Improving hotel demand forecasting accuracy by integrating machine learning with pick-up methods

Yueqian (Rachel) Zhang1 and Chris Anderson2

1School of Hotel Administration, Cornell University, Ithaca, NY 14850 USA

2School of Hotel Administration, Cornell University, Ithaca, NY 14850 USA

Corresponding author: Yueqian (Rachel) Zhang (e-mail: yz2334@cornell.edu).

The data of work was supported by xxx Inc.

ABSTRACT A critical aspect of hotel revenue management is the ability to predict future demand. Pick-up methods, which scale current reservations on hand by parameters retrieved from historical observations, have been long adopted in the hotel industry because of its straightforwardness and conciseness. However, classic pick-up methods are not able to capture the complex relationship between current bookings and future demand. Therefore, this research proposes an innovative perspective, which embeds machine learning approaches in pick-up methods in hotel demand forecasting. By applying five mainstream machine learning methods, the empirical results show that the proposed models generate lower prediction errors than classical pick-up methods. It is notable that given higher accuracy, machine learning methods do have the disadvantages of longer computing time and harder to decipher. This research initiated the conversations on the potential of applying machine learning based approaches in hotel demand forecast.

KEYWORDS Forecasting, Hotel Revenue Management, Machine Learning, Advance Booking, Pick-up

INTRODUCTION

Hotel revenue consists of price and demand, and demand forecast is essential for successful revenue management since it has high uncertainties and has a direct impact on dynamic pricing. Different from industries such as retail where most of the bookings happen instantly when the clients arrive, the hotel industry usually sells the room ahead of the customer's arriving, and thus generates valuable information from the patterns between booking and arriving.

The pick-up method, an approach conducting forecast based on current realized bookings and historical booking patterns, is widely used both in the academia and in the industry. On top of the existed bookings so far, pick-up method predicts what will happen from today and the target future by estimating the incremental bookings. In practice, this method takes an average of the incremental bookings in history, or average the incremental ratios in percentage, then add on or multiple to the Reservations on Hand (ROH) today. The origin of the name of this approach is it estimates the number of incremental bookings "picked up" from today's reservation.

Pick-up methods are initially used in airline revenue management (L’heureux, 1986), and this concept has been rapidly applied in hotels due to the high similarities between the two industries. Generally, there are four types of pick-up models: additive & multiplicative pick-ups, and traditional & advanced pick-ups (Zakhary, Gayar, & Atiya, 2008). From the perspective of the relationship between ROH and final arrivals, Additive Pick-up models conduct prediction by adding an estimated incremental booking to the current ROH, while Multiplicative Pick-up models multiplied an average ratio on current ROH. On the other hand, pick-up models can also be categorized according to data completion. Traditional pick-up methods only use completed booking curves and ignored records where an arrival day is still beyond “today”. In comparison, advanced pick-up methods use both complete and incomplete booking information. Many researchers have tested the performances of pick-up models in different settings (Weatherford & Kimes, 2003), or combined pick-up models with other methodologies such as quadratic regression (Tse & Poon, 2015), exponential smoothing (Chen & Kachani, 2007), Poisson process (Lee, 2018), etc. However, most of these researches still rely on statistical assumptions to describe the relations between ROH and final arrivals. Given the uncertain demand change with various external factors' impact, the industry urges for more practice-adapting methods to improve accuracy.

Machine learning methods, on the other hand, have been picking up attention among both the industry and academia. Machine learning is a statistical method that “learns” from experience and automatically improve its calculation efficiency. Since the algorithm has the ability to capture any patterns from data, it usually has higher accuracy, is able to accommodate high-dimension data sources, and is free of statistical assumptions.

Even though machine learning has been extensively applied in different areas (business failing prediction, stock price, exchange rate, etc.), it has not been given full attention in the hotel industry. As GlobalData (2017) mentioned, hotel has not been one of the industries considered to be at the forefront of technological innovation. Some researchers have attempted using machine learning in the hotel industry such as online review analysis (Ma et al, 2018), hotel success indicators (Phillips et al. 2015), hotel online booking simulation (Corazza 2014), etc.

There are a few attempts in academia using machine learning to forecast hotel demand. Phumchusri & Ungtrakul (2020) compare the performance of ANN and Support Vector Regression (SVR) with other time series models and find that ANN outperforms other models with an mean absolute percentage error of 8.96%. Duan (2020) proposes WaveNet, a dilated causal convolution network method, to forecast hotel online sales. This approach generates a prediction error lower than 30%. However, this research uses aggregated hotel sales by province, and thus has less instructive insights for hotel properties. Sánchez-Medina & C-Sánchez (2020) use machine learning to predict if an individual traveler will cancel the hotel booking and their Artificial Neural Network (ANN) approach reach an 98% accuracy. Including 13 predictors in their study, this research aims at exploring relationships between external variables and the decision whether a cancellation will happen.

To our best knowledge, there are no existed research integrating machine learning with pick-up methods when forecasting hotel demand. However, we do deem this integration promising due to three reasons: firstly, the amount of transaction data in the hotel industry has been surging in the recent decade, which provides a foundation for machine learning models. Secondly, machine learning is able to capture the complex and non-parametric relations between existed bookings and final sales (Zhang, 2019). Hotel sales are impacted by multiple internal and external factors, and there has not been a clear relationship to capture those patterns. By simulating the arbitrary function from the data itself, machine learning approaches have the potential to better capture the complex relations. Last but not least, machine learning models are capable of dealing with high dimensional data (in the hotel industry’s case, the long days before arrivals). Given the current paucity of using machine learning in hotel demand forecasting, this study is able to provide insights on the prediction abilities of machine learning in the hotel setting, and make the first step to introduce this powerful approach in hotel revenue management.

This paper is arranged as follows: Section 2 discusses the theoretical background of both baseline pick-up models and machine learning embedded models. Section 3 introduces the empirical study where we applied the proposed models and tests their performances. Section 4 concludes the main takeaway from the empirical study and discusses the implication in both academia and the industry.

METHODOLOGY

*Pick-up Models*

Pick-up models estimate the increments of reservations over a time window then aggregate increments on top of the current reservations. Generally speaking, there are two types of pick-up models: additive pick-up and multiplicative pick-up. Additive pick-up models regard the final arrivals independent of the current ROH and calculate final arrivals by adding pick-ups to the current ROH. Instead, multiplicative pick-up models regard the ROH as a certain ratio of the final arrivals:

where is the forecasting target with the accumulated ROH on the arrival day (on DBA = 0). stands for the date with the newest reservation (usually “today” when the forecast is made). are the historical reservations observed on for day t where .

Linear regression wasn’t traditionally regarded as a pick-up model in hotel forecasting. However, the logic of linear regression is technically the combination of additive and multiplicative pick-up. By estimating both coefficients and intercept, linear regression model considers both additive (intercept) and multiplicative (coefficients) parts when describing the pattern relations:

(3)

All ­of the pick-up models can only accommodate the latest ROHs due to the restrictions of the statistical model. For instance, if include all previous ROHs in the regression model, it is highly likely to trigger multicollinearity.

*Machine Learning Embedded Pick Up Models*

There are various machine learning models and their variants widely used in different industries. In this research, we select six mainstream machine learning models and integrate them with pick-up approaches.

1. *Neural Network*

Neural Network is generally regarded as a multistep regression. It usually takes two steps: firstly, it extracts linear combinations of the inputs as derived features using activation function and secondly, models the target as a linear function of the derived features (2.2) (Friedman, Hastie, & Tibshirani, 2001):

,

,

A picture containing drawing

Description automatically generated

FIGURE 1.  Visualization of a Neural Network model using day of week of the predicting day and reservations on hand five days before.

In practice, it is usually necessary to scale the inputs before establishing the neural network model. Standardization all of the inputs ensures the model treats all inputs equally regardless of its numeric values. Besides, it is also critical to choose the appropriate number of hidden layers and the number of units in each layer. Without a standard rule of selecting the optimal number of hidden units and layers, the industry usually set the size of the hidden layers around 2/3 of the size of the input layer, also noting that the number of hidden units should not exceed the size of either input layer or output layer.

1. *Nearest Neighbors*

Nearest Neighbors models make forecasts by finding the most similar training data points of the target and then take the average of those "nearest neighbors" as the predicted value. K-nearest neighbor is the most common model in the Nearest Neighbors family. It calculates the distances between all training samples and the target, then makes the forecast by taking the average of the K nearest neighbors. On top of simply taking an arithmetic average, the weighted K-NN model assigned larger weights to closer neighbors with a shorter distance.

Using Nearest Neighbor methods, the model is trying to find the most similar ROH and DOW with the day of interest. Assuming only considering the newest ROH, the Euclidean distance between the day of interest and all other bookings with the same DBA:

It is critical to select the number of selected neighbors, the value of , to make a rational prediction. In practice, resampling method with the -fold cross-validation is usually used to select the optimal parameter. By splitting the data into roughly equal-sized folds, the model uses the first buckets to train the model with different parameter values, then tests the performances on the th fold.

1. *Tree*

Tree-based models conduct forecast by partitioning the features into a group of binary splits, then fit a simple model in each split. Decision tree, the most straightforward model among the tree models, recursively finds the optimal split for features, then make forecast using the average value of the feature space the target falls into. For DBA = , the decision tree model divides the values of into distinct and non-overlapping regions, . Each region is selected by trying out various splits. In other words, is split into and , then whichever generates the lowest RSS (Formula 3.2) is selected:

A close up of a device

Description automatically generatedwhere is the mean response for the observations within the th region. This recursive process continues until some thresholds (decreased RSS, the number of observations within each tree, etc.) are met.

FIGURE 2.

Visualization of a Decision Tree model using the day of week of the predicting day and reservations on hand five days before. Even day of week is given in the dataset, however, the tree model deems it less useful compared to information from the reservations on hand.

It is clear that decision tree is easy to understand since it mimics how the human brain works: chose among options then follow whatever the optimal option leads you. Another advantage is decision tree can accommodate both quantitative and qualitative variables without creating dummies for the latter. However, decision tree is usually not stable in the practical case, because the tree split will be driven by observations in the training set easily, and therefore may commit the fallacy of overfitting.

Random forest, a more advanced algorithm developed following decision tree, makes up decision tree’s flaw by reducing variance with a method called bootstrap. Random forests subsets predictors randomly then generate multiple trees using given predictors. The variance of decision tree models can be significantly decreased through this procedure since the random subset captures the complex interaction structures in the data. However, with the increased accuracy brought by the randomness, the interpretability of random forest models is lost. Very few insights can be driven from the random forest models besides its high prediction power.

1. *Support Vector Regression*

Different from tree models’ ‘square boxes’ partition, Support Vector Regression splits feature space using flexible boundaries. In other words, SVR draws non-parametric hyperplanes to divide features into various spaces. The optimal hyperplane is selected using the maximal margin rule, which finds the boundary farthest from the surrounding training samples.

The construct of hyperplane for SVR is beyond the scope of this research, however, this procedure can be conceptualized as manipulating input variables. For example, a quadratic boundary can be constructed through , and a polynomial boundary can be constructed through . More complicated space such as th degree polynomial, radial basis, Gaussian, etc. will transform the input variables into a new feature space, which makes it easier to find a separable hyperplane.

(Insert formula 3.15)

One of the main advantages of SVR is its ability to capture complicated data patterns without specific assumptions. The kernel trick SVR is using enables it to model highly non-parametric patterns between the predictors and response. For hotel demand forecasting, since many measurable or unmeasurable factors are impacting the demand, models like SVR have a higher ability to capture the "invisible" relationship between factors and results. However, on the other hand, since the hyperplane could be highly flexible, it is of difficulties to interpret the insights. Besides, since there are no rules of thumb to find the appropriate kernel shapes for SVR, practitioners usually use cross-validation methods which takes long computation time.

In summary, pick-up models are straightforward but leave out rich information from historical booking curves. Besides, pick-up models assume the relationship between current ROH and final arrivals are linear (either additive, multiplicative, or two combined). On the other hand, machine learning algorithms can utilize all information from historical ROHs and can thus capture invisible patterns. Machine learning models can also fit in highly non-parametric patterns, which in other words, capture the non-defined impacts from other factors outside the given input. As a drawback, machine learning models are less easy to interpret insights and articulate specific relations between certain variables and the final prediction.

(Insert a table to list the pro & cons for each model)

TABLE 1.

Properties of Machine Learning Models

|  |  |  |  |
| --- | --- | --- | --- |
| Models | Advantages | | Disadvantages |
| *Neural Network* | Non-parametric  No restrictions on input variables  Can handle high dimensional settings | | Hard to interpret the coefficients and hidden layer  Subjective to choose parameters (requires statistical expertise) |
| *Nearest Neighbor* | Can accommodate high dimensional data | Long computation time to find an optimal parameter | |
| *Decision Tree* | Intuitive, easy to interpret  Can handle qualitative predictors without creating dummy variables | | Low accuracy  Non-robust due to its binary splitting |
| *Random Forest* | Robust | | Hard to decipher |
| *Support Vector Regression* | Very flexible to capture the non-parametric patterns | | Hard to decipher  High computational cost |

EMPIRICAL STUDY

*Data Description*

An empirical study to demonstrate the performances of the proposed models is undertaken in collaboration with a hotel property.

This research uses one-year-long booking records of a hotel, with the arrival dates from December 27, 2017, to December 31, 2018, of a hotel property. For each, this dataset records the booking date (the day the client makes this reservation) and the arrival date (the day the client checks in). Hence, for each arrival date, the ROH can be calculated accumulatively from the earliest booking date of each day. For instance, if the hotel is predicting the number of final rooms sold on February 14, 2021, while "today" is January 1, 2021, the realized booking (ROH) can be calculated by adding up all of the reservations for February 14, 2021 happened before January 1. In this way, a new variable, the ROHs is calculated as the main independent variable.

Another derivative variable, days before arrival (DBA), is used to describe the time window between the booking day and the arrival day. This study examines 12 different horizons: 1, 2, 3, 4, 5, 6, 7, 14, 21, 30, 60, 90, and beyond. The cutoffs are made in this way since in the hotel industry, the dynamic pricing plan is usually set following those horizons. It is noticeable that during early periods when the booking day is far away from the stay date, the reservations accumulate very slow. Therefore, a wider horizon in earlier periods allows information to accumulate, while when the stay dates are approaching closer, the booking window is broken into smaller horizons for closer attention.

A picture containing object, table

Description automatically generated

FIGURE 2.

The reservations of a Single Hotel from December 27, 2017 to December 31, 2018.

The whole dataset has a booking record of 370 consecutive days with ROHs on 12 DBAs and the DOW of the arrival date. The dataset is randomly held out as 80% in the training set, 20% in the test set. For all the models, this study uses the nearest ROH and DOW as the independent variables on different arrival dates and DBA, to make predictions. The dependent variable here is the final accumulated reservations on the arrival day, in other words, the accumulative ROHs when DBA=0.

*Models*

To better understand the compatibility of various models, we bring up four assumptions (H1: H4) which will be tested in two experiments: one with the nearest ROH (formula 1.1), and the other with all realized ROHs on the booking curve (formula 1.2):

**Hypothesis 1 (H1):** machine learning embedded methods have higher accuracy when predicting future demands with the nearest ROH.

**Hypothesis 2 (H2):** machine learning embedded methods have higher accuracy when predicting future demands with the whole booking curve.

The experiments are designed to test the performances of the above models within the following relationships:

1.1

1.2

where represents three pick-up methods and six machine learning approaches respectively. The model constructing is iterated on DBAs at 1, 2, 3, 4, 5, 6, 7, 14, 21, 30, 60, and 90 days ahead. Taking DBA=5 as an example, the training set only uses the DOW and the newest ROH as input, and the predicting results are tested only given the relevant DOW and ROH accordingly. In other words, there are 12 models built using each method, and the performances are tested accordingly.

Table `pick-ups` presents the calculated pick-ups for the additive model, and table 3 displays the results for multiplicative pick-ups. Generally speaking, the longer the DBA, the larger the value and the higher level it will need to add on or multiple to. There are also visually significant differences among DOW on any given DBA.

Table `regression` shows the results of the regression models. The results are in line with our previously stated observation that the regression model combines both additive and multiplicative models: the longer the DBA, the higher the add-on value (as the intercept in regression), and the larger the weights to be multiplied on (as the coefficient in regression). As in the table, some of the DOWs are significant, and all of the nearest ROHs are significant (with p<0.001) in the model. When there are more days between today and the day the forecast is made, less information is given, therefore the standard deviations of the fitted coefficients are larger. Similarly, the explained variances of further models are smaller than those with closer DBA.

Machine learning models, on the other hand, have difficulties showing the specific coefficients or formula between the predictors and response. However, there are additional preprocessing and cleaning steps for each model. Besides, machine learning methods use cross-validation or bootcamp methods to increase accuracy. The specific preprocessing and cross-validation parameters are displayed as follows.

For the neural network model, numeric variables are standardized as follows to remove the effect of different numeric values on network weights:

where is the standard deviation of all in the training set.

For K-NN, the critical element, the value of was selected through cross-validation. The model firstly randomly selects 5 different values, then uses each of them to build a model. Beforehand, the training set was randomly split into 10 equal-size buckets. Nine of them were used to build models using , then the best performing model with was chosen to build the final model. The same cross-validation was applied to the weighted K-NN model as well. 10 randomly selected values were tested to find the optimal On top of selecting values, weighted K-NN also tests the kernel shapes. This research allows the models to test among the rectangular, triangular, epanechnikov, gaussian, rank, and optimal kernel shapes.

No additional pre-processing is needed for the decision tree model. For random forest, a 10-fold cross-validation was also applied to select the optimal number of variables used when growing each tree. Whichever number of variables generating the least Root Mean Square Error (RMSE) is selected. It worth noticing that a critical parameter when building a random forest is the number of variables selected to build every single tree, and that parameter is different pruned for experiments 1 and 2 (more to add depending on results).

For Support Vector Regression, a list of kernel shapes is tested manually: linear, polynomial, radial, and sigmoid. Through cross-validation, the model further tested gamma values of 0.1, 0.5, 1, and 2 with the radial kernel each time and selected the optimal gamma value to build the model.

*Results*

Table X presents the errors and computing time of the pick-up models and machine learning models. We select the Mean Error (ME), Mean Absolute Error (MAE), Standard Deviation Error (SDE), Mean Percentage Error (MPE), and Mean Absolute Percentage Error (MAPE) to measure model performances, as well as a computing time metric. The ME and MPE are used to describe the biases of the models. Generally speaking, an unbiased model tends to have a ME and MPE close to 0 when the number of observations is large enough. It is of more importance in the hotel industry to evaluate the biases of models since it plays a significant role in dynamic pricing. If a model tends to underestimate the demand, more tuning actions are needed in the pricing stage. MAE and MAPE are common metrics to measure the prediction accuracy of models. SDE is used to measure the variance of predicted values:

To test H1, only the newest ROH and DOW information is used to build the model. As the result indicates, most of the machine learning models tend to have lower biases (measured by ME and MPE). SVR, among all machine learning models, has a superior performance in accuracy (either MAE or MAPE) and variance (measured in SDE). However, machine learning models do take significantly longer modeling time. Similar results are confirmed in the robust test in Table X.

To have a closer look, Multiplicative Pick-up and Neural Network tend to under-estimate hotel demand, while all other models predict a higher demand. Except for Neural Network and K-NN, machine learning models tend to have MEs lower than 1 and MPEs lower than 0.13. In terms of accuracy, all models except multiplicative pick-up and neural network have MAEs from 5 to 6, and MAPE from 0.2 to 0.3. In other words, when predicting future hotel arrivals, both pick-up models and machine learning models tend to over-estimate the demand by around 5 rooms, among which the SVR model has the lowest error with 5.104. Similar results are presented in the robust test as well: SVR has the lowest MAE and SDE followed by Regression and Random Forest.

Therefore, we can accept H1 stating that machine learning models, specifically SVR, have higher performances in either bias, accuracy, or variance.

TABLE 2.

Models Performances when Predicting Hotel Demand with ROHs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | With the Newest ROH | | | | | |  | With the Whole Booking Curve | | | | | |
|  | ME | MAE | SDE | MPE | MAPE | Time (s) |  | ME | MAE | SDE | MPE | MAPE | Time (s) |
| Additive Pick-up | 1.302 | 5.372 | 6.673 | 0.164 | 0.251 | 0.390 |  | 1.302 | 5.372 | 6.673 | 0.164 | 0.251 | 0.390 |
| Multiplicative Pick-up | -1.202 | 7.154 | 9.142 | -0.039 | 0.232 | 0.383 |  | -1.202 | 7.154 | 9.142 | -0.039 | 0.232 | 0.383 |
| Regression | 1.064 | 5.262 | 6.546 | 0.144 | 0.238 | 0.218 |  | 1.064 | 5.262 | 6.546 | 0.144 | 0.238 | 0.133 |
| Neural Network | -14.218 | 14.294 | 7.179 | -0.398 | 0.405 | 77.073 |  | -14.276 | 14.554 | 8.114 | -0.397 | 0.415 | 146.707 |
| K-NN | 2.001 | 5.975 | 7.321 | 0.210 | 0.291 | 19.264 |  | 1.850 | 6.381 | 7.830 | 0.202 | 0.301 | 19.904 |
| Weighted K-NN | 0.984 | 5.754 | 7.238 | 0.132 | 0.247 | 3.462 |  | 1.285 | 6.268 | 7.769 | 0.173 | 0.286 | 2.952 |
| Decision Tree | 0.758 | 5.623 | 6.960 | 0.104 | 0.230 | 0.341 |  | 0.644 | 5.613 | 6.942 | 0.100 | 0.228 | 0.509 |
| Random Forest | 0.736 | 5.332 | 6.676 | 0.105 | 0.228 | 141.085 |  | 0.716 | 5.380 | 6.700 | 0.098 | 0.228 | 312.845 |
| SVR | 0.872 | 5.104 | 6.387 | 0.134 | 0.229 | 18.293 |  | 0.703 | 5.339 | 6.721 | 0.123 | 0.231 | 26.108 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | With the Newest ROH (Robust test) | | | | | |  | With the Whole Booking Curve (Robust Test) | | | | | |
|  | ME | MAE | SDE | MPE | MAPE | Time (s) |  | ME | MAE | SDE | MPE | MAPE | Time (s) |
| Additive Pick-up | 1.025 | 5.407 | 6.781 | 0.233 | 0.321 | 0.136 |  | 1.025 | 5.407 | 6.781 | 0.233 | 0.321 | 0.136 |
| Multiplicative Pick-up | -0.613 | 6.917 | 8.833 | -0.008 | 0.210 | 0.137 |  | -0.613 | 6.917 | 8.833 | -0.008 | 0.210 | 0.137 |
| Regression | 0.888 | 5.311 | 6.667 | 0.210 | 0.301 | 0.064 |  | 1.024 | 5.344 | 6.796 | 0.205 | 0.297 | 0.229 |
| Neural Network | -14.586 | 14.634 | 7.097 | -0.392 | 0.402 | 33.730 |  | -14.535 | 14.674 | 7.324 | -0.386 | 0.402 | 197.364 |
| K-NN | 1.630 | 5.775 | 7.536 | 0.308 | 0.393 | 18.631 |  | 1.464 | 6.318 | 8.175 | 0.311 | 0.408 | 24.257 |
| Weighted K-NN | 1.242 | 5.853 | 7.571 | 0.231 | 0.329 | 1.830 |  | 1.397 | 6.092 | 7.857 | 0.256 | 0.355 | 2.537 |
| Decision Tree | 1.016 | 5.606 | 7.015 | 0.201 | 0.299 | 0.096 |  | 1.034 | 5.612 | 7.062 | 0.201 | 0.299 | 0.156 |
| Random Forest | 0.942 | 5.360 | 6.814 | 0.165 | 0.265 | 174.460 |  | 0.901 | 5.394 | 6.771 | 0.166 | 0.268 | 330.700 |
| SVR | 0.818 | 5.201 | 6.570 | 0.212 | 0.304 | 40.203 |  | 1.044 | 5.399 | 6.839 | 0.215 | 0.308 | 21.096 |

To test H2, we build models with all existed ROHs on the realized booking curve. The models here for additive pick-up and multiplicative pick-up are identical to models where only the newest ROH is used. As a result, both pick-up models and regression perform superior in this experiment. Machine learning models overall still have lower biases, while regression has the lowest MAE of 5.252 followed by SVR with 5.339. Multiplicative pick-up generates a lower percentage errors, however, its variance is significantly higher than other models. Tree models have decent performances in this experiment as well, where the random forest model has the lowest ME, the second-lowest MPE, and the lowest SDE among all models, and it also generates a low MAE and MAPE following regression and SVR.

Similar results appear in the robust test when the training and test sets are differently split and parameter selections for machine learning models vary. Multiplicative models have lower biases and higher accuracy, however, it generates very high standard deviation errors. Overall, when all existed ROHs on the booking curve are used, regression, SVR, and Random Forest models tend to have higher accuracy and lower biases without generating unstable results. Therefore, we partially reject H2 which believes that machine learning models generate superior results since regression has decent performance as well.

Another critical element to consider here is the computing time. As shown in Table X, pick up models need very little time since they are simply algebraic calculations. In comparison, machine learning models tend to need a longer time due to model complexity and cross-validation. For instance, the time needed for constructing Random Forest is significantly higher since it needs repetitive iteration to prune the model. However, it is noticeable that the parameter selection process is flexible, thus the time consumed could be different. Take Support Vector Regression model as the example, in this current research, we tested for kernel shape to narrow down the optimal shape first, then it tested five values (low gamma value, far reach, lower weight to the near-boundary vectors, more linear; vice versa). This procedure can be largely different depends on the specific case and computation power.

To dig deeper into the model performances changing with time, we visualize the performance of five models which perform well considering all metric: regression, SVR, additive pick-up, and Random Forest. The longer the prediction window, the higher the error metrics. When only the newest ROH is used to predict demand, machine learning models lower biases and higher accuracy, especially between 7-30 days ahead. However, these distinctions are not significant for forecast variances.

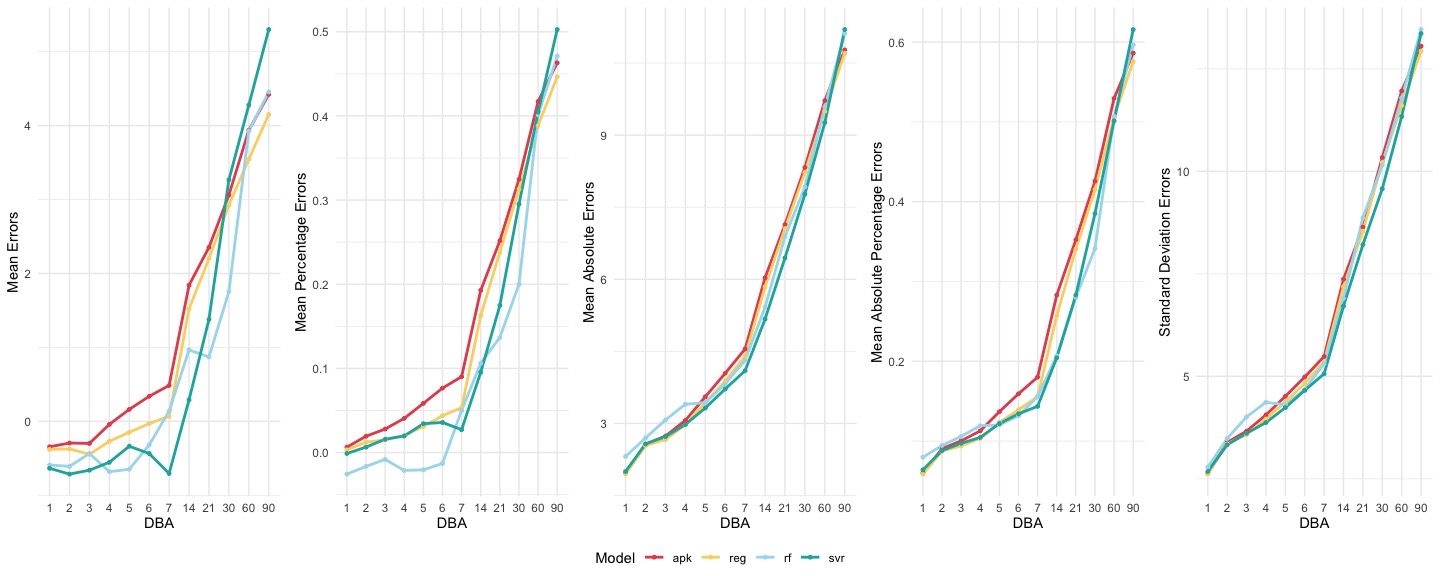


FIGURE 4.

Model Performances of Predicting Hotel Demand using the Newest ROHs by Forecasting Window.

*Note.* apk = additive pick-up, reg = regression, rf = random forest, svr = support vector machine, DBA=days before arrival

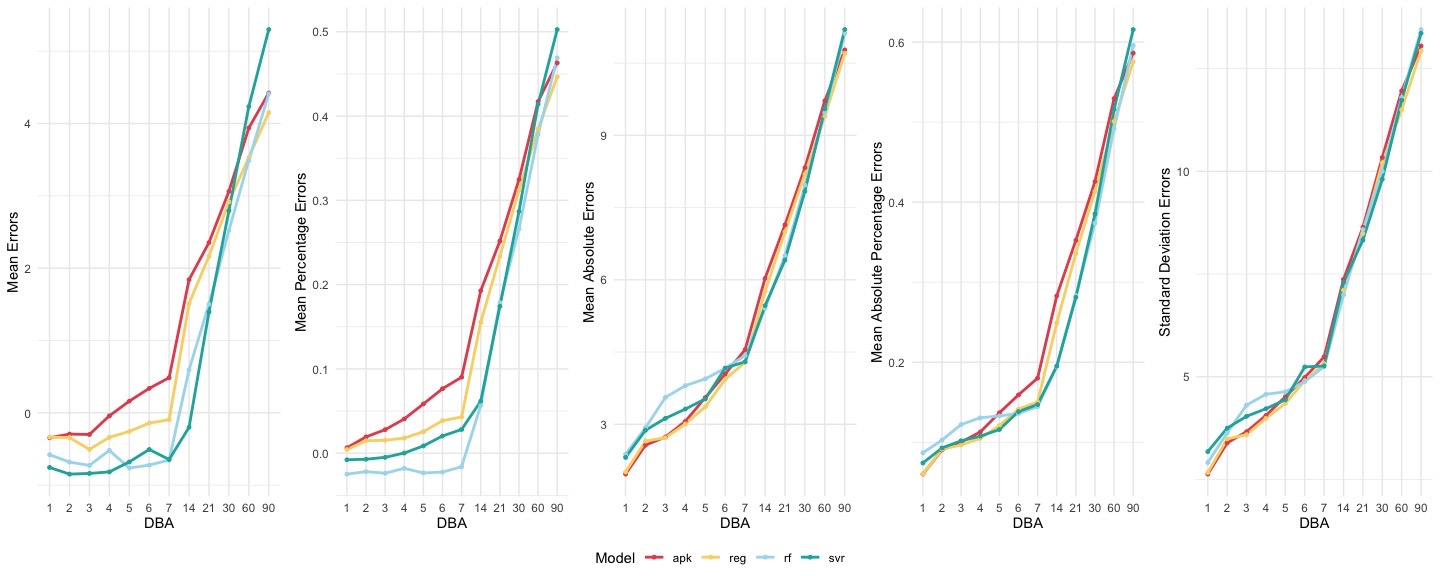


FIGURE 5.

Model Performances of Predicting Hotel Demand using All ROHs on the booking curve by Forecasting Window.

*Note.* apk = additive pick-up, reg = regression, rf = random forest, svr = support vector machine, DBA=days before arrival

For cases where all ROHs on the booking curves are used, the differences are less significant between models. Both random forest and SVR underestimate the demand in the 7-day window and have higher ME than additive pick-up and regression. However, random forest and SVR indeed perform better in terms of MPEs. Even though the random forest and SVR show slightly superiority in MAPE especially from the 7-30 days' window, there are no visually distinguishable differences. Similar to the last experiment, all models have similar variances increasing with booking window length.

CONCLUSIONS AND DISCUSSIONS

On average, pick up models embedded with machine learning approaches are shown to have potential in improving demand forecasting performances. Able to capture the complicated relationships between the newest ROHs and final arrivals, machine learning models are capable of fit the non-parametric relations and improve forecasting performances.

However, this improvement only appears when the newest ROHs are used. It is to our surprise that machine learning models do not show significant superiority when the whole booking curve (e.g. all existed ROHs) is given. Our guess is that even machine learning approaches generally can deal with high-dimensional data and complex forecasting problems, they need exquisite feature selection and parameter pruning to reach the optimum result. This guess is especially obvious for the Neural Network model, which has an averagely 14.29 MAE when a simple linear regression model has 5.26 MAE. In the empirical research, we simply choose the parameters (the number of hidden layers and hidden units) for Neural Network following a general rule without detailed pruning, and the results turn out to be significantly worse than baseline models.

Cross-validation, an approach that randomly splits the current data then receptively builds the model upon data subsets, can make up the challenge of parameter selection for machine learning. One of the machine learning models, the SVR, has higher accuracy and lower biases when predicting future demands than classic pick-up models, and it might be from the cross-validation procedure we apply to find the critic parameter (kernel shape and value). However, it is noticeable than this procedure takes significant computation power and can be challenging for practitioners without statistics background to apply. Given the normality of the hotel industry, we do not recommend this approach to be blindly applied in the daily management even machine learning is a hyped concept in recent years.

We do suggest researchers and practitioners utilize the insights from this current research. Regression models, as a combination of both additive pick-up and multiplicative pick-up, always have better performances than the two each. The additive pick-up model is suitable for situations that there are no significant changes between today and the future. Therefore, for properties or areas the hotel demand is usually regular and stable, the additive pick-up might be a good option. Multiplicative pick-up model, on the other hand, is more unstable, especially in far future forecasts. Given the multiplier is small when the booking window is long, multiplicative pick-up will generate extremely high predictions, which might drag the overall accuracy. However, for closer predictions within a week, multiplicative models do have similar performance as other models. Therefore, managers without profound statistical understandings and properties with less computational power, it is value-worthy to use regression models as the go-to approach.

For researchers refining for better forecasting performances, machine learning is a promising approach to pay more attention to. As an initial attempt to explore the application of machine learning in hotel demand forecasting, this research focuses more on the empirical side instead of model pruning. However, we do see the potential in feature engineering and parameter selection to improve the performances of machine learning.

REFERENCES

Duan, D. (2020). Research on hotel online sales forecast model based on improved WaveNet. *Journal of Physics: Conference Series*, *1544*(1), 0–7. https://doi.org/10.1088/1742-6596/1544/1/012067

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics New York.

Phumchusri, N., & Ungtrakul, P. (2020). Hotel daily demand forecasting for high-frequency and complex seasonality data: a case study in Thailand. *Journal of Revenue and Pricing Management*, *19*(1), 8–25. https://doi.org/10.1057/s41272-019-00221-6

Sánchez-Medina, A. J., & C-Sánchez, E. (2020). Using machine learning and big data for efficient forecasting of hotel booking cancellations. *International Journal of Hospitality Management*, *89*(May), 102546. https://doi.org/10.1016/j.ijhm.2020.102546