into periods of being at home and away from their home city [20]. Consecutive periods abroad are regarded as trips, as long as specific data quality criteria are fulfilled. Given the nature of check-in-based data, we only know the user's location when they tweet, resulting in an incomplete view of the periods between two check-ins. However, given the large corpus of mobility data available to us, we could work with the subset of trips with at least one check-in each day, thus ensuring high data quality. Furthermore, to match the characteristics of the Booking.com hotel reservation data set, we only included multi-destination trips with four or more destinations in the analysis. Similar to the Booking.com data set, we

[1] https://foursquare.com

[2] https://www.bookingchallenge.com

[3] https://developer.twitter.com/en/docs/twitter-api

[4] https://github.com/LinusDietz/tripmining

did not include the available geographic information as training features, despite knowing the real city names in this data set. Thus, we know that most users come from countries where the platform is prevalent, notably the USA, Europe, and Japan. Since these trips do not suffer from an artificial selection bias as the Booking.com challenge data set, the Twitter trips have a balanced ratio of 97 origin countries to 105 destination countries; however, most trips are domestic.

3.3. Preprocessing

Although more information would be available for the Twitter data set, we decided to use the same basic features in both data sets. The list of available features for each stay is as follows: user ID, checkin date, check-out date, user country, destination country, destination ID, and trip ID.

Table 1
Overview of the two data sets.

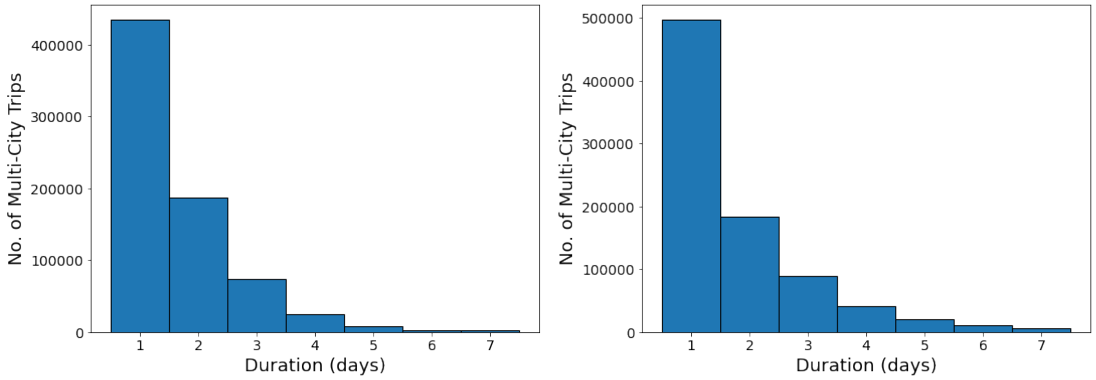
	Booking.com	**Twitter**
#Users	96,643	24,146
#Trips	734,102	852,131
#Origin Countries	5	97
#Destination Countries	107	105
Domestic trips	4.5%	91.3%
Date Range	Jan 2016 – Feb 2017	Oct 2010 – Jul 2021

Even though we have many trips available in both data sets, some travelers and destinations have an insufficient number of trips. Thus, we discard destinations and users with few trips to eliminate users and items still in a cold-start phase. For Booking, we keep travelers who have visited five unique destinations and destinations visited at least 15 times. In the more noisy Twitter data set, we enforce ten unique destinations per user, and each destination needs to be visited at least 25 times. We present the overview of the preprocessed data sets in Table 1. Of the total trips, 95.5% are international trips in the Booking.com data set, whereas 91.3% are domestic trips in the Twitter data set. The difference shows that the two data sets need to be handled and analyzed separately since the Booking.com trips are biased towards international trips. In contrast, the Twitter trips reflect that most travel is actually domestic.

An initial look at the distribution of the stay duration in Figure 1 shows that most travelers visit a given destination for a relatively short period, with the mode being 1 in both data sets.

3.4. Splitting into Training and Test Data

As is common in machine learning, we randomly split the stays into a training and testing set. The test set consists of 20% of the randomly chosen stays of each user. The remaining trips are used for feature engineering to extract advanced representations based on the preferences and mobility patterns of the users. These representations and other features are then used to train different statistical and machine learning models to predict the duration of stays. Consequently, the predictive performance is evaluated with unseen test data.



(a) Booking.com

(b) Twitter

Figure 1: Distribution of the duration of stay

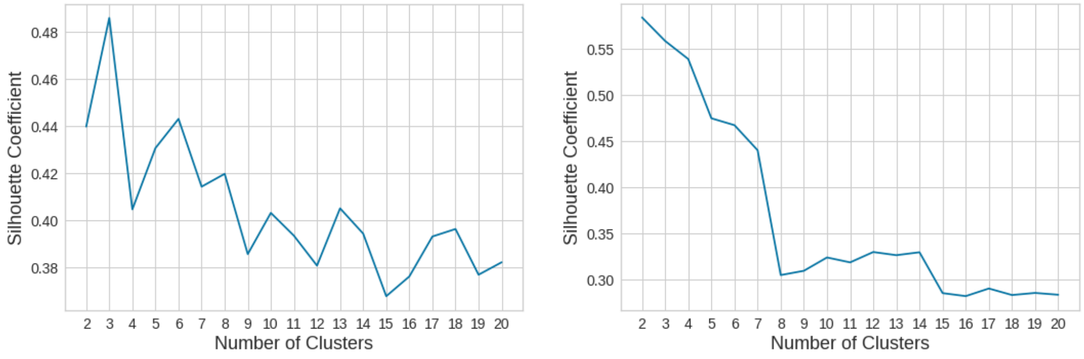
4. Feature Engineering

Nowadays, the performance of machine learning models heavily relies on the information the algorithms can use in the form of directly available but also derived features [21]. For this problem, we derive additional features based on the mobility patterns of the travelers: Using cluster analysis, we uncover types of travelers and compute graph embeddings of the destinations and users. In this way, we transform the information about travelers, trips, and destinations into a representation suitable for the machine learning algorithms to use this information to predict the duration of stays.

4.1. Traveler Types Characterization

The characterization of traveler types can be essential for personalized travel recommender systems. The objective is to discover the users' travel behavior based on their past check-in durations and frequencies. There is quite some research about traveler types [22, 23, 24, 25]; however, most relevant for our work is a clustering approach to identify traveler types by analyzing check-in data from location-based social networks [20], which we adopt to identify such groups of travelers that we use later to predict the duration of the next trip of a given user.

Cluster analysis is an unsupervised learning approach to discover the new groups such that members in the same groups are more similar to other data members in the same group than those in other groups. After extensive experimentation, we decided to use six features for the clustering task: number of domestic trips, number of unique domestic cities visited, mean duration of domestic trips, number of international trips, number of unique international cities visited, and mean duration of international trips. Since we use the Euclidean Distance, we normalize these features using min-max scaling. Using the K-Means algorithm to cluster the data, we separate users into groups in which the users are similar to each other, but the groups are different from each other. Since this algorithm requires the number of clusters to be specified, we run the K-means clustering algorithm by varying the number of clusters between 2 and 20 and assessing the cluster fit quality using the average silhouette score. The silhouette

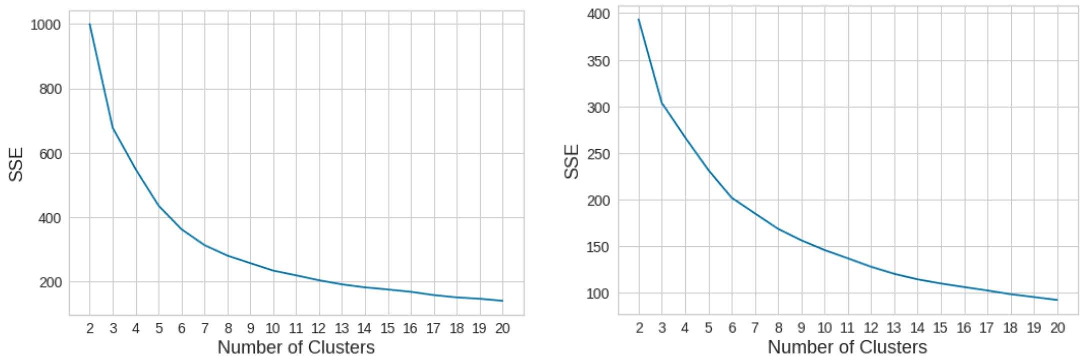


(a) Booking.com

(b) Twitter

Figure 2: Traveler Types Clustering - Silhouette Scores

score helps us understand how close each point in one cluster is to the neighboring clusters and, thus, acts as a suitable metric for assessing the number of clusters. The Silhouette score is normalized from -1 to 1, with 1 being a perfect score, i.e., a dense cluster well-separated from other clusters. A lower score indicates overlapping clusters with samples very close to the decision boundary of the neighboring clusters. We plot the two silhouette scores and the sum of squared errors (SSE) for the two data sets in Figure 2 and Figure 3. Based on these plots, we can infer that a solution of $K = 6$ clusters is suitable for both data sets, as there is a sharp decrease in the silhouette score for higher values.

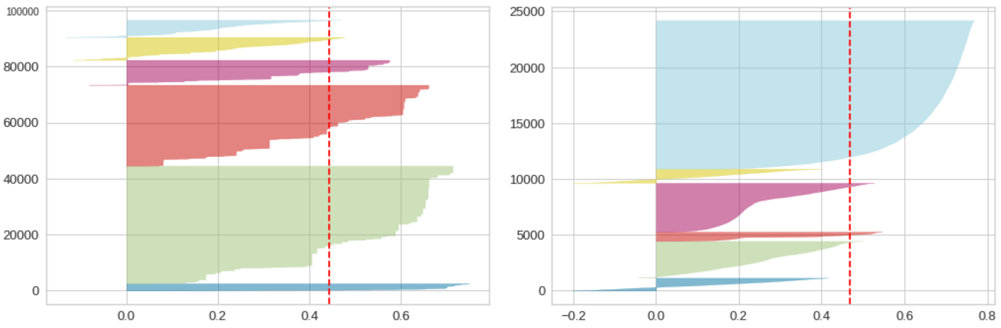


(a) Booking.com

(b) Twitter

Figure 3: Traveler Types Clustering - Sum of squared error (SSE)

The final silhouette plots for the discovered clusters for $K = 6$ of Booking.com and Twitter are presented in Figure 4. The six clusters each represent a distinct type of traveler that is not uniformly distributed. In the Booking.com data set, for example, the two large clusters, 1 and 5, represent the users who travel relatively rarely and take only short international trips with an average of 1.28 and 2.07 days. While for Twitter, the largest cluster represents the users that frequently travel to different cities in their country. To summarize, we have successfully



(a) Booking

(b) Traveler Types Clustering - Twitter

Figure 4: Traveler Types Clustering - Final Silhouette Plot

characterized the different traveler types in our data sets using the clustering approach. The characterization can help us predict the duration of future trips based on the traveler type label a user has.

4.2. Traveler Mobility Patterns

The analysis of human mobility patterns has a wide range of applications in several domains like urban planning, traffic forecasting, epidemic prevention, and location-based services [26]. Since our domain is strongly influenced by mobility, we derive several metrics based on the mobility patterns manifested in the users' trips. These patterns can give us insights into the users' traveling habits and reveal interconnections between the destinations. For example, it has been shown that coherent travel regions can be discovered from past user trips [27] and such features have been proven helpful in the Booking.com WSDM WebTour 2021 Challenge [28].

However, simply clustering the destinations into regions might not be helpful when dealing with the anonymized Booking.com data set, as we cannot map the destinations to previously identified regions. Also, this would lead to further scalability problems in the encoding since the number of dimensions will significantly increase using, e.g., one-hot encoding when training different machine learning-based models. Additionally, one-hot encoding treats the categorical variables as independent and does not consider their relationship. To retain relationship information between destinations, we automatically learn the representation of such categorical features in multi-dimensional spaces using entity embeddings [29].

4.2.1. Graph Embeddings

Several studies on learning latent representations of graphs have emerged in recent years. We can use embedding-based models to learn latent representations from complex data structures like graphs. These representations map each node in the graph into a low-dimensional space, providing insights into nodes' similarity and network structure. The embedding captures the semantics of the input graph by placing similar nodes close together in the embedding space. Such embeddings are applicable in many problems ranging from detecting protein-to-protein interaction in biological networks to friendship recommendations in social networks [30]. Therefore, we explore the idea of learning such latent representations of users' travel behavior using node embeddings to predict the duration of the next trip.

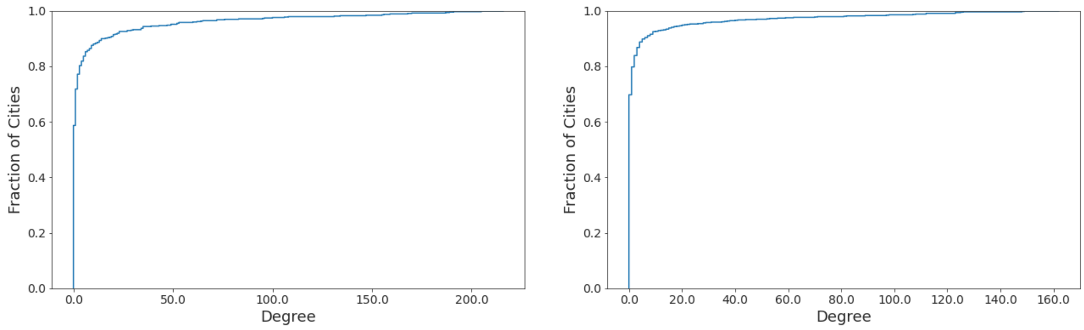
4.2.2. Destination Travel Graph Creation

We represent the destination cities as nodes in an undirected graph. To quantify the weight of the edges between two nodes, we use the ”traveled-together” relation, as our data sets comprise multi-destination trips. Therefore, we draw an edge between each pair of cities a user visits on the same trip. For example, if a user first visits New York City to Philadelphia and then Philadelphia to Washington D.C, we add an edge from New York City to Washington D.C. We argue that this additional transitive connection captures the underlying mobility better than omitting the information that the user visited these destinations within the same trip. The final weight of each edge is the number of co-occurrences of two nodes in the same trip for all trips in the data set. We account for repetition, i.e., if the same user took the same multi-destination trip twice during the observation period, it will be counted twice in the travel graph. To reduce the noise in the graph, we remove all the edges with a weight equal to one, i.e., only one user has traveled between these two destinations.

Table 2
Destination Travel Graph Metrics

	Booking.com	Twitter
#Nodes	5,046	3,523
#Edges	88,623	62,260
Density	0.0069	0.01

Some key metrics of the resulting graphs of the two data sets are tabulated in Table 2. Booking.com has slightly more nodes and edges, resulting in Twitter being slightly denser; however, both graphs are very sparse. Furthermore, analyzing the degree distribution depicted in Figure 5, one can see that there are relatively few high-degree nodes.



(a) Booking

(b) Twitter

Figure 5: Empirical Cumulative Distribution Function (ECDF) for the Destination Travel Graphs

4.2.3. Destination Embeddings

The goal of destination embedding is to encode the cities represented by nodes in the graph so that machine learning algorithms can predict and discover new patterns in such complex networks. Therefore, finding a method that can efficiently generate representations for these networks is essential. Popular methods like DeepWalk [31] and Node2Vec [32] use random walks to generate such embeddings, which is very suitable for capturing mobility patterns. Other deep learning methods like Structural Deep Network Embeddings [33] and Hierarchical Attention Networks [34] employ autoencoders and attention mechanisms, which we don't expect to add value due to the nature of our task and the size of available datasets and number of features.

Experimenting with DeepWalk and Node2Vec, we find these approaches do well at encoding the destination travel graphs we constructed in the previous section. Both software libraries transform each node in a graph into a vector by relying on direct encoding and use a decoder based on the inner product. The local context of nodes is captured using the statistics of random walks, i.e., random walks of size l try to simulate the context of a word with window size l used in SkipGram. DeepWalk runs fixed-length, unbiased random walks starting from each node, whereas Node2Vec uses flexible, biased random walks that can trade-off between local and global views of the network. It uses the parameter p to control the likelihood of the walk immediately revisiting a node, and q controls the likelihood of the walk revisiting a node's one-hop neighborhood. This additional feature made us choose Node2Vec to learn a low-dimensional destination representation in the two data sets. A fixed-length vector of size 100 represents each destination.

To create a visual overview, we map the learned Node2Vec city embeddings to 2-D space using t-distributed stochastic neighbor embedding (t-SNE) [35]. The color-coding in Figure 7 represents countries, although due to the anonymization of the Booking.com data set, we do not know which country is which. However, we have this information present in our Twitter data set. As one can see in Figure 7b, countries like USA and Canada, as well as Philipines and Myanmar, are close in embedding space, indicating awareness of geographical context despite that this information was not available to the algorithm. Further zooming in, we can also see that embeddings also capture the relative positions of cities well, as shown in an example with the cities in the Netherlands in Figure 6.

4.2.4. Traveler Embeddings

The previous section defined an approach to represent the destination; however, we can also extend the use of such embeddings to personalize the prediction of the duration of stay. For example, it is relatively common in natural language processing to create a sentence embedding using the weighted average of word embeddings [36]. Similarly, it is common to summarize a section of the graph in graph neural networks by averaging the node embeddings and using these embeddings for several downstream tasks later on [37]. We follow a similar idea to create traveler embedding that computes the average embedding of all the past cities the user has visited. The average embedding can help us personalize the recommended duration of stay for each user since it is a feature of their past travel behavior.

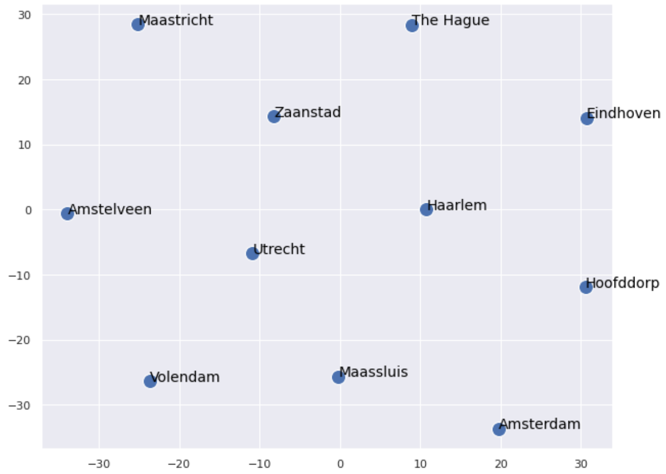


Figure 6: T-SNE Projection for City Embeddings for the Netherlands generated using Destination Travel Graph

5. Experiments

In this section, we describe how we evaluate the performance of various algorithms utilizing different settings of the embeddings as mentioned above against some statistical baselines.

5.1. Evaluation Settings

In our experiments, we use both Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure the performance of our different approaches [38]. Both these metrics have been widely used to assess the performance of such problems. RMSE does not use the absolute value, which makes it desirable in many cases, mainly when calculating the gradient or sensitivity for certain model parameters. Thus, combining both these metrics helps us assess the performance better.

The prediction task is to determine the number of calendar days at a destination, as this information is available in both data sets. It also means that all algorithms that output continuous values need to be rounded to integers since the durations in the test set are also integers. It would be misleading to include the decimal places when computing the evaluation metrics [39]. Therefore, we also analyze the effect of rounding on the overall performance of our models.

5.2. Baseline Methods

We compare the machine-learning-based approaches with four simple baselines. These comprise simple statistical measures and two variations of the method proposed by Dietz and Wörndl [11].

- **User Mean**: The mean value of the duration of all the past trips for a given user.
- **User Mode**: The mode value of the duration of all the past trips for a given user.

- **User Percentile - Country Level**: Computes the user's trip duration by comparing their mean percentile to the quantiles of all other travelers who have visited the same country as proposed in [11].
- **User Percentile - City Level**: Computes the user's trip duration by comparing their mean percentile to the quantiles of all other travelers who have visited the same city as proposed in [11].

5.3. Features Variations

To learn which features are useful for recommending the duration of stay, we evaluate three variations that take the above-discussed advanced representations and basic features as inputs. The basic features are trip type (domestic or international), user home country, and destination country. The three different variations of the representation of the advanced features are as follows:

- **Mobility One-Hot Encoded (M-OHE)**: a baseline machine learning-based approach that uses the conventional one-hot encoding of the booker and destination country instead of embeddings.
- **Mobility Global Embeddings (M-GE)**: uses city embeddings discussed in Section 4.2.3 instead of the booker and destination country.
- **Mobility Personalized Embeddings (M-PE)**: uses city embeddings as well as traveler embeddings discussed in Section 4.2.3 and Section 4.2.4, respectively.

Table 3
Features used in the different evaluation scenarios.

	M-OHE	M-GE	M-PE
Trip type	Y	Y	Y
Traveler type clustering	Y	Y	Y
User home country	Y	N	N
Destination country	Y	N	N
City embeddings	N	Y	Y
Traveler embeddings	N	N	Y

We summarize the features used in the above approaches in Table 3. We always use the trip and the traveler types as input features, as they are fundamental to the approach. The M-GE and M-PE implicitly encode the home and destination countries via the city embeddings; thus, we do not input this information once more to avoid redundancy.

5.4. Algorithms

To evaluate these variations of features and encodings, we use the decision trees from scikit-learn[5] and Gradient Boosting from CatBoost[6]. CatBoost is a state-of-the-art open-source library

[5] scikit-learn decision trees: https://scikit-learn.org/stable/modules/tree.html

[6] CatBoost library: https://catboost.ai

for Gradient Boosting on Decision Trees developed by Yandex, which has similar or better performance than most other Gradient Boosting libraries, even with default parameters [40]. We chose these algorithms to gain additional analytic insights into the importance and success of the different embedding strategies. Even though these tree-based models are long-established, their performance is on par with deep neural network approaches, and they are easier to tune [41]. To speed up the training process, we use GPU-accelerated learning in the Google Cloud Platform[7]. Finally, we use four-fold cross-validation to tackle overfitting and determine the best set of hyperparameters for each model towards the best RMSE value using `scikit-optimize` package, which employs a Bayesian search method [42]. We summarize the optimal parameters for the different models in Table 4.

Table 4
Optimal hyperparameters as determined using the Bayesian search method of `scikit-optimize`.

Data set	Approach	CatBoost				Scikit Decision Tree		
		itr	**l2_leaf_reg**	**lr**	**max_depth**	**max_depth**	**min_samples_leaf**	**min_samples_split**
Booking	**M-OHE**	1100	49	0.049	11	49	39	48
	M-GE	2100	50	0.024	12	40	39	48
	M-PE	2100	50	0.231	15	14	45	81
Twitter	**M-OHE**	1000	29	0.034	8	44	50	76
	M-GE	550	26	0.048	12	49	39	48
	M-PE	1050	30	0.046	15	24	42	71

The recommended duration of stay is the outcome of the machine learning regression models of the respective algorithms using the different feature configurations.

5.5. Discussion

This evaluation procedure comprises two data sets, four statistical baselines, and two algorithms with three encoding variations as independent variables. This setup results in 20 experimental variations for which we compute the evaluation using four dependent variables.

The main goal is to measure the difference between the different embeddings, providing insights into which input features are helpful and which are not. Furthermore, we are interested in whether more features improve the results in a significant way, as it might be that the training costs of tuning a model with more parameters might not be worth the effort due to diminishing improvements in accuracy. Finally, we emphasize that the input for all models is solely based on traveler mobility. We can further improve the embeddings in real-world scenarios with contextual info and metadata, typically available on commercial platforms.

6. Results

We evaluate the overall performance of both data sets. The MAE and RMSE values for all methods of the test set are presented in Tables 5 and 6 for the Booking.com and Twitter data sets, respectively.

[7] Google Cloud Platform https://cloud.google.com/

6.1. Booking.com

For the Booking.com data set, we can observe that our proposed approaches consistently outperform the baselines in terms of MAE and RMSE. Interestingly, the user mean duration provides worse results than the mode, which we attribute to the mode being more robust against outliers. Recall that one-day stays at a destination are most frequent across all trips. Furthermore, the user percentile approach [11] performs better when the aggregation target is on a country level instead of the city level. Analyzing the embedding-based methods, M-GE and M-PE perform better than the M-OHE approach, indicating the node embeddings are better in capturing travelers' mobility patterns than the conventional one-hot encoding of the travel destinations. The traveler embedding in the M-PE approach helps the model understand the user's preferences better than the M-GE approach, which only considers the city embeddings. In the direct comparison of the decision tree model, CatBoost performs better than the scikit-learn Decision Tree in the embedding-based encoding, with similar performance for one-hot encoding. We can also observe that the MAE Rounded is much lower than MAE, while RMSE Rounded is higher than RMSE. Thus, rounding the predicted values helps make more accurate recommendations for most users, except for outlier users who travel for more extended periods.

Table 5
Prediction error metrics for Booking.com. The best scores are bold with a dagger.

Approach		MAE	MAE Rounded	RMSE	RMSE Rounded
User Mean		0.835	0.821	1.168	1.214
User Mode		0.742	0.742	1.233	1.233
User Percentile – Country		0.726	0.726	1.185	1.185
User Percentile – City		0.769	0.769	1.209	1.21
M-OHE	Scikit-DT	0.678	0.6	0.955	1.01
	CatBoost	0.678	0.601	0.954	1.01
M-GE	Scikit-DT	0.545	0.483	0.787	0.837
	CatBoost	0.541	0.475	0.777	0.827
M-PE	Scikit-DT	0.563	0.491	0.804	0.853
	CatBoost	**†0.534**	**†0.466**	**†0.767**	**†0.815**

6.2. Twitter

We can not replicate the clear results of the Booking.com data set in the Twitter data set. First of all, the data set seems to be noisier since, without exception, all error metrics are higher compared to the Booking.com data. Further, our proposed methods only outperform the baseline in terms of RMSE and RMSE Rounded, while the baseline approach of simply taking the mode of the stay duration of the user achieves the lowest MAE score. We attribute this surprising result to the characteristics of check-in-based data as opposed to hotel bookings: If a user visits two or more cities on the same day, each destination will be recorded with a stay duration of one calendar day, even though it was only a few hours. This effect does not happen with the Booking.com hotel reservations, as one typically does not book two accommodations for the same night. Besides that, we observe similar trends for machine learning methods as for the Booking.com data set, with M-PE performing better than M-GE and M-OHE. Again, the best model in terms of the more outlier-robust RMSE metric is CatBoost using the M-PE encoding.

Table 6
Prediction error metrics for Twitter. The best scores are bold with a dagger.

Approach		**MAE**	**MAE Rounded**	**RMSE**	**RMSE Rounded**
User Mean		0.962	0.958	1.307	1.336
User Mode		**†0.806**	**†0.806**	1.49	1.49
User Percentile – Country		0.823	0.823	1.471	1.472
User Percentile – City		0.856	0.856	1.442	1.443
M-OHE	Scikit-DT	0.932	0.974	1.261	1.286
	CatBoost	0.931	0.975	1.259	1.283
M-GE	Scikit-DT	0.884	0.872	1.229	1.268
	CatBoost	0.882	0.876	1.222	1.262
M-PE	Scikit-DT	0.892	0.898	1.236	1.278
	CatBoost	0.844	0.820	**†1.183**	**†1.228**

6.3. Discussion

We tested ten different models to recommend the optimal duration of stay at a destination and evaluated them using two data sets. Our most relevant conclusion is that based on the performance of both data sets, we can see that more advanced representations based on traveler types characterization and mobility patterns improve the predictive performance regarding the duration of stay for travelers.

We obtained clear results from the Booking.com data sets that more sophisticated feature engineering efforts lead to increased predictive performance. We further confirm this in Table 7 with a brief ablation analysis where we revisit the impact of various features of the M-PE model in The values are computed using the CatBoost library and are based on the influence of the individual input features. We observe that both the embedding types play an important role in predicting the duration of the stay. The City Embeddings have the highest importance among all features, followed by Traveller Embeddings and Traveler Cluster. The Trip Type (domestic or

Table 7
Importance of the different features used in CatBoost M-PE model for the Booking.com and Twitter data sets

Feature	**Importance**	
	Booking.com	**Twitter**
City Embedding	44.84%	46.32%
Traveler Embedding	26.45%	44.77%
Traveler Cluster	28.60%	7.54%
Trip Type	0.09%	1.35%

international) feature does not play a role in the predictions, as both data sets are predominately formed of one trip type.

In the case of the Twitter data, the results of Table 6 underline the task's difficulty compared to simple baselines, such as taking the mode duration. Recall from Figure 1 that the distribution of the stays at individual destinations is negatively skewed, with about half of all stays being only one day. This circumstance makes it easy for the baselines to perform well. While this is roughly the same for both data sets, the most international (95.5%) hotel reservations from Booking.com are way less noisy than the mostly domestic (91.3%) mobility derived from Twitter users. For this reason, we see limited direct comparability between the two data sets and observe that the duration of hotel reservations is better suited to predict with the proposed algorithms than unconstrained mobility observed from a location-based social network.

When it comes to the generalizability of the results, we have strong doubts about the Booking.com challenge data set as stemming from a data science challenge; it is probably too clean to be a realistic benchmark for the challenge of recommending the personalized duration of stay. The Twitter data set is likely a closer reflection of reality, which is a sobering insight: Due to the reality of most trips in the data lasting precisely one day, a strategy of simply recommending one day – essentially the user mode approach – can be competitive with machine learning approaches.

7. Conclusions

In this paper, we explored how we can effectively leverage traveler behavior and mobility patterns to model the user's preferences when predicting the duration of stay at a destination. Using two data sets, one based on hotel bookings and one on check-in-based traveler mobility from location-based social networks, we evaluate machine learning algorithms with domain-driven feature engineering against statistical baselines. The unavailability of high-quality data impeded the feature engineering; however, we could automatically identify different destination embeddings even though the Booking.com data set was anonymized. Consequently, to the best of our knowledge, we are first to evaluate the duration of stays for destination recommendation systematically. Given the input features used to do the feature engineering, one can see the potential to generalize our approaches and experimental setup to any check-in-based data source.

In the future, we plan to extend this work to not only recommending the duration of stay at one destination but determining the optimal durations for all legs of composite trips in relation to each other [43, 9]. Furthermore, we are interested in exploring the benefits of additional methods to embed user mobility patterns and experiment with deep learning to optimize the prediction task's performance further.

Recommending the amount of item consumption is not only relevant on a destination-level, but could also be analyzed on a finer level, e.g., determining how long to stay at specific points-of-interest [44] or the duration of exercise activities [45]. We believe our approach could supplement existing approaches and establish an evaluation standard for this problem.

References

[1] Y. Koren, R. Bell, Advances in collaborative filtering, in: Recommender Systems Handbook, Springer US, 2015, pp. 77–118. doi:10.1007/978-1-4899-7637-6_3.

[2] M. F. Dacrema, P. Cremonesi, D. Jannach, Are we really making much progress? a worrying analysis of recent neural recommendation approaches, in: 13th ACM Conference on Recommender Systems, RecSys'19, ACM, New York, NY, USA, 2019, pp. 101–109. doi:10.1145/3298689.3347058.

[3] A. Singh, T. Joachims, Fairness of exposure in rankings, in: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 2219–2228. doi:10.1145/3219819.3220088.

[4] A. J. Biega, K. P. Gummadi, G. Weikum, Equity of attention: Amortizing individual fairness in rankings, in: The 41st International ACM SIGIR Conference on Research, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 405–414. doi:10.1145/3209978.3210063.

[5] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Multistakeholder recommendation: Survey and research directions, User Modeling and User-Adapted Interaction 30 (2020) 127–158. doi:10.1007/s11257-019-09256-1.

[6] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, M. Orgun, Sequential recommender systems: Challenges, progress and prospects, in: 28th International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2019. doi:10.24963/ijcai.2019/883.

[7] D. Herzog, L. W. Dietz, W. Wörndl, Tourist trip recommendations – foundations, state of the art and challenges, in: M. Augstein, E. Herder, W. Wörndl (Eds.), Personalized Human-Computer Interaction, de Gruyter Oldenbourg, Berlin, Germany, 2019, pp. 159–182. doi:10.1515/9783110552485-006.

[8] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, A survey on algorithmic approaches for solving tourist trip design problems, Heuristics 20 (2014) 291–328. doi:10.1007/s10732-014-9242-5.

[9] D. Herzog, W. Wörndl, A travel recommender system for combining multiple travel regions to a composite trip, in: CBRecSys, 2014, pp. 42–48.

[10] M. Xie, L. V. S. Lakshmanan, P. T. Wood, Composite recommendations: from items to packages, Frontiers of Computer Science 6 (2012) 264–277. doi:10.1007/s11704-012-2014-1.

[11] L. W. Dietz, W. Wörndl, How long to stay where? On the amount of item consumption in travel recommendation, in: ACM RecSys 2019 Late-breaking Results, 2019, pp. 31–35.

[12] P. Covington, J. Adams, E. Sargin, Deep neural networks for YouTube recommendations, in: 10th ACM Conference on Recommender Systems, RecSys'16, ACM, New York, NY, USA, 2016, pp. 191––198. URL: https://doi.org/10.1145/2959100.2959190. doi:10.1145/2959100.2959190.

[13] X. Yi, L. Hong, E. Zhong, N. N. Liu, S. Rajan, Beyond clicks: Dwell time for personalization, in: 8th ACM Conference on Recommender Systems, RecSys'14, ACM, New York, NY, USA, 2014, pp. 113–120. URL: https://doi.org/10.1145/2645710.2645724. doi:10.1145/2645710.2645724.

[14] Y. Tian, K. Zhou, D. Pelleg, What and how long: Prediction of mobile app engagement, ACM Transactions on Information Systems 40 (2022) 1–38. URL: https://doi.org/10.1145/3464301. doi:10.1145/3464301.

[15] D. Kitayama, K. Ozu, S. Nakajima, K. Sumiya, A Route Recommender System Based on the User's Visit Duration at Sightseeing Locations, Springer, Cham, 2014, pp. 177–190. doi:10.1007/978-3-319-11265-7_14.

[16] L. W. Dietz, Data-driven destination recommender systems, in: 26th Conference on User Modeling, Adaptation and Personalization, UMAP '18, ACM, New York, NY, USA, 2018, pp. 257–260. doi:10.1145/3209219.3213591.

[17] D. Goldenberg, P. Levin, Booking.com multi-destination trips dataset, in: 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2021. doi:10.1145/3404835.3463240.

[18] J. Melià-Seguí, R. Zhang, E. Bart, B. Price, O. Brdiczka, Activity duration analysis for context-aware services using foursquare check-ins, in: International Workshop on Self-aware Internet of Things, Self-IoT '12, ACM, New York, NY, USA, 2012, pp. 13–18. doi:10.1145/2378023.2378027.

[19] D. Herzog, C. Laß, W. Wörndl, Tourrec: A tourist trip recommender system for individuals and groups, in: 12th ACM Conference on Recommender Systems, RecSys '18, ACM, New York, NY, USA, 2018, pp. 496–497. doi:10.1145/3240323.3241612.

[20] L. W. Dietz, A. Sen, R. Roy, W. Wörndl, Mining trips from location-based social networks for clustering travelers and destinations, Information Technology & Tourism 22 (2020) 131–166. doi:10.1007/s40558-020-00170-6.

[21] L. Bernardi, T. Mavridis, P. Estevez, 150 successful machine learning models: 6 lessons learned at Booking.com, in: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, ACM, New York, NY, USA, 2019, pp. 1743–1751. doi:10.1145/3292500.3330744.

[22] H. Gibson, A. Yiannakis, Tourist roles: Needs and the lifecourse, Annals of Tourism Research 29 (2002) 358–383.

[23] J. Neidhardt, R. Schuster, L. Seyfang, H. Werthner, Eliciting the users' unknown preferences, in: 8th ACM Conference on Recommender Systems, RecSys '14, ACM, New York, NY, USA, 2014, pp. 309–312. doi:10.1145/2645710.2645767.

[24] M. Sertkan, J. Neidhardt, H. Werthner, Mapping of tourism destinations to travel behavioural patterns, in: B. Stangl, J. Pesonen (Eds.), Information and Communication Technologies in Tourism, Springer, Cham, 2017, pp. 422–434.

[25] M. Sertkan, J. Neidhardt, H. Werthner, Eliciting touristic profiles: A user study on picture collections, in: 28th ACM Conference on User Modeling, Adaptation and Personalization, UMAP '20, ACM, New York, NY, USA, 2020, pp. 230–38. doi:10.1145/3340631.3394868.

[26] M. C. González, C. A. Hidalgo, A.-L. Barabási, Understanding individual human mobility patterns, Nature 453 (2008) 779–782. doi:10.1038/nature06958.

[27] A. Sen, L. W. Dietz, Identifying travel regions using location-based social network check-in data, Frontiers in Big Data 2 (2019). doi:10.3389/fdata.2019.00012.

[28] D. Goldenberg, K. Kofman, P. Levin, S. Mizrachi, M. Kafry, G. Nadav, Booking.com wsdm webtour 2021 challenge, in: ACM WSDM Workshop on Web Tourism, WebTour'21, ACM, New York, NY, USA, 2021.

[29] C. Guo, F. Berkhahn, Entity embeddings of categorical variables, arXiv preprint arXiv:1604.06737 (2016).

[30] W. L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: Methods and applications, arXiv preprint arXiv:1709.05584 (2017).

[31] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 701–710.

[32] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 855–864.

[33] D. Wang, P. Cui, W. Zhu, Structural deep network embedding, in: International Conference on Knowledge Discovery and Data Mining, ACM, 2016. doi:10.1145/2939672.2939753.

[34] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, 2016, pp. 1480–1489. doi:10.18653/v1/n16-1174.

[35] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605.

[36] Q. Chen, Z.-H. Ling, X. Zhu, Enhancing sentence embedding with generalized pooling, in: 27th International Conference on Computational Linguistics, ACL, Santa Fe, New Mexico, USA, 2018, pp. 1815–1826. URL: https://aclanthology.org/C18-1154.

[37] S. Ivanov, E. Burnaev, Anonymous walk embeddings, in: J. Dy, A. Krause (Eds.), 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 2186–2195.

[38] T. Chai, R. R. Draxler, Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature, Geoscientific Model Development 7 (2014) 1247–1250.

[39] W. Wang, Y. Lu, Analysis of the mean absolute error (mae) and the root mean square error (rmse) in assessing rounding model, in: IOP conference series: materials science and engineering, volume 324, IOP Publishing, 2018.

[40] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, CatBoost: unbiased boosting with categorical features, Advances in neural information processing systems 31 (2018).

[41] R. Shwartz-Ziv, A. Armon, Tabular data: Deep learning is not all you need, Information Fusion 81 (2022) 84–90. doi:10.1016/j.inffus.2021.11.011.

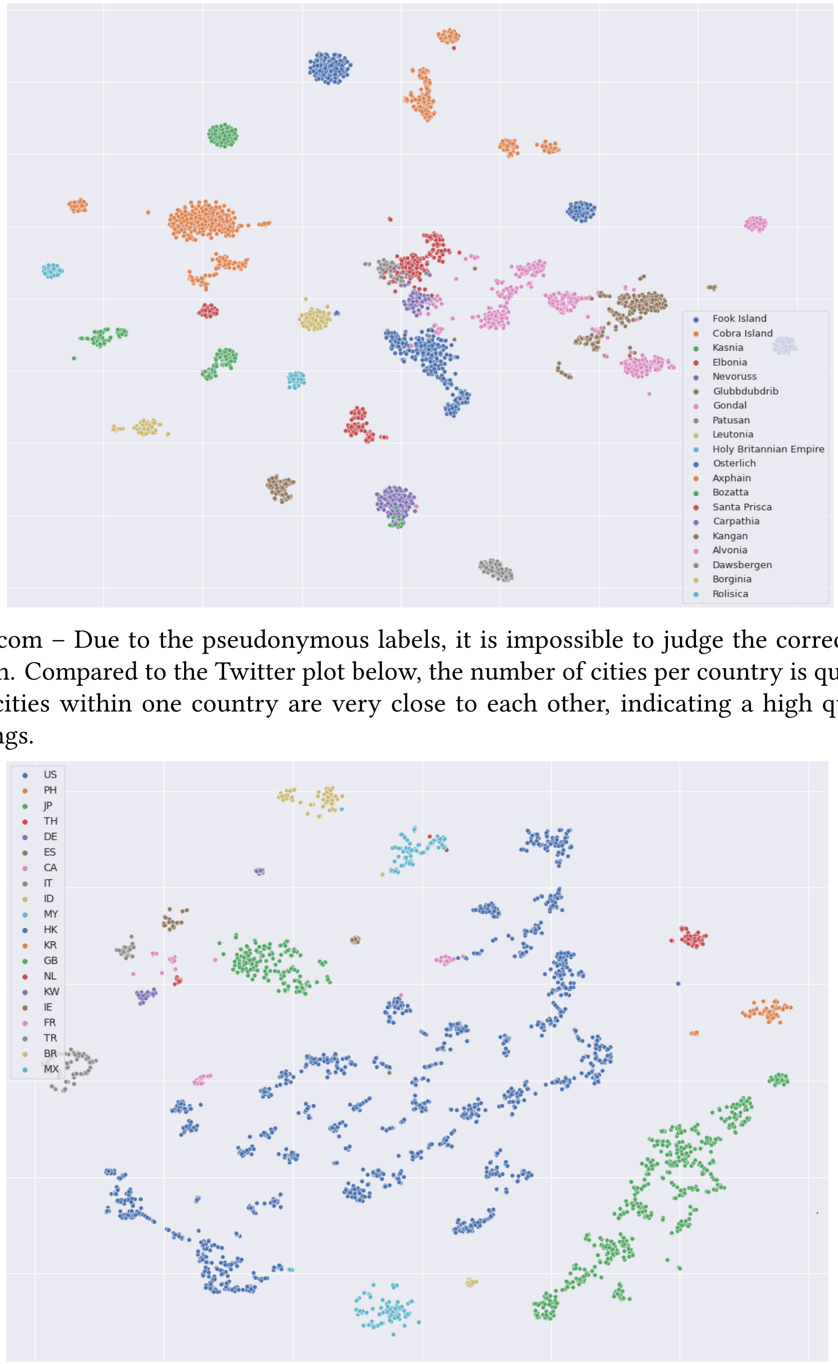
[42] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, I. Guyon, Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020, in: H. J. Escalante, K. Hofmann (Eds.), NeurIPS 2020 Competition and Demonstration Track, volume 133 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 3–26.

[43] R. Roy, L. W. Dietz:, Triprec – a recommender system for planning composite city trips based on travel mobility analysis, in: ACM WSDM Workshop on Web Tourism, WebTour'21, ACM, New York, NY, USA, 2021.

[44] D. Herzog, Recommending a sequence of points of interest to a group of users in a mobile

context, in: 11th ACM Conference on Recommender Systems, RecSys ’17, ACM, New York, NY, USA, 2017, pp. 402–406. doi:10.1145/3109859.3109860.

[45] B. Smyth, A. Lawlor, J. Berndsen, C. Feely, Recommendations for marathon runners: on the application of recommender systems and machine learning to support recreational marathon runners, User Modeling and User-Adapted Interaction (2021). doi:10.1007/s11257-021-09299-3.



(a) Booking.com – Due to the pseudonymous labels, it is impossible to judge the correctness of the projection. Compared to the Twitter plot below, the number of cities per country is quite balanced, and the cities within one country are very close to each other, indicating a high quality of the embeddings.

(b) Twitter – Here we have the true labels for the countries and cities. Unsurprisingly, most data is from the United States, followed by Great Britain. In the top center, one can observe the proximity of Canada to the US (pink and blue clusters).

Figure 7: T-SNE Projection of City Embeddings generated using Destination Travel Graphs