Capsone Project Summary: Predicting New and Future Hotel Performance Using R

Author: Anna Barbayeva

Academic Advisor: Gergely Daroczi

19.06.2019

Background

The project was implemented for a large operator of hotels, condotels, resorts, serviced suites, etc., spanning more than 18,500 rooms and apartments under management for property owners in Indonesia, the Philippines and Malaysia.

Main Objectives

The original main objective of the project was to predict a new, future, or unopened hotel performance for business development and planning purposes. The main steps were identified: data collection for one location, modelling, and scaling up the developed algorithm for more locations. APIs were considered to be the primary source of data.

However, the project developed unexpectedly: the API access could not be obtained for various reasons and therefore the project objectives had changed. The following steps were identified in order to proceed with the project:

- Collect detailed data about all hotels in the location of interest
- Identify the competitor set for the hotel of interest
- Make projections of the price and market share
- Testing on a different location

Data Collection

The original approach was to use APIs as a primary source of data and this option was explored. Such options as EPS Rapid, SkyScanner, TripAdvisor, and Google APIs were considered, however only the latest has proved to be accessible.

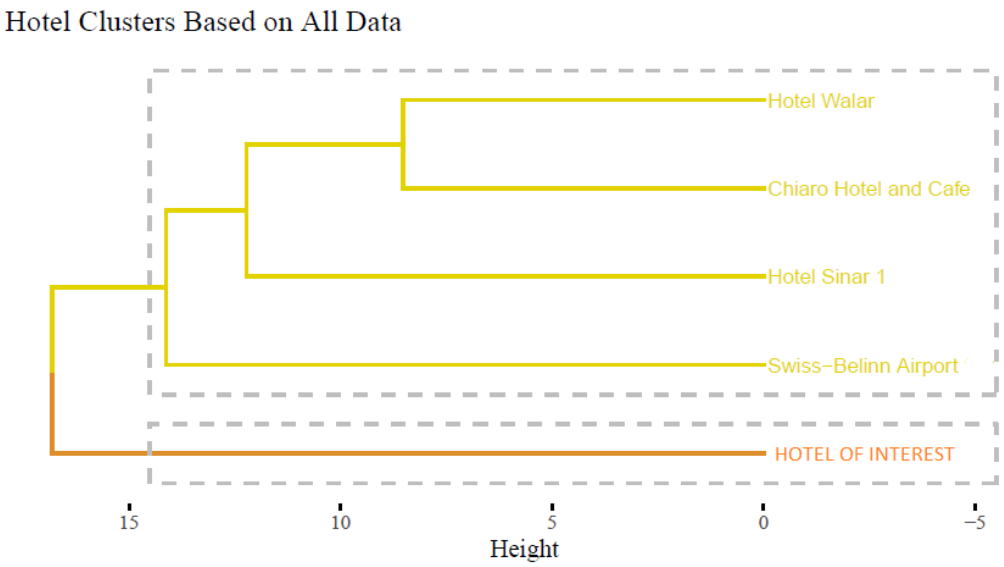
Therefore, data-scraping and data-cleaning functions were developed for one location. Such websites as *Google.com*, *TripAdvisor.com*, and *Agoda.com* were scraped. Scraped data has proved to be consistent, but, unfortunately, unlikely to be merged. The following features were obtained: rating and number of reviews (from each data source), number of stars, number of rooms, average price range, hotel and room amenities, room types, etc. Such limitations to scraping were considered: unstability, biased sample selection, legal issues. Overall, the scraped data serves as a valid prototype of the potential competitor data.

The scraping algorithm was tested on a different location and has proved to be stable and consistent since it ran with no errors and generated expected datasets.

Identifying Competitors

Hierarchical clustering was used in order to identify competitors of a given hotel. This method allows to control for how similar a competitor should be to the hotel of interest and works well on the small datasets.

Separate sets of potential competitors were identified for each data source (*Google.com*, *TripAdvisor.com*, and *Agoda.com*). Merging scraped data has resulted in a dataset of just few (less than 5) possible competitors, which makes clustering pointless. However, even such result makes it possible to check whether a new hotel of interest is similar to the ones in the area. The dendogram below provides an example for that:



This approach was also tested on a different location and yeilded similar results.

Price Projection

Regression analysis was conducted to estimate price level which would comparable with the competitors'. Only *TripAdvisor.com* provided the average price ranges for properties, therefore only that dataset was used in the modelling. Unfortunately, no significant results were generated due to the low number of observations.

As an alternative solution, it was suggested to define a price level by taking an average of all hotels that belong to the same cluster as the hotel of interest (by cluster, I refer to the clusters identified in the previous section).

Conclusion

Even though the project's focus has been shifted from modelling the market and predicting single hotel's performance to developing algorithm to obtain data, validating it, and checking whether obtained data can be used for the original purposes, the project has brought value to the company.

Such complications as the data access, scraping instability, and small sample size were encountered and future developments, such as developing more precise scraping and scaling up database were suggested.

.

.

.

.

.

.

.