IDS 561 Project Report – Hotel Recommendation System

Team Members:
Arjit Jain
Shikha Dubey
Siddhant Yadav

Introduction:

Expedia wants to provide personalized hotel recommendations to their users. Expedia uses search parameters to adjust their hotel recommendations, but there isn't enough customer specific data to personalize them for each user. The project requires to contextualize customer data and predict the likelihood a user will stay at 100 different hotel groups.

Data:

Expedia has provided you logs of customer behavior. These include what customers searched for, how they interacted with search results (click/book), whether the search result was a travel package. The data comprises of train data, test data, and a destination information data set. Expedia is interested in predicting which hotel group a user is going to book. Expedia has in-house algorithms to form hotel clusters, where similar hotels for a search (based on historical price, customer star ratings, geographical locations relative to city center, etc.) are grouped together. These hotel clusters serve as good identifiers to which types of hotels people are going to book, while avoiding outliers such as new hotels that don't have historical data. The goal of the project is to predict the booking outcome (hotel cluster) for a user event, based on their search and other attributes associated with that user event. The train and test datasets are split based on time: training data from 2013 and 2014, while test data are from 2015. destinations.csv data consists of features extracted from hotel reviews text.

train/test.csv Column name	Description	Data type
date_time	Timestamp	string
site_name	ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp)	int
posa_continent	ID of continent associated with site_name	int
user_location_country	The ID of the country the customer is located	int
user_location_region	The ID of the region the customer is located	int
user_location_city	The ID of the city the customer is located	int
orig_destination_distance	Physical distance between a hotel and a customer at the time of search. A null means the distance could not be calculated	double
user_id	ID of user	int
is_mobile	1 when a user connected from a mobile device, 0 otherwise	tinyint
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	int
channel	ID of a marketing channel	int
srch_ci	Checkin date	string
srch_co	Checkout date	string
srch_adults_cnt	The number of adults specified in the hotel room	int
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room	int
srch_rm_cnt	The number of hotel rooms specified in the search	int
srch_destination_id	ID of the destination where the hotel search was performed	int
srch_destination_type_id	Type of destination	int
hotel_continent	Hotel continent	int
hotel_country	Hotel country	int
hotel_market	Hotel market	int
is_booking	1 if a booking, 0 if a click	tinyint
cnt	Numer of similar events in the context of the same user session	bigint
hotel_cluster	ID of a hotel cluster	int

Tools/Algorithms:

The project is done in Apache Spark with Python (PySpark, Pandas) as the primary language. The data required cleaning initially so that it can be used to get meaningful insights. For each user activity a hotel cluster (hotel cluster ID) will be found using content based filtering and top hotel cluster will be predicted based on their ranks.

Analysis:

The packages “sql”, “mlfeature”, “stringindexer” and “classification” is used for the analysis. The big data set comprising of roughly 37.5 million is imported using spark.read.csv command.

The irrelevant columns were removed through initial eyeball analysis. For applying content based filtering the data frame was vectorized. The date and time columns were used to generate a new column “num_OfDays” which reflected the total number of days booking was requested.

The “srch_destination_id” was grouped and its total count for each calculated. The count reflects the number the times the “srch_destination_id” was clicked.

The weights were added to the columns, 1.25 for “is_booking” and 0.684 for “count”. To create further differentiation the two, an additional weight of 0.0005 was assigned for “num_OfDays”. A partition was applied to “srch_destination_id” and it was sorted in descending order by “weight”. A new column for rank was created. The top 5 hotel clusters were chosen based on their ranking. The column values of “hotel_clusters” were vectorized as per rank for corresponding “srch_destination_id” using a collection set. This was the intermediate result.

The data frame is converted into Pandas for further processing. The test data was loaded using PySpark. The test data frame was inner joined on “srch_destination_id” with the intermediate result to get the result containing the “user _id” with corresponding recommended “hotel_clusters”.

Results:

The result will comprise of 5 (if 5 are available) hotel cluster ID for each of the user event based on their search patterns, there might be a common hotel cluster ID for several IDs since the user search patterns can be similar.

Future Scope:

For the new 14,500 “srch_destination_id” the “hotel_clusters” can be predicted using machine learning algorithms such as KNN or logistic regression. Also, parallel/ distributed computation could be used for faster processing.