Machine Learning for Hotel Reservation Prediction

Han Chen[†]

Project advisor: Hien Tran[‡]

Abstract. The online hotel reservation channels have caused increased cancellations, which is a revenue-diminishing factor for the hotels to deal with. Therefore, it is important to understand the bookers' behavior and make good prediction on the cancellation decision. In this work, we analyzed the characteristics of distinct sub-populations among the bookers by a hotel reservation data. We built a range of machine learning models to make predictions on whether a customer is going to cancel the reservation. We also improved our methods by selecting the important features in the data. Moreover, we investigated deep into the important features to find rational explanation on the effect. Hopefully, our work can provide suggestions on room management for hotels.

Key words. Hotel Booking, Multilayer Perceptron, Support Vector Machine, Random Forest, Boosting

1. Introduction.

The past few decades have witnessed a dramatic growth in the online hotel booking due to the popularity of booking websites such as Expedia [17], Hotels.com [16], Booking.com [20], priceline.com [2], and Orbitz [19]. On one hand, online hotel reservation channels provide flexibility and convenient access to hotel guests, on the other hand, hotel cancellations create difficulties for hotels to manage the occupancy rate and bring financial risks to the hotel industry [13]. To improve revenue as well as ensure maximum occupancy, the hotel industry employs several strategies, including the dynamic pricing policy [23, 3], also known as time-based pricing, and hotel cancellation policy [4, 8]. Dynamic pricing refers to the continual, real-time adjusting of room prices based on consumer demand, competitor pricing, seasonality, and current occupancy rate [10]. While dynamic pricing strategy has been a common practice in the travel industry, it is now gaining popularity in the hotel industry for automating revenue management [1]. Three decades ago free cancellations of reservations in hotels were common, but nowadays hotel cancellation policies offer hotel guests the opportunity to cancel their reservations until a certain amount of days before arrivals. Once this date has passed, the hotel might charge the guest a cancellation fee, which is a percentage of the booking, or, in some cases, the full amount of booking. While the hotel cancellation policy was designed to help the hotel reducing the number of no-shows, the internet has made it far quicker and easier to book a hotel, which has increased the number of hotel guests making booking “just-in-case”. The “just-in-case” guests are travellers who book more than one hotels within the same region at the same time and choose the best deal before the cancellation deadline since there is no cancellation fee until then. For hotels it not only means more cancellations, thus reducing revenue since it is unlikely to have new reservations in such a short time, but also presents more work to hotel staff. For hotel reservations with a prepayment, if the hotel booking gets cancelled the hotel staff has to undo all booking processes, including refunding the prepayment in case the deadlines were met. Therefore, it is

[†] Tanwei College, Tsinghua University, Beijing, China (han-chen20@mails.tsinghua.edu.cn).

[‡] Department of Mathematics, North Carolina State University, Raleigh, NC, USA (tran@math.ncsu.edu).

important for hotels to understand the hotel guests' booking behavior, so as to improve hotel management policies for higher profits.

The confliction between the travellers' cancellation behavior and the revenue-oriented operation of the hotels has been studied for many years, and most of the research has focused on the influence of cancellation and refund policies [8, 13], the online review and recommendation systems [7, 28], and the service quality of the hotels [22]. In spite of these many studies, it is noted that little attention has been paid to the characteristics of hotel guests themselves, including number of guests, date of arrival, preference for room type and meal, special requests, duration of stay, and how far in advance the booking is taking place. Intuitively, all these factors can influence an individual's decision of cancellation after reservation. In this work, we filled the gap by studying the influence of some of the associated characteristics of the guest on the hotel reservation cancellation. Specifically, we explored the relationship between the cancellation behavior and the personal information of travellers, such as the number of adults and children, as well as their record of previous booking and cancellation, the number of days between the date of booking, and the arrival date, etc. In particular, we are interested in the following questions:

1. How do the characteristics of the travellers and the situation of booking affect the decision of cancellation?
2. Which are the most important factors that influence the decision of cancellation?
3. Is their any efficient way to make prediction on the behavior of cancellation based on the characteristics of the travellers and the booking situation?

To fully understand these questions, we applied a range of statistical methods and machine learning models to analyze the data. Firstly, we conducted an exploratory data analysis to capture some raw features of the data. Further analysis are carried out by applying several machine learning (ML) models, e.g., multilayer perceptron (MLP), support vector machine (SVM), tree-related methods, etc. Based on the results of these ML methods, we picked several important features and refined the process of training. The results suggested that some special characteristics of travellers indeed affect the decision of cancellation. We believe our work can provide the hotel industry with a tool to develop innovative approaches in order to be as attractive as possible for travellers and maximize revenue at the same time.

The rest of the paper is organized as follows. Section 2 details the dataset and introduces the machine learning models utilized in this work; Section 3 presents the results of the machine learning models; Section 4 contains further discussions on the results; we concluded our work in Section 5 with some additional remarks.

2. Data and methods.

The hotel reservations dataset we used is from Kaggle [1]. As depicted in Table 1, in addition to the unique identifier of each booking, `Booking_ID`, there are 9 categorical variables, and 9 numerical variables in the dataset. During the process of model training, we used the variable `booking_status` as the label. Intuitively, the 17 potential influential attributes of travellers' reservation details fall into several categories: the first two variables describe the number of guests, and the following few variables depicted the various requests from the guests; some variables are related to the time of arrival and possible spending spree; some variables are associated with the previous record in bookings,

[1] https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset/data.

Table 1

Description of variables in the Hotel Reservation Data.

Variable	Description	Type*
`Booking_ID`	Unique identifier of each booking	–
`no_of_adults`	Number of adults	N
`no_of_children`	Number of children	N
`no_of_weekend_nights`	Number of weekend nights booked to stay at the hotel	N
`no_of_week_nights`	Number of weekday nights booked to stay at the hotel	N
`type_of_meal_plan`	Type of meal plan booked by the customer	C
`required_car_parking_space`	Dose the customer require a car parking space?	C
`room_type_reserved`	Type of room reserved by the customer	C
`lead_time`	Number of days between the booking date and the arrival date	N
`arrival_year`	Year of arrival date	C
`arrival_month`	Month of arrival date	C
`arrival_date`	Date of the month	C
`market_segment_type`	Market segment designation	C
`repeated_guest`	Is the customer a repeated guest?	C
`no_of_previous_cancellations`	Number of previous bookings canceled by the customer prior to the current booking	N
`no_of_previous_bookings_not_canceled`	Number of previous bookings not canceled by the customer prior to the current booking	N
`avg_price_per_room`	Average price per day of the reservation	N
`no_of_special_requests`	Total number of special requests made by the customer	N
`booking_status`	Flag indicating if the booking was canceled or not	C

*N: numerical variables. C: categorical variables.

while others focusing on the average price of the room, which may indicates the quality of hotels and the levels of travellers' spending.

The marginal distribution of 18 variables of interest are shown in Figure 1. We also looked into the distributions of the 15 potential factors with respect to the label, `booking_status`, in Figure 2. In Table 2 we listed some basic information, e.g. quantiles, of the numerical variables.

Table 2

Details of some numerical variables. Starting from the second column: mean, standard deviation, minimum, the first quartile, median, the third quartile, maximum.

Variable	Mean	Std	Min	25%	50%	75%	Max
`no_of_adults`	1.85	0.52	0	2	2	2	4
`no_of_children`	0.11	0.40	0	0	0	0	10
`no_of_weekend_nights`	0.81	0.87	0	0	1	2	7
`no_of_week_nights`	2.20	1.41	0	1	2	3	17
`lead_time`	85.23	85.93	0	17	57	126	443
`no_of_previous_cancellations`	0.02	0.37	0	0	0	0	13
`no_of_previous_bookings_not_canceled`	0.15	1.75	0	0	0	0	58
`avg_price_per_room`	103.42	35.09	0	80.30	99.45	120	540
`no_of_special_requests`	0.62	0.79	0	0	0	1	5

From Figures 1 and 2 we can find some interesting facts about the behaviors of customers. We are surprised to see that the amount of cancellation is proportional to that of not-canceled

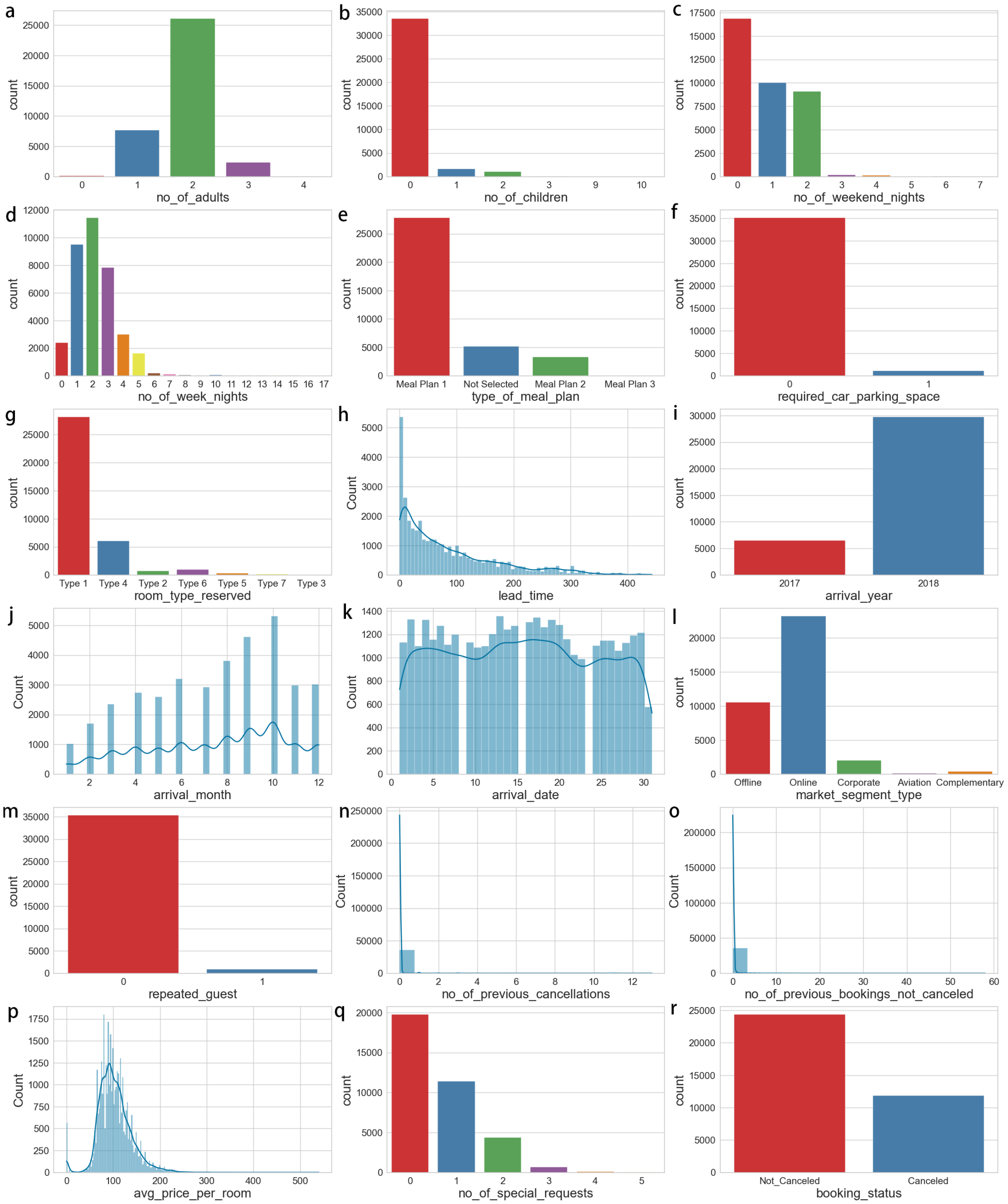

Figure 1. *Marginal distribution of the 18 variables of interest. The x-axis of each subplot is the value of variable, with 0 for No and 1 for Yes for binary variables; the y-axis are number of observations with the corresponding values.*

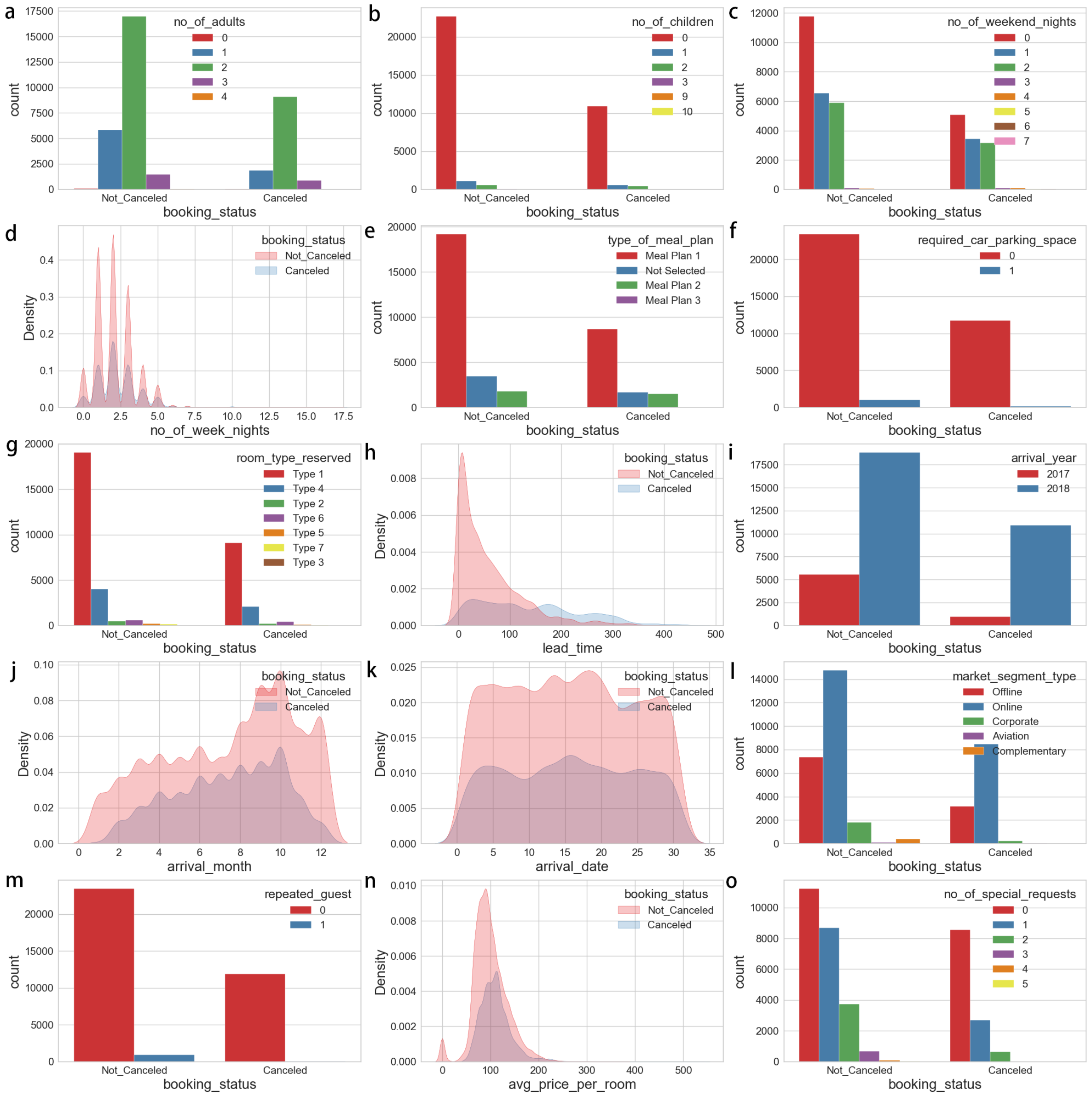

Figure 2. *Marginal distribution of 15 variables with respect to the booking status. The classified distribution of two booking record, i.e.,* `no_of_previous_cancellations` *and* `no_of_previous_bookings_not_canceled` *are not demonstrated. The distribution of these two variables are relatively concentrated with long right tails.*

(Figure 1r), which emphasizes the importance of accurately predicting the cancellation behavior of customers since the hotel reservation cancellation is far from a rare phenomenon. Besides, it seems that most adults tend to go traveling in pairs taking no children (Figure 1b). The duration of the journey is usually shorter than a week, since the number of weekday nights is mostly less than 5 (Figure 1d) and the number of weekend nights is mostly less than

2 (Figure 1c). Many people have preferences to one specific type of room (Figure 1g), perhaps the twin room, which is reasonable considering the number of the bookers. As for the time of arrival, the number of reservations arrived at a peak in October (Figure 1j), and there is no significant trend in the date of the arrival in a specific month (Figure 1k).

Figure 2 suggests that the marginal distribution of potential influential variables are similar in both canceled and not-canceled groups, indicating that the decision of cancellation may be a combined results rather than depending on a single factor. Therefore, ML methods are required for an accurate prediction of the hotel reservation cancellation.

Before we dive into the details of the main ML models implemented in this work, we introduced some common notation. Assume that we have N observations, i.e., a total of N reservations in the dataset, let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^p$ denoted the p-dimensional vector of features or attributes of the hotel guests, and $y_1, \dots, y_N \in \{\pm 1\}$ the labels `booking_status`. For $t \in \{1, \dots, N\}$, let $\boldsymbol{x}_t = (x_{1t}, \dots, x_{p,t})^\top$. For notation simplicity, we omit the index t when there is no confusion. For $t \in \{1, \dots, N\}$, let $\hat{y}_t \in \{\pm 1\}$ be the predicted value of y_t by the machine learning models.

2.1. MLP. In this work, we applied an MLP with two hidden layers [25, 21]. For $k = 1, 2$, let n_k be the number of neurons in the k-th hidden layer. For $i = 1, \dots, n_k$, let $h_i^{(1)}$ and $h_i^{(2)}$ be the output of the i-th neuron in first and second hidden layer, respectively, and $w_{ij}^{(k)}$ be the weights of x_j $(j = 1 \dots, p)$ in the i-th neuron of the k-th hidden layer, and $b_i^{(k)}$ be the bias in the i-th neuron of the k-th hidden layer, with $i \in \{1, \dots, n_k\}$, and $k \in \{1, 2\}$. Let $\boldsymbol{w}_i^{(k)} = \big(w_{i1}^{(k)}, \dots, w_{ip}^{(k)}\big)^\top$. It is well-known that the neuron-like processing unit is

$$h_i^{(k)} = \varphi\left(\sum_{j=1}^{p} w_{ij}^{(k)} x_j + b_i^{(k)}\right) = \varphi\big(\boldsymbol{x}^\top \boldsymbol{w}_i^{(k)} + b_i^{(k)}\big), \tag{2.1}$$

where φ the activation function. In the analysis we fixed the size to $n_1 = 25$ and $n_2 = 5$ for the two hidden layers, respectively, and tried 4 widely-used activation functions - `identity, logistic, tanh, relu`. Upon choosing the activation function, we obtain $w_{ij}^{(k)}$ by initiating with a random choice of $w_{ij}^{(k)}$ for all $i \in \{1, \dots, n_k\}$, $j \in \{1, \dots, p\}$, and $k \in \{1, 2\}$, and each time solving an optimization problem

$$\boldsymbol{w}_i^{(k)} = \arg\min_{w \in \mathbb{R}^p} J, \quad \text{where } J := \sum_{t=1}^{N} (\hat{y}_t - y_t)^2. \tag{2.2}$$

The simplest way to solve the optimization problem, that is, to update or to learn the weights is by using the steepest descent as follows

$$\boldsymbol{w}_i^{(k)} := \boldsymbol{w}_i^{(k)} - \eta \nabla_{\boldsymbol{w}_i^{(k)}} J, \tag{2.3}$$

for some positive learning rate η. Since the MLP could get stuck in a local minimum, we repeated the random choice of the initiate values of w_{ij}^{k}'s for multiple times and chose the best weights in the end. We summarize the whole process in Algorithm 2.1. In practice, the

Algorithm 2.1 MLP algorithm with two hidden layers.

```
Define the set of activation functions
$\mathcal{F} = \{\texttt{identity, logistic, tanh, relu}\}$.
Define the time of repetition $R$.
Define the maximal iteration $T$.
Define the threshold of convergence $\varepsilon$.
Define the learning rate $\eta \in (0, 1)$.
for $\varphi$ in $\mathcal{F}$ do
  for $r$ in $1 \to R$ do
    Start with a random guess on $w_{ij}^{(k)}(0)$, for $k \in \{1, 2\}$, $j \in \{1, \dots, p\}$, $i \in \{1, \dots, n_k\}$.
    for $\tau$ in $1 \to T$ do
      $e = 0$.
      for $k$ in $2 \to 1$, step by $-1$ do
        for $i$ in $1 \to n_k$ do
          Compute $J$ using (2.2) with $w_{ij}^{(k)}(\tau - 1)$ for $k \in \{1, 2\}$, $j \in \{1, \dots, p\}$, $i \in \{1, \dots, n_k\}$.
          for  $s$ in $1 \to T$ do
            Update $\boldsymbol{w}_i^{(k)}(\tau - 1)$ by (2.3);
            Update $J$ by (2.2).
          end for
          $\boldsymbol{w}_i^{(k)}(\tau) = \boldsymbol{w}_i^{(k)}(\tau - 1) - \eta \nabla_{\boldsymbol{w}_i^{(k)}(\tau-1)} J$.
          $e = e + \|\boldsymbol{w}_i^{(k)}(\tau) - \boldsymbol{w}_i^{(k)}(\tau - 1)\|_2$.
        end for
      end for
    end for
  end for
  if $e < \varepsilon$ then
    $w_{ij}^{(k)}(T) = w_{ij}^{(k)}(\tau)$, for $k \in \{1, 2\}$, $j \in \{1, \dots, p\}$, $i \in \{1, \dots, n_k\}$.
    break
  end if
end for
return  $w_{ij}^{(k)}(T)$, $k \in \{1, 2\}$, $j \in \{1, \dots, p\}$, $i \in \{1, \dots, n_k\}$ for each $\varphi \in \mathcal{F}$.
```

learning rate η is set to be 0.02, the time of repetition $R = 500$, and the maximal number of iterations $T = 500$, with the threshold $\varepsilon = 10^{-10}$. The choice of activation function φ are determined by the performance on the validation set later using 5-fold cross validation.

2.2. SVM. Using SVM we try to find the best hyperplane that separates the observations with distinct labels as much as possible [27]. The procedure is equivalent to finding a weighting vector $\boldsymbol{w} \in \mathbb{R}^p$ and the bias b, which are the solutions of the constrained optimization problem

$$\min_{b \in \mathbb{R},\ \boldsymbol{w} \in \mathbb{R}^p} \|\boldsymbol{w}\|, \quad \text{s.t.} \quad y_t(\boldsymbol{w}^\top \boldsymbol{x}_t + b) \geq 1,\ t = 1, \dots, N. \tag{2.4}$$

Here $\|\cdot\|$ is the Euclidean norm. SVM also indicates the relative importance of features in the training data. Consider the coordinates of $\boldsymbol{w} = (w_1, \dots, w_p)^\top$. For $j \in \{1, \dots, p\}$, the larger the absolute value $|w_j|$, the more important the corresponding variable x_j would be. Another perspective is that the algorithm places more weight on the influential variables. Therefore, we can only select the top several important features for training, which is likely to facilitate the process. However, the boundary between two subsets may be highly nonlinear, and the computation complexity increase dramatically if we add more higher order terms in the optimization problem. Therefore, we considered the soft-margin problem with the radial basis function (RBF) kernel [9, 14, 26] in training. In this case, the problem can be expressed by the dual formulation

$$\max_{\boldsymbol{\alpha}} D_\gamma(\boldsymbol{\alpha}) = \sum_{t=1}^{N} \alpha_t - \frac{1}{2} \sum_{t=1}^{N} \sum_{s=1}^{N} \alpha_t \alpha_s y_t y_s k_\gamma(\boldsymbol{x}_t, \boldsymbol{x}_s) \tag{2.5}$$
$$\text{s.t.} \quad \begin{cases} 0 \le \alpha_t \le C, \ t = 1, \dots, N, \\ \sum_{t=1}^{N} y_t \alpha_t = 0, \end{cases}$$

where k_γ denotes the RBF kernel

$$k_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p, \tag{2.6}$$

and $C \ge 0$ is a hyperparameter to be tuned. Intuitively, C indicates the scale of margin of separation. A small C tend to make the algorithm to search for a hyperplane with a large margin of separation, while a large C will lead to a small margin of separation, which may lower the proportion of misclassification. After determining the α_t^*'s by solving the optimization problem, we derive b^*, the estimation of the interception b, by minimizing the classification error $\sum_{t=1}^{N} |\hat{y}_t - y_t|^2$. Then we make prediction by the formula

$$\hat{y}(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{N} \alpha_t^* y_t k_\gamma(\boldsymbol{x}, \boldsymbol{x}_t) + b^* \right) \tag{2.7}$$

for given $\boldsymbol{x}$ in the test set. In the training, we search in grid for C and γ, and selected the best combination via cross validation.

2.3. Tree-based methods.

To explore the usage of non-parametric models in hotel reservation prediction, we applied decision tree (DT), random forest (RF) and other advanced tree-based methods.

For DT, we used two algorithms, classification and regression tree (CART) [6]. and iterative dichotomiser 3 (ID3) [24]. The algorithms start with all of the training data, then consider the j-th ($j \in \{1, \dots, p\}$) splitting variable, finding the split point $\zeta \in \mathbb{R}$, which defines the pair of half-planes

$$R_1(j, \zeta) = \{\boldsymbol{x} | x_j \le \zeta\} \text{ and } R_2(j, \zeta) = \{\boldsymbol{x} | x_j > \zeta\}. \tag{2.8}$$

Basing on the impurity criteria, we seeked the splitting variable and split point that solve

$$\min_{j,\zeta}\left\{\min_{c_1\in\mathbb{R}}\sum_{\boldsymbol{x}_t\in R_1(j,\zeta)}(y_t-c_1)^2+\min_{c_2\in\mathbb{R}}\sum_{\boldsymbol{x}_t\in R_2(j,\zeta)}(y_t-c_2)^2\right\}, \tag{2.9}$$

where y_t are the response corresonding to x_t (e.g. whether or not a customer cancelled his/her reservation in our scenario).

We used Gini impurity and the entropy as impurity criteria. For a specified node, denote by p_k the probability of class k falling in the region R_1, where $p = 1, 2$. The Gini impurity has the form $\sum_{k=1}^{2} p_k(1-p_k)$, which measures how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset; the entropy has the form $-\sum_{k=1}^{2} p_k \log p_k$, which measures the information gain of the node. Therefore, Gini impurity and entropy encourage pure nodes.

Having found the best split, we then partitioned the data into the two resulting regions and repeated the splitting process on each of the regions, until we meet a stopping criterion, which we set to be a minimum node size of 5. In this way we grew the tree $\mathfrak{T}_0$.

To avoid overfitting, we then pruned the decision tree to a smaller one with fewer splits. We define a subtree $\mathfrak{T} \subset \mathfrak{T}_0$ to be any tree that can be obtained by pruning $\mathfrak{T}_0$, i.e., collapsing any number of its internal nodes. We denote by $|\mathfrak{T}|$ the number of terminal nodes in $\mathfrak{T}$. Given $\alpha > 0$, our goal is to find the subtree $\mathfrak{T}_\alpha \subset \mathfrak{T}_0$ to minimize the cost complexity

$$C_\alpha(\mathfrak{T}) = \sum_{m=1}^{|\mathfrak{T}|} n_m \cdot g(R_m) + \alpha|\mathfrak{T}|, \tag{2.10}$$

where n_m represents the number of units in the m-th terminal, and $g(\cdot)$ is the measure of impurity, i.e., Gini impurity or entropy. Note that the $\mathfrak{T}_\alpha$ can be obtained by successively collapse the internal node that produces the smallest per-node increase in the residual sum of squares (RSS) until we produce the single-node tree. Then $\mathfrak{T}_\alpha$ must be contained in the sequence. Moreover, we can determine the optimal α via cross validation.

We summarize the whole procedure in Algorithm 2.2.

A random forest fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting [5]. When building the decision trees, each time a split in a tree is considered, we made a random selection of q predictors as split candidates from the full set of p variables, where $q \approx \sqrt{p}$.

We also applied boosting methods since in general they performed better than previous ones [12]. We summarize the procedures of AdaBoost [11] and gradient boosting [18] that we applied in Algorithm 2.3 and 2.4, respectively. The boosting algorithms primarily depend on the construction of decision trees, whereas they include various penalty and heterogeneous weights for a better result.

2.4. Naïve Bayes (NB) classifier. We also explored the performance of NB [15] in the hotel reservation cancellation prediction. We assume that units fall into two classes with equal probability. Since the obervations in the class of not-canceled booking are significantly more than those in the canceled group (Figure 1), in practice, we sampled the observations labeling with ‘not-canceled’ so that the number of observations in the two classes are the same.

Algorithm 2.2 Fitting a decision tree

```
Initiate $\mathfrak{T}_0 = \emptyset$.
Define $D \subset \mathbb{R}$ the set of possible values of $\alpha$.
Define the set of node impurity measures $\mathcal{G} = \{\texttt{Gini, entropy}\}$.
while node size $\geq 5$ do
  Find $(j, \zeta)$ that solve $\min_{j,\zeta}\Big\{\min_{c_1} \sum_{\boldsymbol{x}_t \in R_1(j,\zeta)} (y_t - c_1)^2 + \min_{c_2} \sum_{\boldsymbol{x}_t \in R_2(j,\zeta)} (y_i - c_2)^2\Big\}$.
  Update $\mathfrak{T}_0 := \mathfrak{T}_0 \cup \{\text{node}(x_j, \zeta)\}$.
end while
for $g$ in $\mathcal{G}$ do
  for $\alpha$ in $D$ do
    Find $\mathfrak{T}_\alpha \subset \mathfrak{T}_0$ that minimizes the cost complexity criterion $C_\alpha(\mathfrak{T})$.
    Compute the errors corresponding to $\alpha$ by 5-fold cross validation.
  end for
  Choose the optimal $\alpha$ according to the results of cross validation.
end for
Choose the optimal $g$ according to the results of cross validation.
return $\mathfrak{T}_\alpha$ that corresponds to the optimal $g$ and $\alpha$.
```

Algorithm 2.3 AdaBoost

```
Initialize the weight of each unit $w_t = 1/N$, for $t = 1, \ldots, N$.
for $m$ in $1 \to M$ do
  Build a decision tree $\mathfrak{T}_m$ for the training data using weights $w_t$.
  Compute the error rate
      $$r_m = \frac{\sum_{t=1}^{N} w_t \mathbb{1}\{y_t \neq \mathfrak{T}_m(\boldsymbol{x}_t)\}}{\sum_{t=1}^{N} w_t}.$$
  Compute $\nu_m = \log\big((1 - r_m)/r_m\big)$.
  Update $w_t := w_t \cdot \exp\big\{\nu_m \cdot \mathbb{1}\{y_t \neq \mathfrak{T}_m(\boldsymbol{x}_t)\}\big\}$ for $t = 1, \ldots, N$.
end for
return $G(\boldsymbol{x}) = \text{sign}\big\{\sum_{m=1}^{M} \nu_m \mathfrak{T}_m(\boldsymbol{x})\big\}$.
```

Our classification are based on the strong Gaussian assumption, i.e., units in each class follows a multivariate Guassian distribution, $\boldsymbol{x}|_{y \in C_k} \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$ for $k = 1, 2$, where C_k is the class indicator. Denote by π_k the probability of units falling in the k-th class [2]. Our goal is to maximize the total probability

$$\pi_k \cdot \prod_{i=1}^{p} \mathbb{P}(x_i | C_k). \tag{2.11}$$

Here we further assumed the coordinate-wise independence among variables for simplicity,

[2] As previously discussed, we only take $\pi_1 = \pi_2 = 0.5$, but here we give a general expression.

Algorithm 2.4 Gradient boosting

```
Initialize $f_0(\boldsymbol{x}) = \arg\min_\gamma \sum_{t=1}^{N} L(y_t, \gamma)$, where $L(y, \gamma)$ is the exponential loss.
for $m$ in $1 \to M$ do
  for $t$ in $1 \to N$ do
    Compute
      $$r_{tm} = -\left\{ \frac{\partial L\big(y_t, f(\boldsymbol{x}_t)\big)}{\partial f(\boldsymbol{x}_t)} \right\}_{f=f_{m-1}}$$
  end for
  Fit a regression tree to the targets $r_{tm}$ giving terminal regions $R_{im}$, $i = 1, \ldots, n_m$.
  for $j$ in $1 \to n_m$ do
    Compute $\gamma_{im} = \arg\min_\gamma \sum_{\boldsymbol{x}_t \in R_{im}} L(y_t, f_{m-1}(\boldsymbol{x}_t) + \gamma)$.
  end for
  Update $f_m(\boldsymbol{x}) = f_{m-1}(\boldsymbol{x}) + \sum_{i=1}^{n_m} \gamma_{im} \mathbb{1}\{\boldsymbol{x} \in R_{im}\}$.
end for
return $\hat{f}(\boldsymbol{x}) = f_M(\boldsymbol{x})$.
```

which, however, is a strong assumption. But it turns out that the correlation between variables are generally low (Section 3, Figure 3). Therefore, we used this assumption to build the model.

At the end of this section, we want to note that in training, all of the methods are applied after data standardization, with categorical variables transformed to dummy variables. In this way, we brought the effects of scaling to a negligible degree.

3. Results.

First, we compute the correlation of variables after data preprocessing (Figure 3). We notice that the correlation is not very significant, with the highest being 0.54, the correlation between `repeated_guest` and `no_of_previous_cancellations`. Based on the results, we include all the features in the model at the beginning.

The trained and tuned results are shown in Table 3. We fitted 9 different models, some belonging to the parametric methods and others nonparametric (the tree-based methods).

Table 3

Accuracy of models on the test set. MLP, multilayer perceptron; SVM, support vector machine; DT, decision tree; RF, random forest; ET, extra trees; AdaBoost, decision tree with AdaBoost; Gradient, decision tree with gradient boosting; NB, Naïve Bayes; Logistic, logistic regression. The first line (type) indicates specific use of certain algorithms in the training.

	MLP	**SVM**	**DT**	**RF**	**ET**	**AdaBoost**	**Gradient**	**NB**	**Logistic**
Type	`relu`	`RBF`	`Gini`	`Gini`	`Gini`	–	–	Gaussian	–
Accuracy	.88	.83	.87	.87	.93	.81	.88	.54	.78

In the case of MLP, we found that the activate function `relu` performed better than other choices, with a accuracy of 0.88 on the test set. Though we used RBF kernel in the SVM, the model performed not well, for which we thought may due to the curse of dimensionality. The tree-based methods performed well in general. Particularly, ET has the bset performance

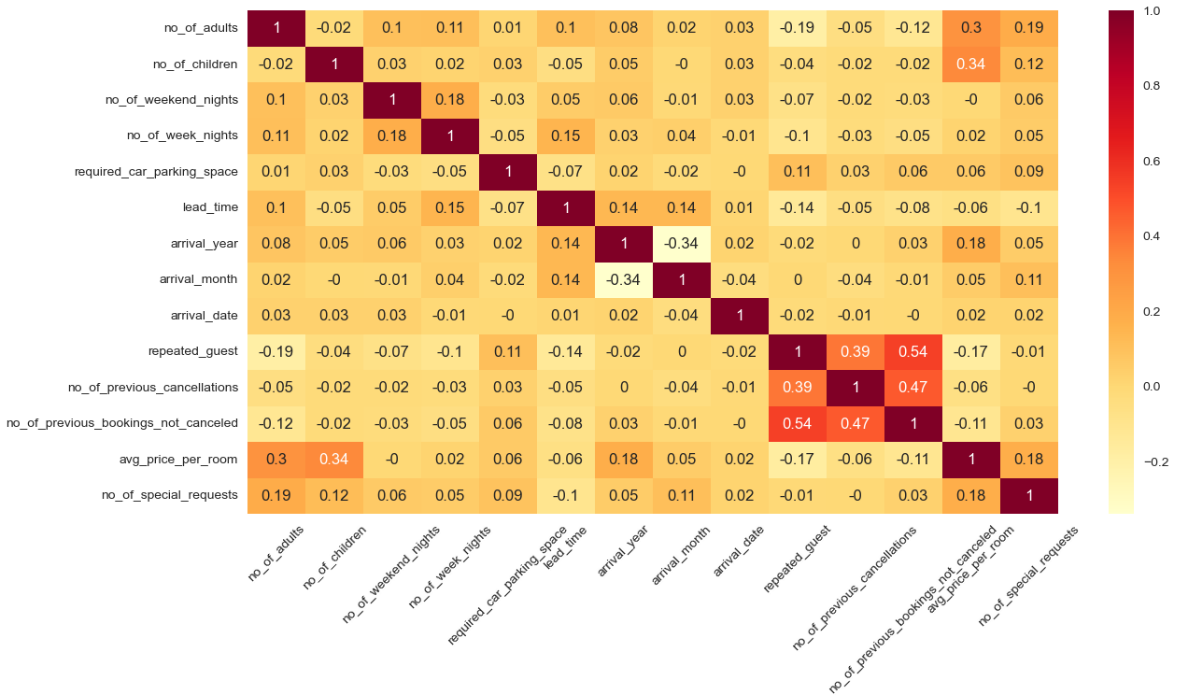

Figure 3. *Correlation heat map*

among all the nine methods, with a accuracy reached 0.93. We attributed the good results of tree-based methods to the fact that they are more likely to mirror human decision-making and can handle highly nonlinear patterns. The performance of NB is poor, partially due to the strong Guassian and independence assumption of the method. We also ran a logistic regression, but the result were not comparable with the previous methods.

We notice that there is still room for improvement in our models, for the dimension of variables are high. We think that removing some of the nuisance features might help in improving our training efficiency and the overall model performance. Moreover, by figuring out the importance of all the variables and studied the effects (positive or negative) they have on the decision of cancellation, we can have a clearer picture of the customers' behavior, which may be helpful in improving the management policy for the hotels.

3.1. Feature selection and model optimization.

To figure out the most important features influencing the cancellation behavior, we applied a range of methods, e.g., DT, RF, extra trees, and the chi-square statistics of each feature, to get the relative importance of variables. The chi-square test measures dependence between variables, the lower the chi-square score, the more likely that the features are independent of class and therefore irrelevant for classification. The results are listed in Table 4.

We found that the results of the four approaches bare some similarity. Among all the features, `lead_time` has the dominantly highest score. We found that variables like the number

Table 4

Relative importance of features under different methods (round up to 2 decimal places). For categorical variable, we summed up the relative important scores of all its dummy variables.

Feature	Decision Tree	Random Forest	Extra	χ^2
`no_of_adults`	.02	.02	.03	62.61
`no_of_children`	–	–	.01	97.71
`no_of_weekend_nights`	.03	.02	.05	183.33
`no_of_week_nights`	.03	.03	.07	441.53
`type_of_meal_plan`	.01	.01	.01	374.34
`required_car_parking_space`	.01	.01	.01	485.70
`room_type_reserved`	–	–	–	103.91
`lead_time`	.41	.35	.27	797722.63
`arrival_year`	.02	.05	.04	0.13
`arrival_month`	.06	.07	.10	14.25
`arrival_date`	.03	–	.10	22.78
`market_segment_type`	.15	.09	.03	1486.76
`repeated_guest`	–	.01	.01	819.60
`no_of_previous_cancellations`	–	–	–	418.67
`no_of_previous_bookings_not_canceled`	–	.01	–	5390.79
`avg_price_per_room`	.10	.10	.12	12869.91
`no_of_special_requests`	.13	.17	.13	4088.34

of adults, the number of weekday and weekend nights, the requirement for car parking space, the average room price and the number of special requests all have significant influence on the reservation cancellation. We conjectured that more lead time places more uncertainty on the schedule, thus more likely the bookers will change their mind to cancel the reservation. Besides, it is more convenient for a booker to change his/her reservation for another hotel when the duration of stay is relatively short, which contributes to the influence of number of weekday and weekend nights. We leave further discussion with concrete statistical analysis to the next subsection. On the contrary, the number of children and the type of room reserved merely effect the behavior of the bookers, and we found with surprise that little did the previous record provide reference for the present booking. These factors seem to play a little role in the prediction. Therefore, we consider dropping them to improve the model performance.

According to the scores of significance, we selected the top 10 features following the significance order, with which we retrain our models on the training set. The results are shown in Table 5.

Table 5

Comparison of accuracy on the test set between models with/without feature selection. The first line shows the results of models that all features were included in training; the second line shows the results of models that were trained with those selected features: `lead_time`, `no_of_special_requests`, `avg_price_per_room`, `arrival_date`, `arrival_month`, `arrival_year`, `no_of_week_nights`, `no_of_weekend_nights`, `no_of_adults`, `market_segment_type`.

	MLP	SVM	DT	RF	ET	AdaBoost	GradientBoost	NB	Logistic
Raw	.88	.83	.87	.87	.92	.81	.88	.54	.78
Selected	.86	.91	.87	.88	.93	.82	.88	.75	.77

The SVM and NB demonstrated significant improvement after feature selection. We think the improved accuracy is mainly due to a reduction in dimension to lower the noise in training. The performance of other methods are similar before and after feature selection, indicating that removing some nuisance features does no harm to the accuracy of our prediction. Moreover, the time of training decreased significantly. In the case of SVM, the training time shrank from 388 min to 347 min, which was about 10.6% faster than the previous one.

3.2. Feature effects. Besides the improved performance and efficiency in hotel reservation prediction, we also performed some additional analysis of feature selection. Recall that in Table 4 we showed that the variable `lead_time` had the highest significant score in all the approaches we adopted, we are interested in the actual effect of this variable, i.e., how the number of days between the booking date and the arrival date affect the bookers' decision and the interaction between this feature and other variables.

In Figure 4, we investigated the relation between `lead_time` and other variables, including the label `booking_status` and some categorical variables. We found that the proportion of lead time of the canceled group and the not-canceled group are significantly different - reservations made within one month are less likely to be canceled, while those who made reservations more than three months earlier often tend to cancel it. Therefore, early reservations usually contains more uncertainty, which is consistent with our commonsense. Moreover, we detected difference in the booking date between different market segment designation. Compared to the online customers, the customers offline tend to make reservations earlier, which may due to a lack of flexibility and convenience for offline booking.

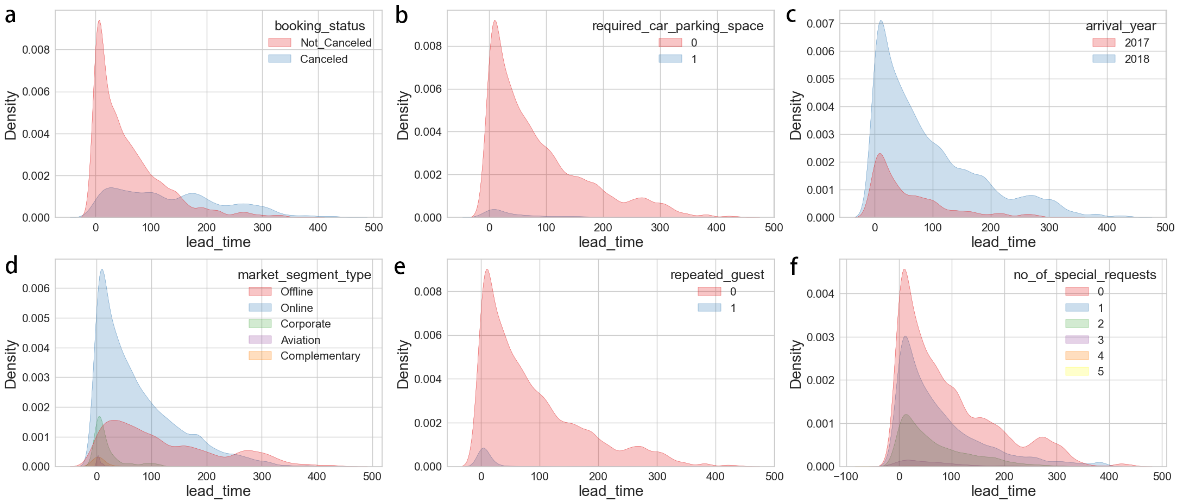

Figure 4. *Changes of some variables with respect to* `lead_time`.

We also looked into the extent to which the important features differed in the canceled and not-canceled group. We employed the simple one-way ANOVA to check the significance of difference in two classes, and the results are shown in Table 6 . It turned out that our chosen features indeed have significant difference between two groups, which makes the feature selection step more rational. Specifically, we found that those canceled reservations tend to have longer time between the day of booking and the day of arrival; their special requests are relatively less, and the average price per room during the gap days are higher. Moreover, longer staying tend to be associated with lower probability of cancellation, and reservations with fewer people (adults) are more likely to be canceled.

Table 6
One-way ANOVA in feature importance

Variable	F-value	P-value
`lead_time`	8637	$< 2e-16$ ***
`no_of_special_requests`	2482	$< 2e-16$ ***
`avg_price_per_room`	752.6	$< 2e-16$ ***
`arrival_date`	4.098	0.0429 *
`arrival_month`	4.578	0.0324 *
`arrival_year`	1208	$< 2e-16$ ***
`no_of_week_nights`	316.4	$< 2e-16$ ***
`no_of_weekend_nights`	138	$< 2e-16$ ***
`no_of_adults`	276.1	$< 2e-16$ ***

4. Discussion. According to the results in the previous section, now we are able to answer the questions we listed in Section 1. We analyzed the hotel reservation data and found some interesting phenomenon in the population of bookers.

- The majority of the customers prefer traveling in pairs without any child.
- The demand for twin rooms are dramatically higher than any other type of rooms.
- The majority of travellers schedule their journey to be within one week.
- In general, online users make reservation later than the offline customers, which is due to the convenience and timeliness of the online channels.

In addition, we made prediction in hotel reservation via ML classification models, which also reveals the important factors that influence a cancellation decision. The results suggested that the number of days between the date of booking and the arrival date is the most significant factor influencing the cancellation decision. Besides, the number of special requests, the average price of the room, the date of arrival, the length of staying, and the number of adults all have effects on the bookers' behavior. Note that the combination of factors included both the characteristics of the customer and the hotel. These results can be explained with the following reasonings:

- The earlier bookers are more likely to cancel the reservation, since a longer period of reserving time leads to more uncertainty. On the other hand, bookers who make reservations near the date of arrival are more sure of their schedule, thus not likely to change their plan due to the time confliction.
- Reservations with special requests are not likely to be canceled, comparing to those without special requests. We think that in general, bookers with special requests consider more thoroughly than those do not. Therefore, the former is less likely to change their mind. Additionally, those low-price seekers making reservations in a range of hotels at the same time tend not to make special requests, since they lay more emphasis on the financial issue. This also partially accounts for the higher probability of cancellation for those without special requests.

- Reservations with a higher average price of room are more likely to be canceled than those with lower price. We attribute the difference to the fact that the dynamic pricing practice encourages people to make more reservations than needed and finally choose the one with the lowest price.
- Reservations with a longer period of staying are less likely to be canceled. We conjecture that this is because people booking for a long time are more sure about their schedules. Moreover, they are less likely to be the lower-price seekers, since it takes a lot of time to booking for a long stay in a range of hotels and cancel most of them in a relatively short period of time.
- We found it interesting that reservations with more adults are less likely to be canceled, while the number of children seems to make no difference. We attribute this phenomenon to the fact that a travel with children is usually for fun and relax, therefore not so necessary and more likely to be canceled by other issues that are more important.

From the discussion above, we noticed that the interaction between online booking and dynamic pricing practice may have a significant influence on the cancellation decision. We leave it to the future work to investigate the mechanism of interaction and its effect on the bookers' cancellation behavior.

As for model training and prediction, by dropping nuisance features and only considering those we deemed important, we improved our training results with gained efficiency. Therefore, we developed a more effective strategy in modeling.

In addition to applying the ML classification models mentioned above, we also investigated the advantages of supervised learning in our settings. We trained models with unsupervised learning methods, k-nearest neighbor and k-means [12] following the same procedure, and the accuracy turned out to be 0.83 and 0.67, respectively. The results indicate that unsupervised learning in this case can make prediction, but the efficiency are far more lower than supervised learning.

5. Conclusion. In this work, we investigated the hotel reservation data and developed predictors for hotel reservation cancellation using several ML classification models. In addition, ML models also reveals important factors that influence a cancellation decision, and helps in explaining how the influence takes place. Our work also shows that dropping unimportant features results in no loss in accuracy in general and provides a more efficient strategy in modeling.

REFERENCES

[1] G. Abrate, G. Fraquelli, and G. Viglia, *Dynamic pricing strategies: Evidence from european hotels*, International Journal of Hospitality Management, 31 (2012), pp. 160–168.

[2] C. K. Anderson, *Setting prices on priceline*, Interfaces, 39 (2009), pp. 307–315.

[3] V. F. Araman and R. Caldentey, *Dynamic pricing for nonperishable products with demand learning*, Operations research, 57 (2009), pp. 1169–1188.

[4] R. D. Badinelli, *An optimal, dynamic policy for hotel yield management*, European Journal of Operational Research, 121 (2000), pp. 476–503.

[5] L. Breiman, *Random forests*, Machine learning, 45 (2001), pp. 5–32.

[6] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*, CRC

press, 1984.

[7] L. V. Casalo, C. Flavian, M. Guinaliu, and Y. Ekinci, *Do online hotel rating schemes influence booking behaviors?*, International Journal of Hospitality Management, 49 (2015), pp. 28–36.

[8] C.-C. Chen, Z. Schwartz, and P. Vargas, *The search for the best deal: How hotel cancellation policies affect the search and booking decisions of deal-seeking customers*, International Journal of Hospitality Management, 30 (2011), pp. 129–135.

[9] C. Cortes and V. Vapnik, *Support-vector networks*, Machine learning, 20 (1995), pp. 273–297.

[10] A. V. Den Boer, *Dynamic pricing and learning: historical origins, current research, and new directions*, Surveys in operations research and management science, 20 (2015), pp. 1–18.

[11] Y. Freund, R. E. Schapire, et al., *Experiments with a new boosting algorithm*, in icml, vol. 96, Citeseer, 1996, pp. 148–156.

[12] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2, Springer, 2009.

[13] E. J. Kim, E. L. Kim, M. Kim, and S. Tanford, *Post-pandemic hotel cancellation policy: Situational cues as perceived risk triggers*, Journal of Hospitality and Tourism Management, 55 (2023), pp. 153–160.

[14] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, *Learning the kernel matrix with semidefinite programming*, Journal of Machine learning research, 5 (2004), pp. 27–72.

[15] P. Langley and S. Sage, *Induction of selective bayesian classifiers*, in Uncertainty Proceedings 1994, Elsevier, 1994, pp. 399–406.

[16] R. Law and S. Chan, *Internet and tourism-part xiv: hotels.com*, Journal of Travel & Tourism Marketing, 17 (2005), pp. 79–81.

[17] H. A. Lee, B. Denizci Guillet, and R. Law, *An examination of the relationship between online travel agents and hotels: A case study of choice hotels international and expedia. com*, Cornell Hospitality Quarterly, 54 (2013), pp. 95–107.

[18] L. Mason, J. Baxter, P. Bartlett, and M. Frean, *Boosting algorithms as gradient descent*, Advances in neural information processing systems, 12 (1999).

[19] D. Mattioli, *On orbitz, mac users steered to pricier hotels*, Wall Street Journal, 23 (2012), p. 2012.

[20] J. P. Mellinas, S.-M. M. María-Dolores, and J. J. B. García, *Booking. com: The unexpected scoring system*, Tourism Management, 49 (2015), pp. 72–74.

[21] L. Noriega, *Multilayer perceptron tutorial*, School of Computing. Staffordshire University, 4 (2005), p. 444.

[22] A. B. Ozturk, A. Bilgihan, K. Nusair, and F. Okumus, *What keeps the mobile hotel booking users loyal? investigating the roles of self-efficacy, compatibility, perceived ease of use, and perceived convenience*, International Journal of Information Management, 36 (2016), pp. 1350–1359.

[23] P. K. K. PK Kannan, *Dynamic pricing on the internet: Importance and implications for consumer behavior*, International Journal of Electronic Commerce, 5 (2001), pp. 63–83.

[24] J. R. Quinlan, *Induction of decision trees*, Machine learning, 1 (1986), pp. 81–106.

[25] D. W. Ruck, S. K. Rogers, and M. Kabrisky, *Feature selection using a multilayer perceptron*, Journal of neural network computing, 2 (1990), pp. 40–48.

[26] B. Scholkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, *Comparing support vector machines with gaussian kernels to radial basis function classifiers*, IEEE transactions on Signal Processing, 45 (1997), pp. 2758–2765.

[27] L. Wang, *Support vector machines: theory and applications*, vol. 177, Springer Science & Business Media, 2005.

[28] X. Zhao, L. Wang, X. Guo, and R. Law, *The influence of online reviews to online hotel booking intentions*, International Journal of Contemporary Hospitality Management, 27 (2015), pp. 1343–1364.