Analysis of Pricing Strategy of Online Travel Agent

Heng-An Cheng(鄭恆安)
The Affiliated Senior High School of National Taiwan Normal University

Abstract

This project aims to analyze the pricing strategy of a hotel based on the data collected from the online travel agent (OTA) platform. First, we develop a python-based program to automatically crawl the room rates everyday for building the database. Next, we use data visualization to identify key factors that affect the room rates. Then, we choose the key factors as inputs of a long short-term memory (LSTM) model and use the data collected from a hotel to train the LSTM model. Finally, we utilize the LSTM model to predict the room rates for a special room type between the reservation date and the target check-in date based on the historical room rates observed over the past 30 days before the reservation date.

Motivation & Purpose

OTAs, such as Hotels.com, Agoda, Booking.com, generally work on two models to make profit. That is Merchant Model and Agency Model. In order to make profit, OTAs adjust room rate using machine learning technology, according to the conditions of each market, and then obtain future market supply and demand. Thus, we began to find factors causing the price up and down and use machine learning to predict future price trend.

Introduction:

OTA

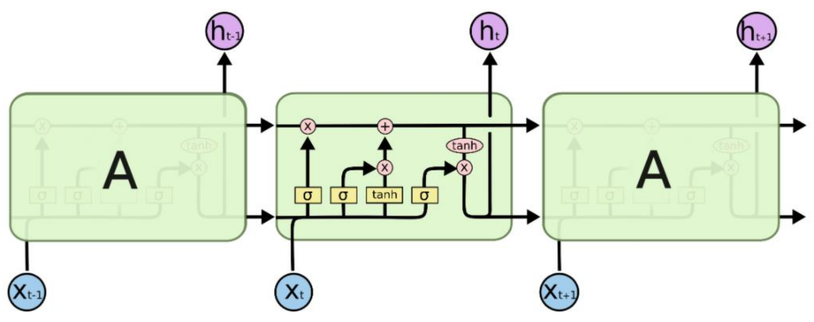
"OTA" stands for Online Travel Agency, which is a travel agency whose primary presence is on digital channels. Consumers can use a website and/or mobile device to search and book travel -- all without the traditional "gatekeeper" travel agent. OTAs connect to the full breadth of travel providers, giving travelers access to all of the inventory that they may want for their next trip.

Web Crawl

Crawl is a term describing a bot, script, or software program that visits a web page and grabs content and links from it. Once completed, the program visits the next link in the list or a link obtained from the web page it had recently visited.

LSTM

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems such as handwriting recognition, image classification, and price prediction.



Flow Chart(process):

Pre-process data format for the selected OTA website

Create hotel database

Crawl room rates for each hotel and store the results to the hotel database everyday

Analyze the data and identify features that affect the room rate

Train LSTM

Predict room rate

Process:

Build room rate database

Since there isn't any opensource room rate data online, we have to build our own database. To meet our goal, we use python requests module to crawl the web page from hotels.com. After that, we use BeautuifulSoup module to parse the web page and choose the room rate of different room types and hotels every day.

Data visualization analysis

According to previous researches, we find out some main factors that influence room rate. Therefore, we plot the data on charts to make it clear to find the correlation between each factor and the room rate. For example, through Fig1, we can discover the lowest price doesn't always occur on the earliest date you search, which means the interval may be one of factors.

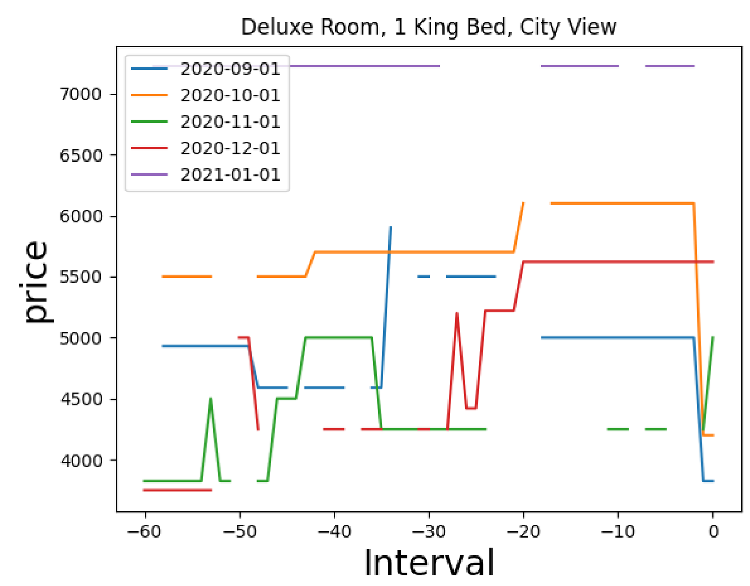


Fig 1. Different check-in date, the same interval between search date and check-in date

Train LSTM model

1.Data preprocessing : Because there is some missing data in our database, so we use interpolation to correct database.

2.Create features: we find out several factors through data visualization analysis. Check-in date(year, month, day, day of week, holiday), room type, interval between search date and check-in date, and the room rate searched on the search date.

3.Create train data : First, we normalize our data by using min max normalization. Then, we use previous 30 days data to predict tomorrow room rate.

4.Build model : Fig 2

5. Test model : Evaluate the mean square error .

6. Predict : Fig3

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_2 (LSTM)                (None, 30, 128)           70144
_________________________________________________________________
dropout_2 (Dropout)          (None, 30, 128)           0
_________________________________________________________________
lstm_3 (LSTM)                (None, 128)               131584
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                4128
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 33
=================================================================
Total params: 205,889
Trainable params: 205,889
Non-trainable params: 0
```

Fig 2. model summary

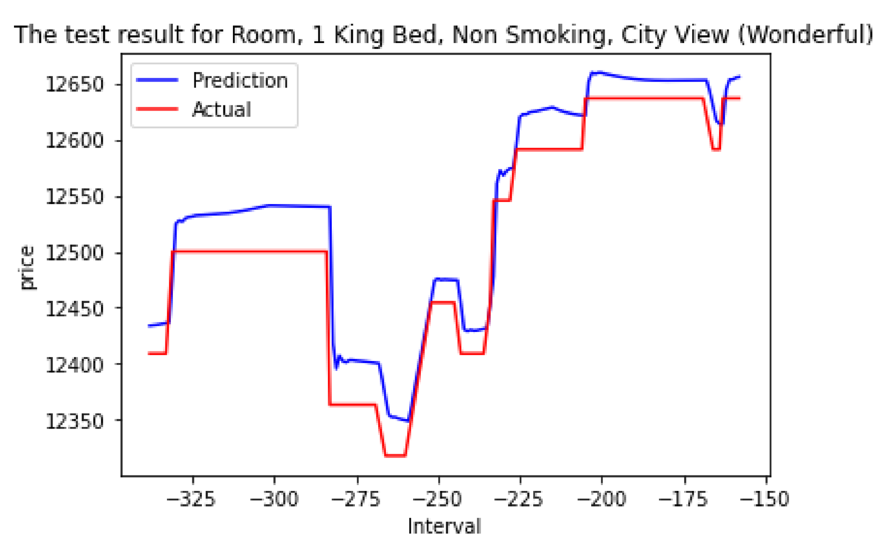


Fig 3. prediction and actual room rate