Pricing Recommendation System

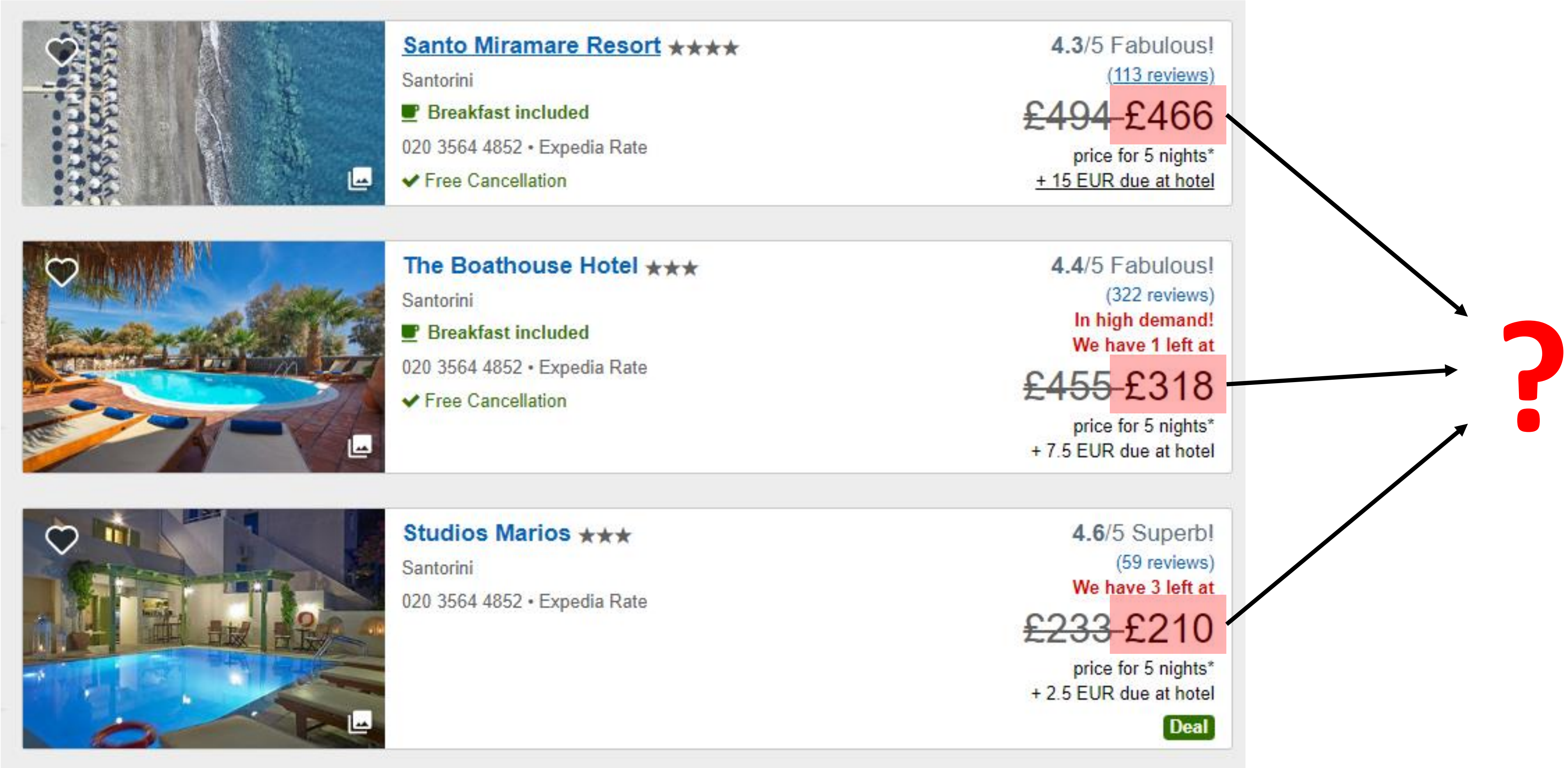
Made by HideInSmoke

OUTLINE

1. **Problem Definition**
    - Overview of Pricing Model
    - Demand Curve
2. **Preliminary Analysis**
    - Some problems
    - Data augmentation
3. **Feature engineering**
    - Temporal features - seasonality
    - Local demand features (K-means)
4. **Booking probability model based on Random Forest/Gradient Boosting Machine**
    - Under-sampling and train test split
    - Grid Search with 3-folds cross validation
    - Results comparison
5. **Discussion**
    - Limitations
    - Further work

Problem Definition

What are the best prices to be displayed here?



Overview of Pricing Model

Booking Probability Model

Estimates the booking probability for each room-night.

↓

Pricing Regression Model

Suggests price for available calendar nights of a hotel room based on the output from the first model.

↓

Personalization

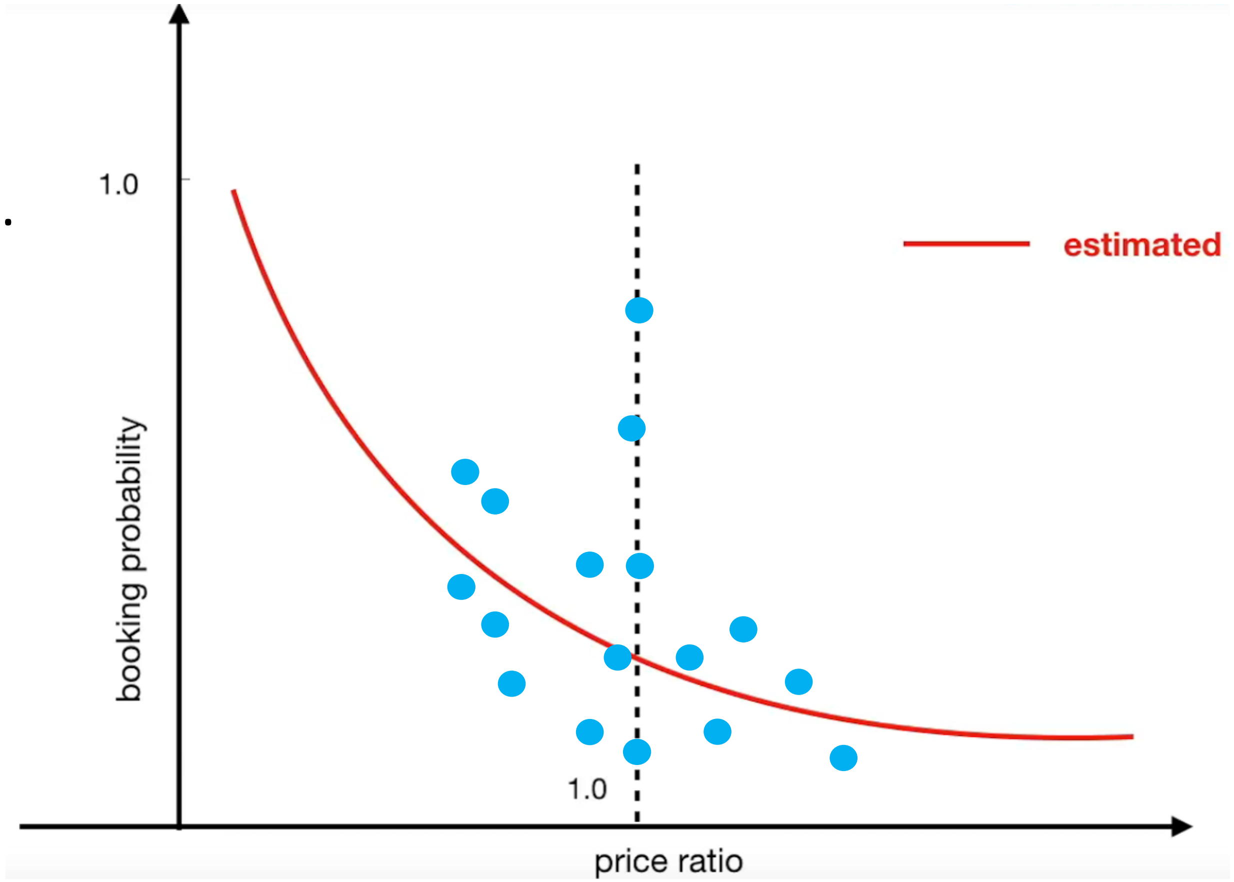
Adjusts price to incorporate hotel's requirements, special events, *etc.* based on the output from the second model.

Demand Curve

- Booking probability model can be used to get some sample points for each hotel/room.
- Then, we can estimate a curve for each hotel/room, such as

$$\boldsymbol{F}(p) = a^{b/p} - 1 \ (p \geq 0)$$

- The optimal price can be founded by maximizing $\boldsymbol{F}(p) * p$.



Preliminary Analysis

Some problems

- **search_date – Jan. 2015**

- **arrival/departure – Jan. 2015 - Apr. 2016**

- **Missing values** – There are some missing values in *hotel_feature_1* and *hotel_feature_2*, I drop the rows with missing value as the number is relevant small.

- **Negative values** – There are some negative values in *hotel_price*, so I convert them into positive values by using abs().

- **Data is imbalanced** – The ratio of bookings vs non-bookings is around 7:1000. And size of minority class is 309 observation so it is not possible to train an advanced model. Therefore, I used a rule-based data augmentation to generate more minority samples. I also tried training the model using class-weights.

Data augmentation

Why?

The ratio of bookings vs non-bookings is around 7:1000. The size of minority class is 309 observation, so it is too less to train a proper model.

How?

Input:

search_date	arrival	departure	adults	children	search_id	hotel_id	h_price	is_promo	h_f_1	h_f_2	h_f_3	h_f_4	h_f_5	booked
26/01/2015	06/04/2016	08/04/2016	2	0	1	517	2077.95	0	64.49031	85	9	0	0	1

Output:

26/01/2015	06/04/2016	07/04/2016	2	0	1	517	2077.95	0	64.49031	85	9	0	0	1
26/01/2015	07/04/2016	08/04/2016	2	0	1	517	2077.95	0	64.49031	85	9	0	0	1

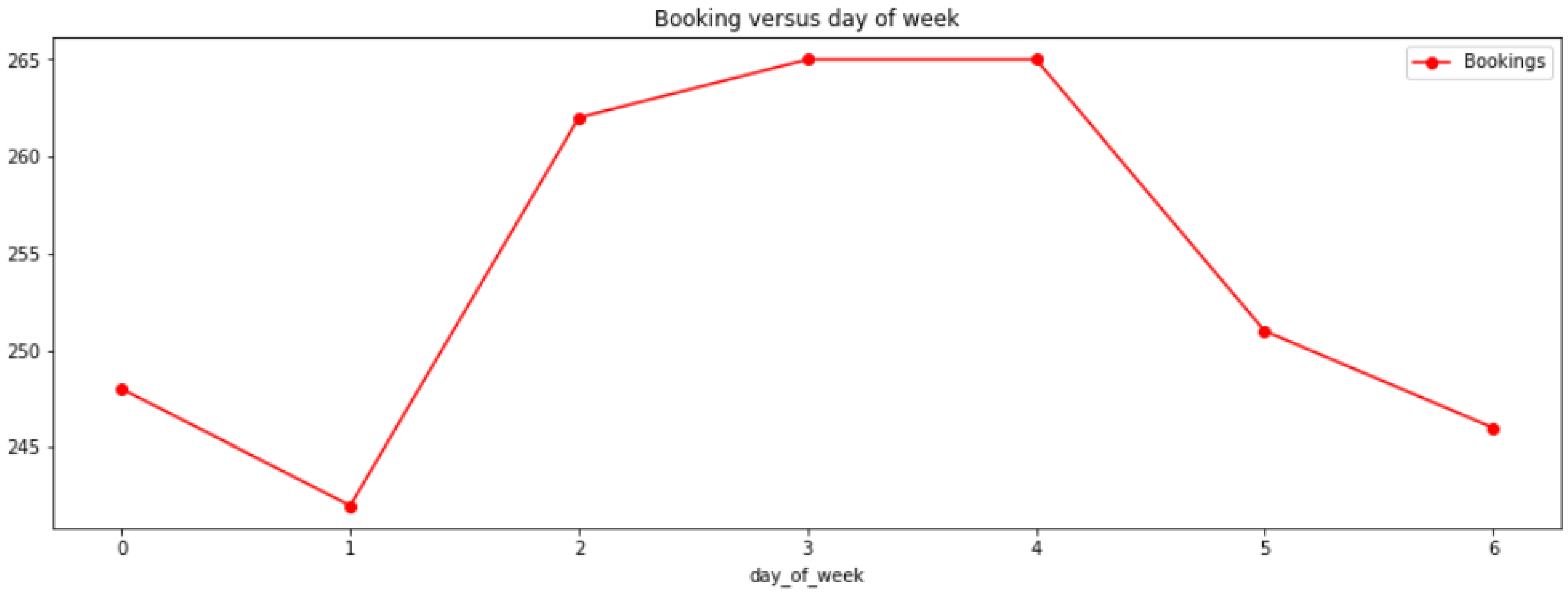
Consider the row as a request when booked = 0, consider the row as a booking when booked = 1.

Feature engineering

Temporal features - Day of week

- ***Weekday vs. booking***

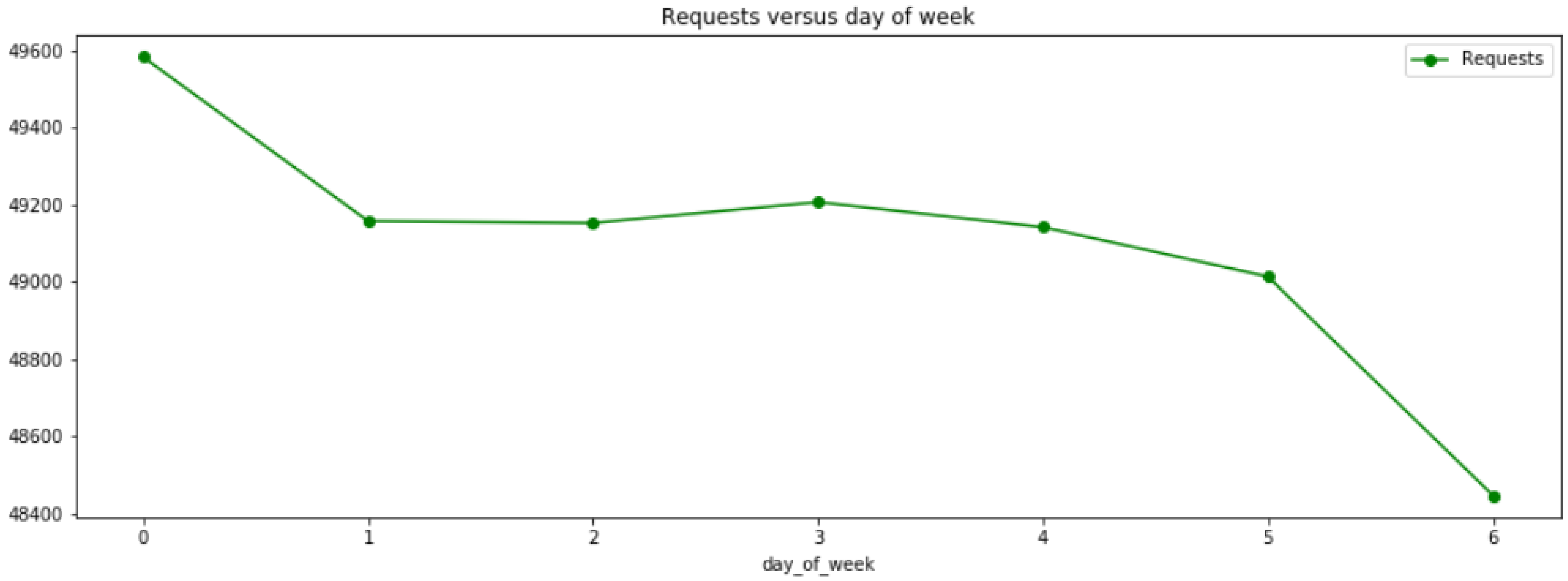
  Thursday and Friday have slightly more bookings.



Temporal features - Day of week

- ***Weekday vs. request***

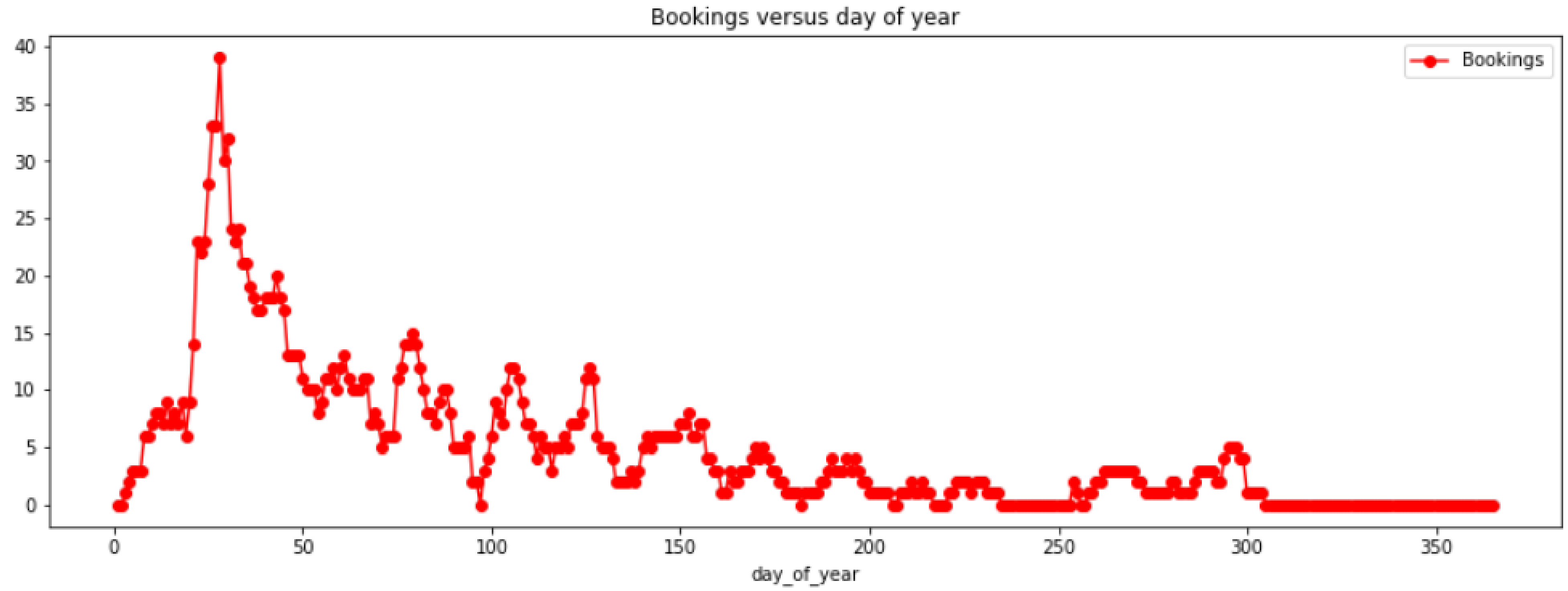
  Monday has slightly more requests.



Temporal features - Day of year

- ***Year day vs. booking***

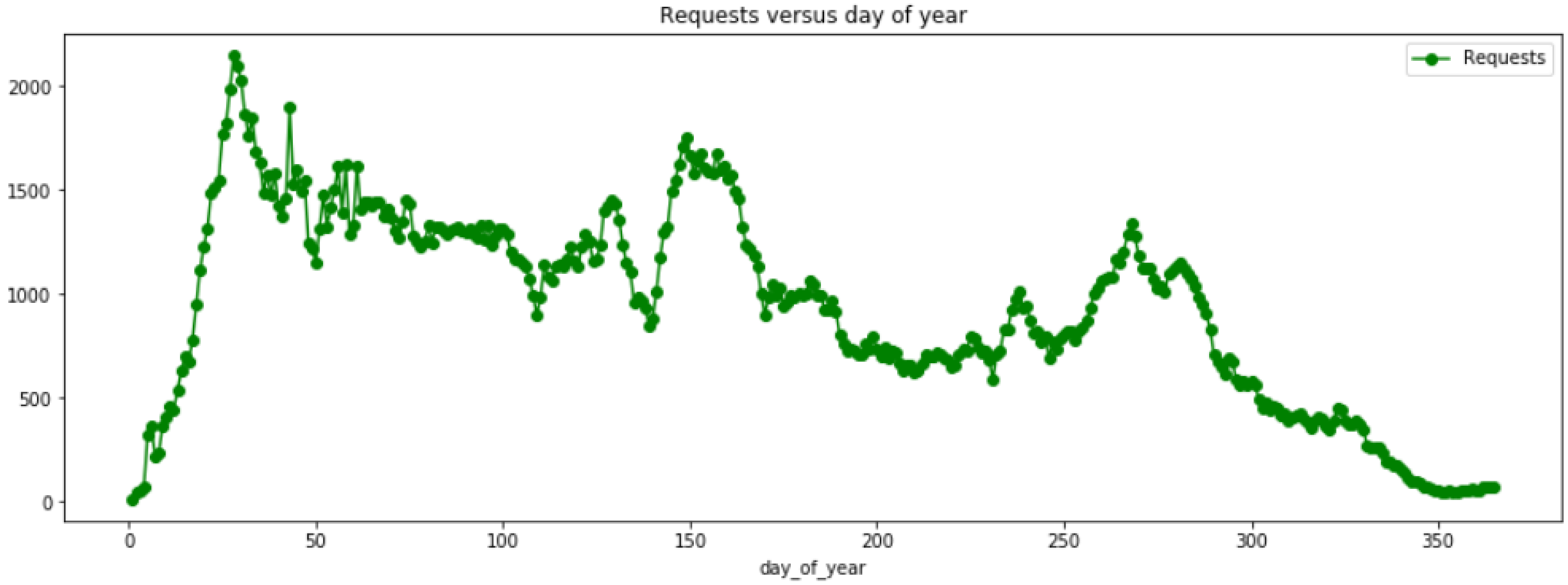
  As the data is for Jan, so more bookings are made for next month.



Temporal features - Day of year

- ***Year day vs. request***

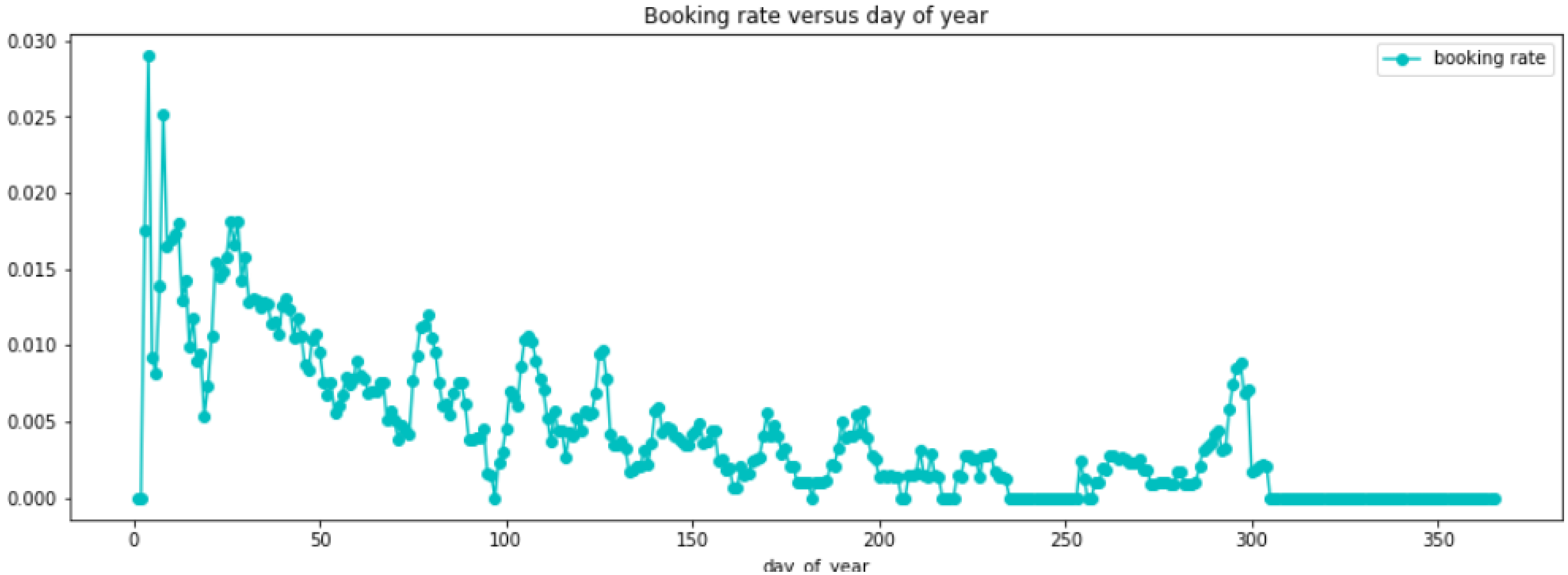
  There are some **seasonality**.



Temporal features - Day of year

- ***Year day vs. booking rate***

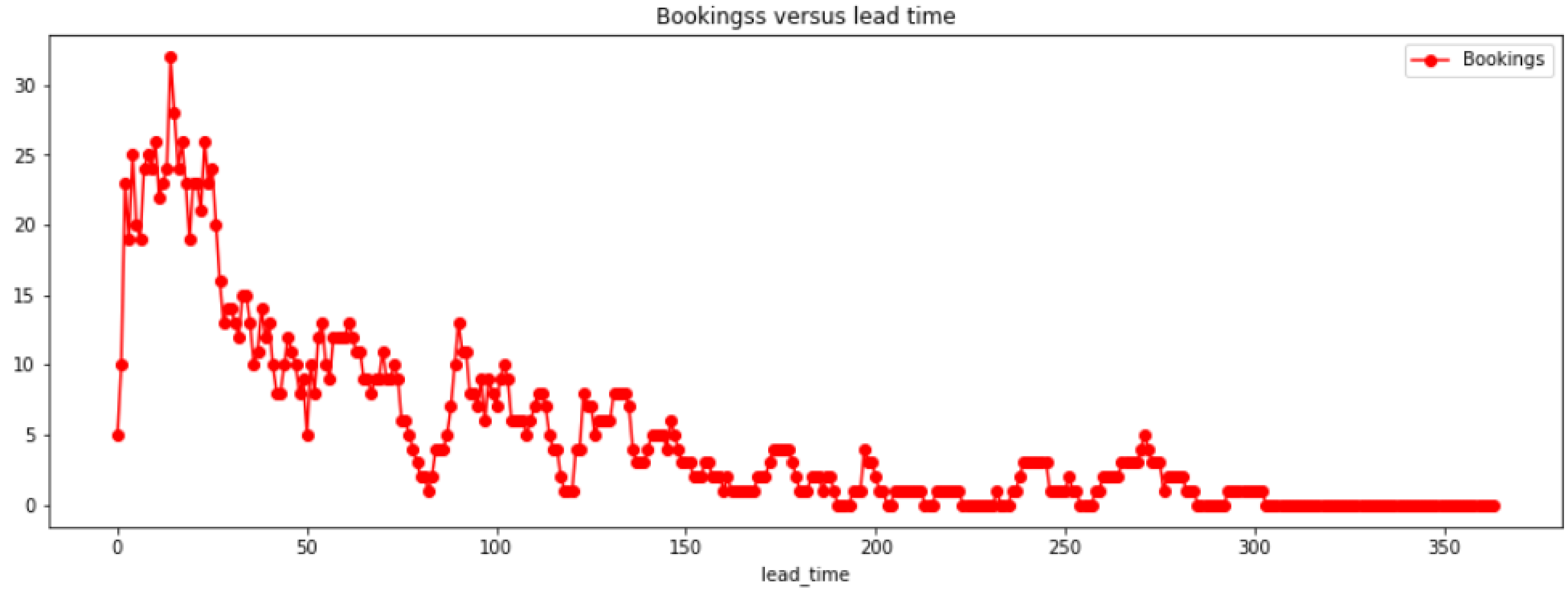
  Higher booking rate in the first quarter of year 2015 (it is reasonable as the data is for Jan).



Temporal features – Lead time

- ***Lead time vs. booking***

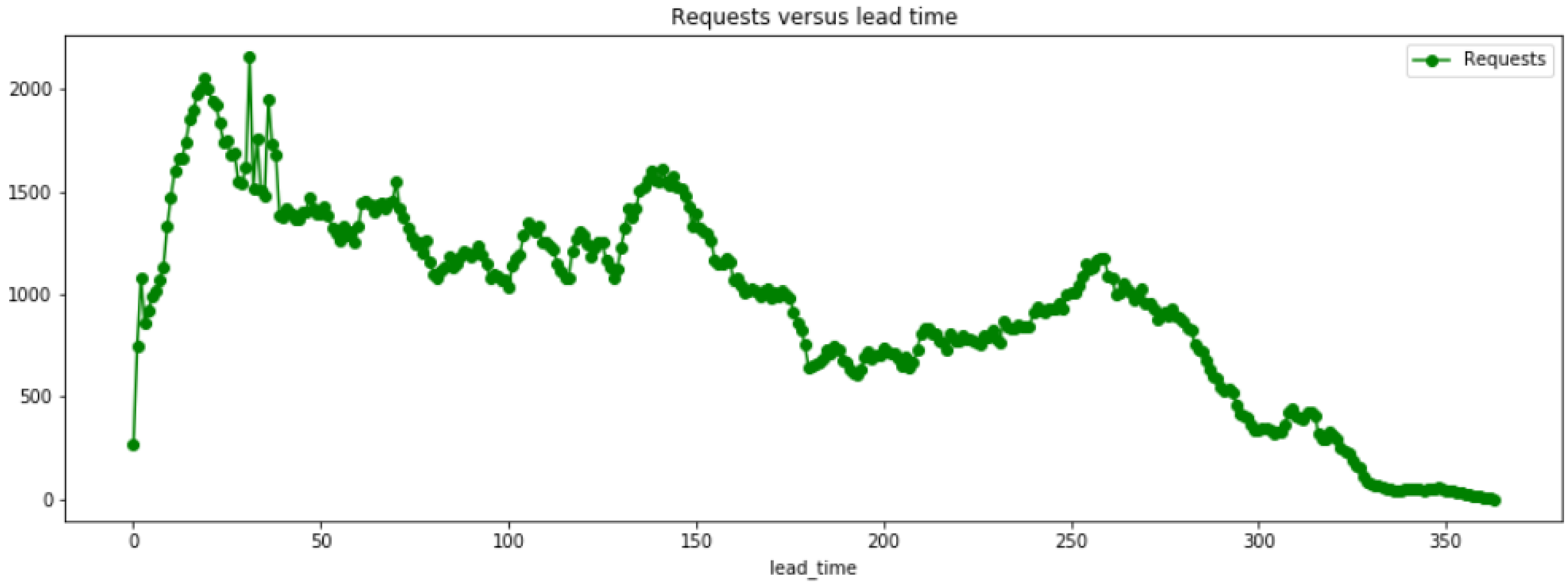
  Users normally book hotel one moth in advance.



Temporal features – Lead time

- ***Lead time vs. request***

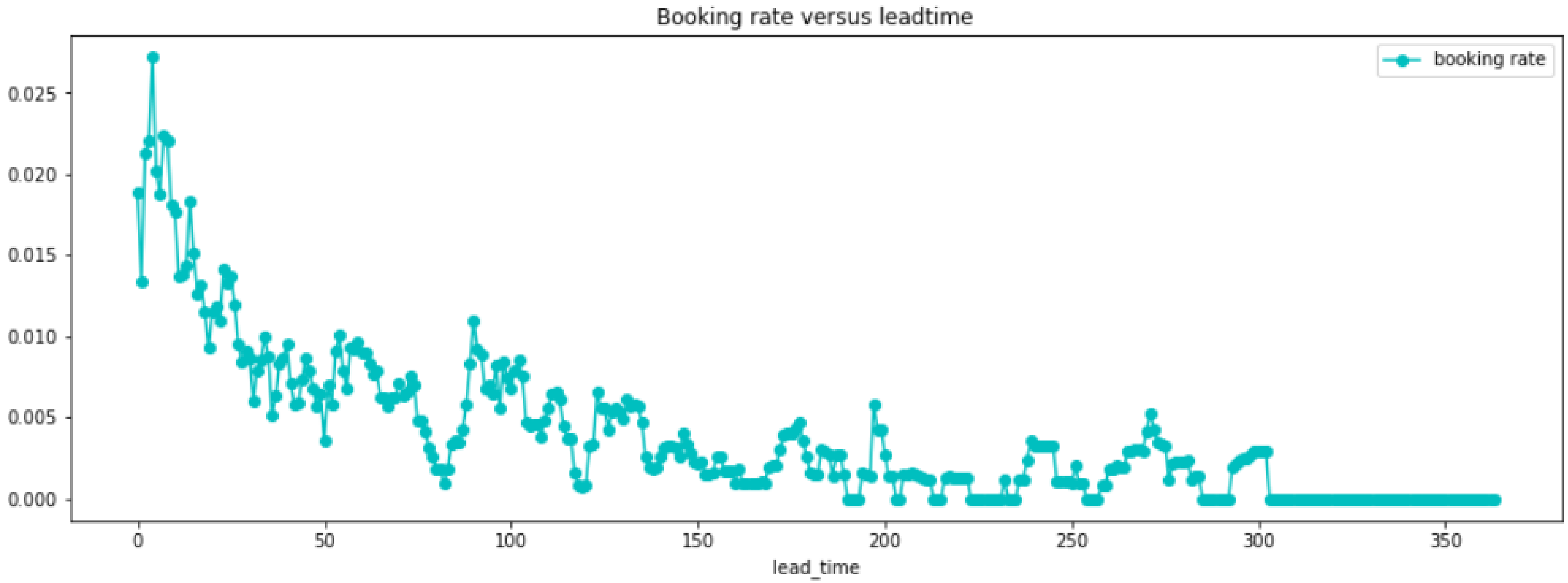
There is a peak around 30 days and another peak around 150 days (Summer).



Temporal features – Lead time

- ***Lead time vs. booking rate***

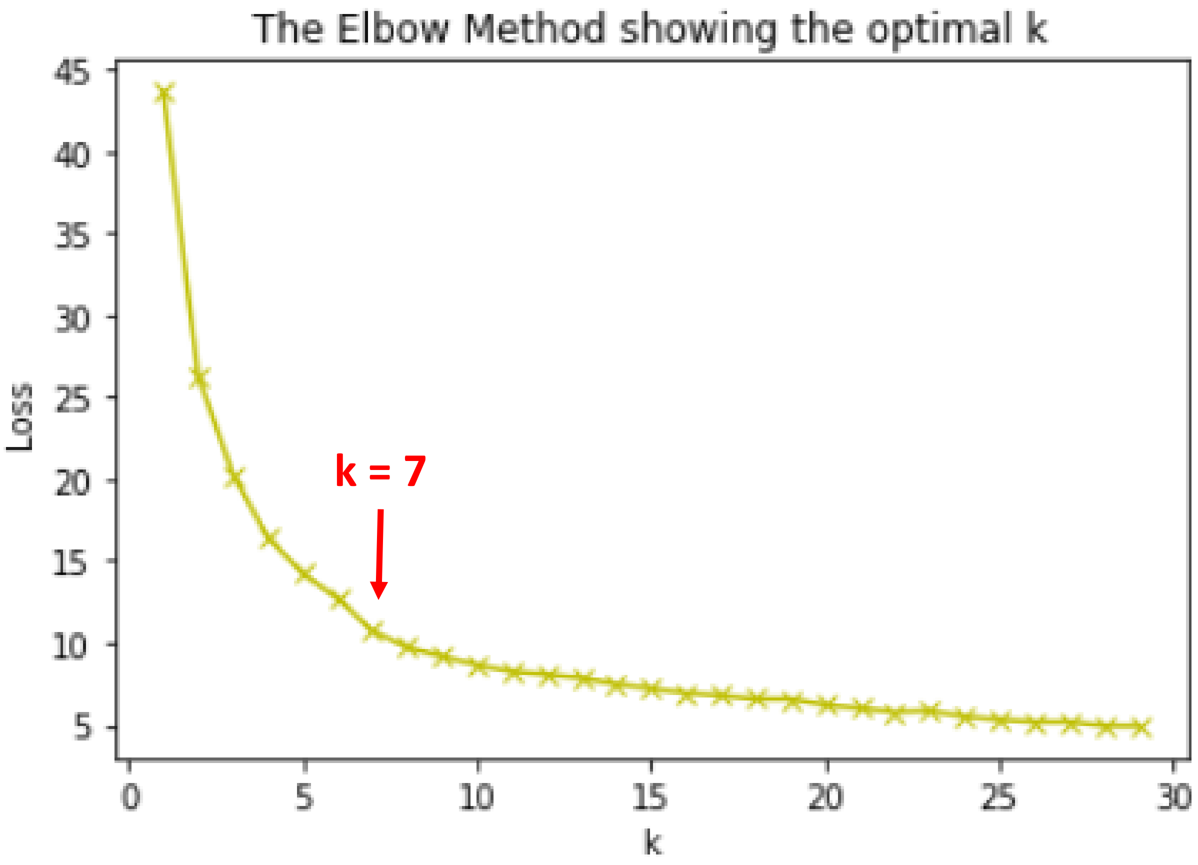
  Lead time < 50, the booking rate is higher.



Local demand feature

- **Split hotels into few clusters** by using K-means clustering.

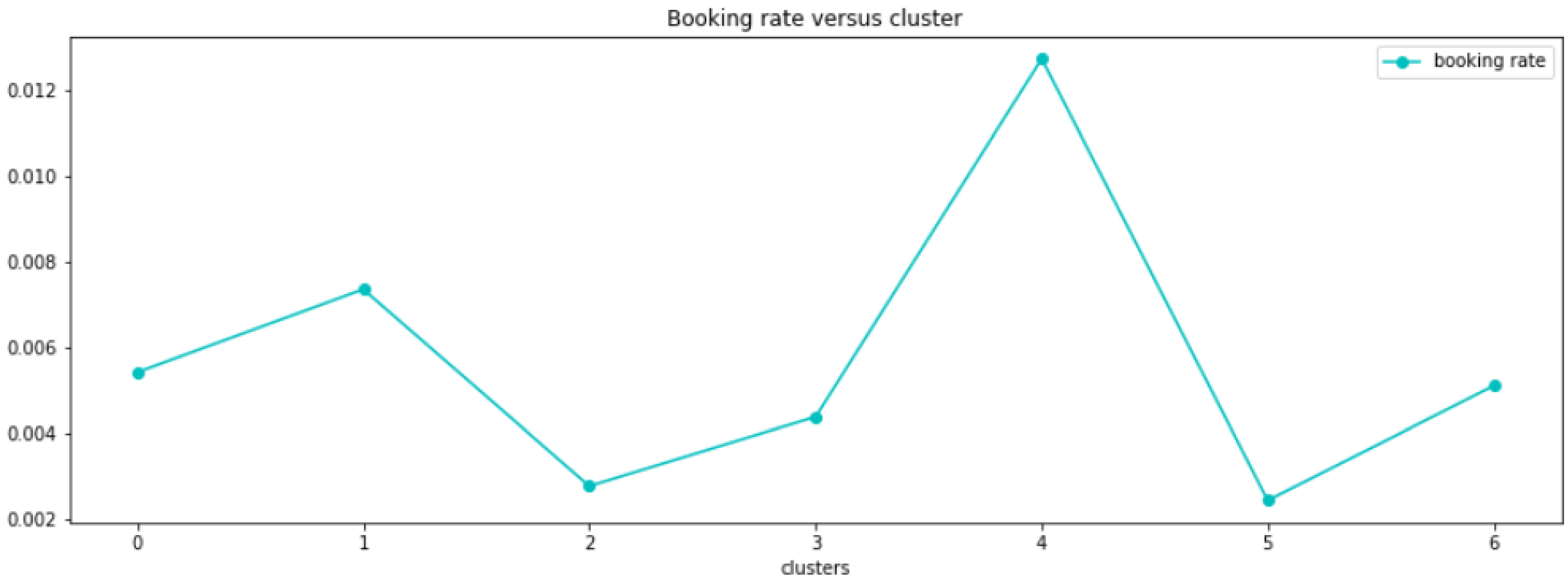
- The number of clusters is chosen according to **Elbow method.**



Local demand feature

- ***Booking rate per cluster***

  Some clusters have better booking rates, so I add another column (booking rate per cluster per year day).



Booking probability model

Under-sampling & Tran test split

The data is imbalanced, so I under-sample the majority class to get the balance. The method I used is random under-sampling.

- Original dataset shape: {0: 341919, 1: 1779}
- Resampled dataset shape: {0: 1779, 1: 1779}

I apportion the data set into training and test set with a 70% - 30% split.

- Training set Shape: (2490, 13)
- Testing set Shape: (1068, 13)

Methodology

Since there are only few thousands of samples, I will try Random Forest and Gradient Boosting Machine instead of Neural Network. I think it would be worthy to try a Neural Net when there is a data set for one year.

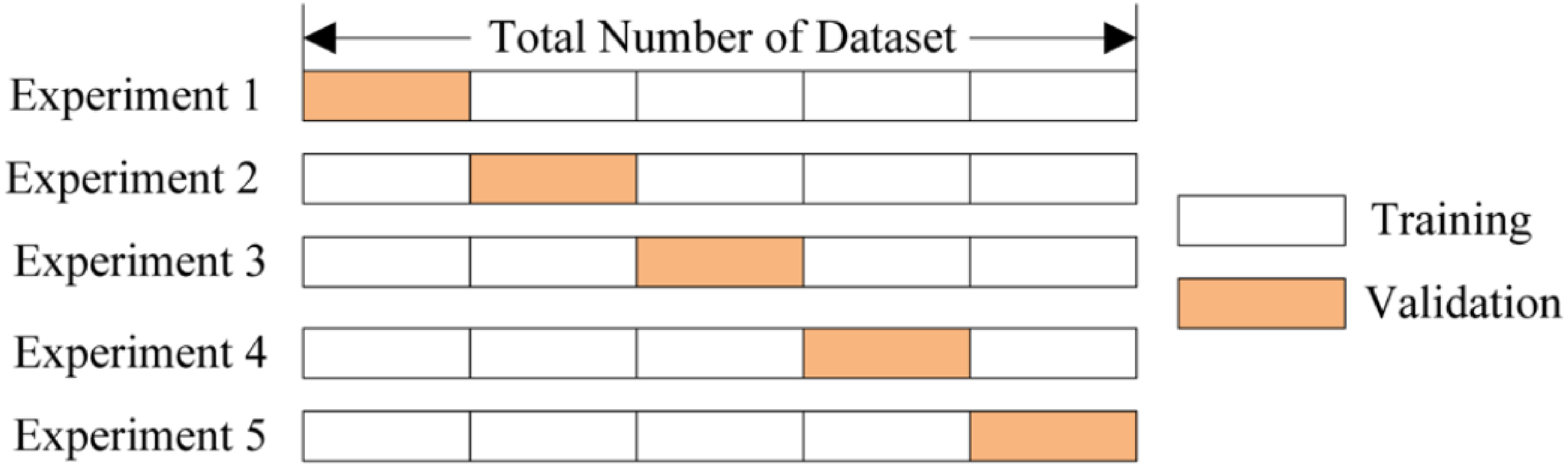
The advantages of using RF and GBM :

- No need to feature scaling.
- No need to encode categorical features.
- A good performance on the middle-sized data sets.
- Fast to train and tune.

Grid Search with 3-folds Cross Validation

Grid Search is used to search the hyperparameters for these 2 models to find the best combination. (Normally, random search should be used to narrow down the range for each hyperparameter).

Cross Validation is used to evaluate what is the best combination.



5 Fold Cross Validation (Source)

Results Comparison

- **ROC curve & AUC**

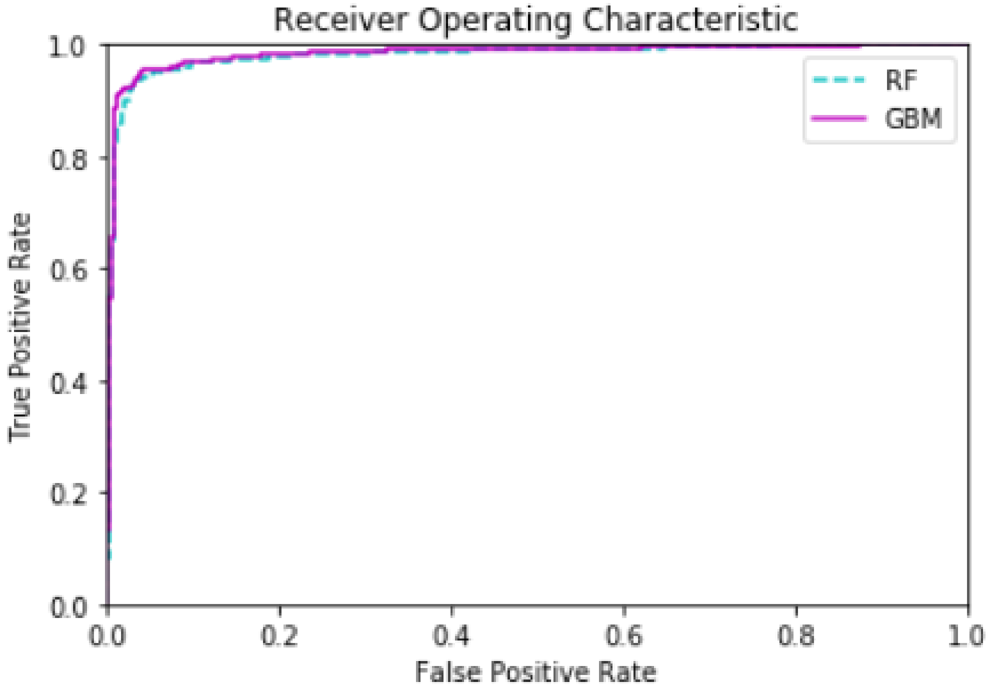
  RF AUC:0.981

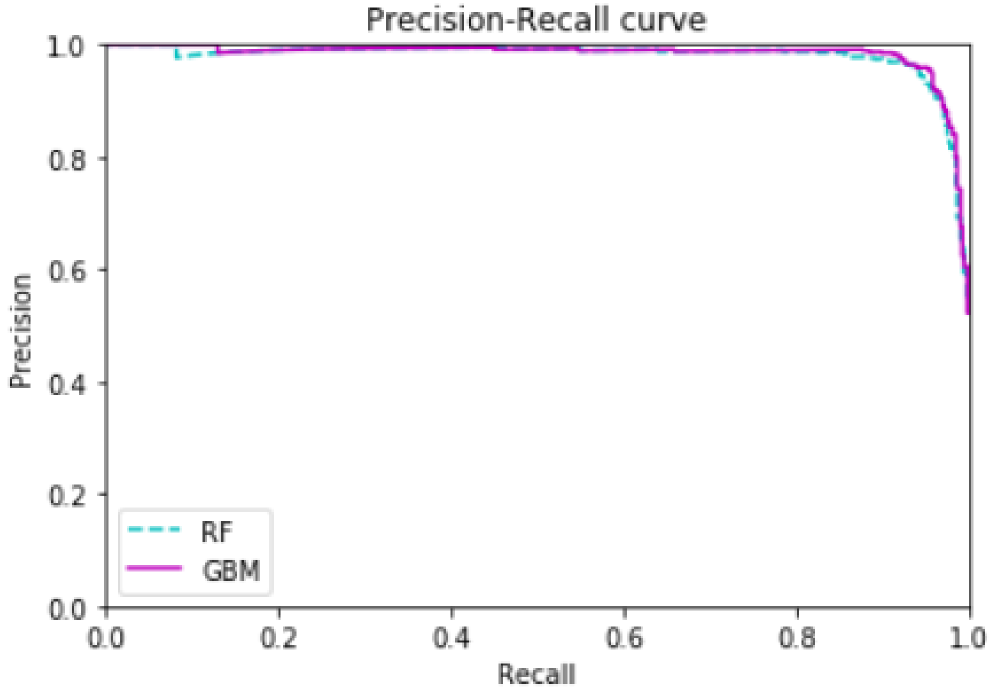
  GBM AUC:0.984

- **PR curve & F1 score**

  RF F1-score:0.933

  GBM F1-score:0.934

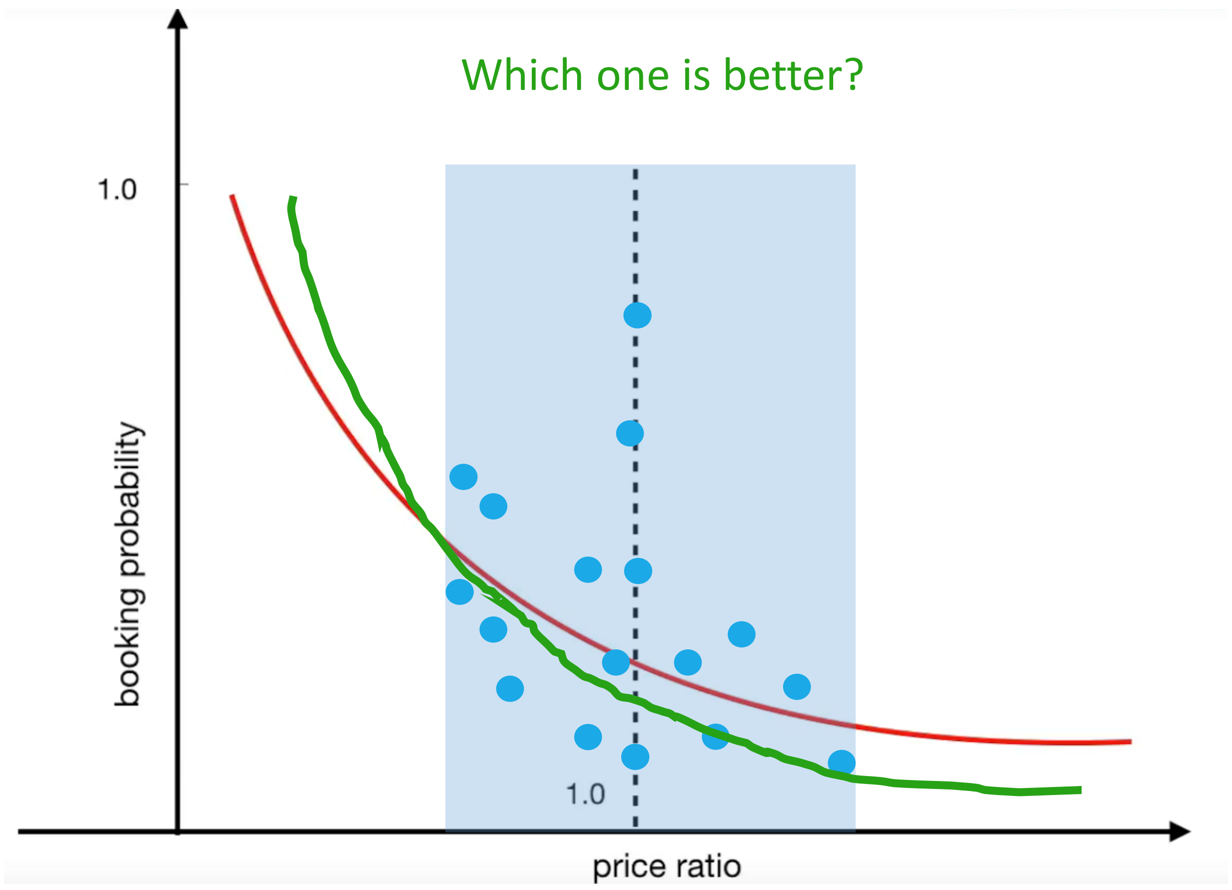




Discussion

Limitation

- It is normally hard to estimate demand curve because of data sparseness. Hotels do not change prices dramatically.
- As a result, we do not have observations of the price points that are far away from the base price (outside the blue zone).
- An alternative solution is to build a regression model to adjust the base price based on the booking probability and other factors.



Future Work

Based on the booking probability model, we can build a regression model which maps input features to price suggestions.

$$P_s = P \cdot V$$

$$V = \begin{cases} 1 + \theta_1 (q^{\phi_H^{-q^D}} - \theta_2) & if D > 0, \\ 1 + \theta_1 (q^{\phi_L^{-(1-q)^D}} - \theta_2) & if D \leq 0; \end{cases}$$

- For the same hotel, the suggested price is positively correlated with the booking probability at current price.
- Price suggestions are cantered around the most representative price that is often set by the hotel, with learnable increasing/decreasing magnitudes.
- Additional demand signals that are not fully captured by the booking probability model should be easily plugged in.

Thank you