VILNIUS TECH
Vilnius Gediminas Technical University

Journal of Business Economics and Management
ISSN 1611-1699 / eISSN 2029-4433
2023 Volume 24 Issue 3: 594–613
https://doi.org/10.3846/jbem.2023.19775

# APPLICATION OF MACHINE LEARNING ALGORITHMS TO PREDICT HOTEL OCCUPANCY

Konstantins KOZLOVSKIS[1], Yuanyuan LIU[1*],
Natalja LACE[1], Yun MENG[2]

*[1]Department of Corporate Finance and Economics, Faculty of Engineering Economics and Management, Riga Technical University, Riga, Latvia*
*[2]School of Management, Guizhou University, Guizhou, China*

Received 28 June 2023; accepted 17 August 2023

**Abstract.** The development and availability of information technology and the possibility of deep integration of internal IT systems with external ones gives a powerful opportunity to analyze data online based on external data providers. Recently, machine learning algorithms play a significant role in predicting different processes. This research aims to apply several machine learning algorithms to predict high frequent daily hotel occupancy at a Chinese hotel. Five machine learning models (bagged CART, bagged MARS, XGBoost, random forest, SVM) were optimized and applied for predicting occupancy. All models are compared using different model accuracy measures and with an ARDL model chosen as a benchmark for comparison. It was found that the bagged CART model showed the most relevant results ($R^2 > 0.50$) in all periods, but the model could not beat the traditional ARDL model. Thus, despite the original use of machine learning algorithms in solving regression tasks, the models used in this research could have been more effective than the benchmark model. In addition, the variables' importance was used to check the hypothesis that the Baidu search index and its components can be used in machine learning models to predict hotel occupancy.

**Keywords:** bagged CART, bagged MARS, XGBoost, random forest, SVM, hotel occupancy.

**JEL Classification:** L10, L80, C10, C40.

## Introduction

Machine learning and data-driven methods have been widely used in research and practice of forecasting (Strielkowski et al., 2023). As a scientific technology of how computers mimic human learning behavior (Al Shehhi & Karathanasopoulos, 2020), it acquires new knowledge or experience and reorganizes existing knowledge structures to improve performance. Machine learning could learn the laws and patterns of massive data through computers, mining poten-

*Corresponding author. E-mail: *guizhouliuyuanyuan@163.com*

tial information, and is widely used to solve problems such as classification, regression, and clustering (Ahani et al., 2019; Aryai & Glodsworthy, 2023; Divasón et al., 2023; Kaya et al., 2022; Viverit et al., 2023). It is related to data learning that could help machines learn patterns from existing complex data, and is therefore widely used to forecast future behavioral outcomes and trends (Boriratrit et al., 2023; Gong et al., 2023; Kamm et al., 2023; Khalil et al., 2022; Kolomoyets & Dickinger, 2023; Sayed et al., 2023). Many types of machine learning methods have been published in research for decades. Multiple classification methods can be based on the emphasis on different learning strategies, such as (1) machine learning that simulates the human brain and (2) directly adopting mathematical methods. The mathematical methods mainly consist of three elements: statistical machine learning, model, strategy, and algorithm. A time series is a classic trend prediction method that assumes future values composed of patterns of current and historical data, and it is applied to constructing a model from historical data and then predict future data in machine learning (Caicedo-Torres & Payares, 2016; Mehmood et al., 2022; Qin et al., 2023; Sun & Lu, 2023). According to Calero-Sanz et al. (2022), Jiang et al. (2023), and Yang et al. (2015), the tourism sector, especially the hotel industry, has adopted machine learning for forecasting room booking and cancellation, demand, prices, and occupancy. This trend is approved by Huang and Zheng (2023), Li et al. (2020), Sánchez et al. (2020), Sánchez-Medina and Sánchez (2020), Koupriouchina et al. (2014), Al Shehhi and Karathanasopoulos (2020), and Zhai et al. (2023).

However, the hotel's corresponding forecasting accuracy has received widespread attention from scholars and industries. Hotel occupancy is the ratio of the total number of rooms a hotel occupies to the total number of rooms, usually expressed as a percentage. The hotel occupancy rate is one of the essential indicators to measure the hotel's operational status and management level, directly affecting the hotel's revenue and profits. Various factors, such as seasonal factors, pricing strategies, market demand, etc., influence hotel occupancy rates. In order to improve hotel occupancy, hotel managers can adopt various strategies, such as regular market analysis and competitive intelligence collection, formulating reasonable price strategies and promotion strategies, and improving customer satisfaction and loyalty. A hotel occupancy rate of over 60% is considered ideal, but different types of hotels and market demand can also impact the occupancy standards. Therefore, hotel occupancy prediction is an essential managerial tool for hotels.

Hotel occupancy prediction could use data analysis and machine learning algorithms to predict future occupancy rates of hotels. This prediction could help hotel managers formulate room prices, promotion strategies, and management strategies to meet market demand and improve hotel operating efficiency. Hotel occupancy forecast usually uses historical data, market trends, and other information to conduct model training and forecast. Prediction algorithms include time series analysis, regression analysis, neural network, random forest, etc. These algorithms could predict future occupancy rates by learning and predicting historical data while evaluating the impact of different factors on occupancy rates. Hotel managers can also use online prediction tools to predict occupancy rates. These tools usually use real-time data and market trends to make predictions and provide suggestions and decision support based on the prediction results. However, one of the worst impacts of the COVID-19 on the hotel industry, the hotel passenger flow has declined significantly. Due to the uncertainty of mobility control policies, travel is sometimes allowed but randomly controlled. Many tourists whose mobility has been restricted for a long time and who start planning to travel suddenly cancel their plans due to the lockdown of their residential areas or tourist destinations. The hotel customers flow significantly decreases and seriously impacts occupancy rates and profits. It leads to increased hotel operating costs, adjustments to hotel marketing strategies, hotel formats, and service models, and challenges the accuracy of hotel occupancy forecast models.

This paper aims to apply several machine learning algorithms to predict high frequent daily hotel occupancy at a Chinese hotel. Five machine learning algorithms (bagged CART, bagged MARS, XGBoost, random forest, and SVM) were used to achieve the research purpose. Usually, these methods are intensively used for solving the tasks of classification. However, in this paper, these methods were used unusually for solving the regression tasks (all these algorithms can be applied for classification as for regression). In addition, all five machine learning tools were compared with a benchmark represented by a traditional econometric model based on the ARDL methodology. Also, the following hypotheses were checked:

1. The machine learning tools are more effective than traditional methods of forecasting (ARDL model) and should show higher forecasting power.
2. The Chinese Baidu search engine index and its components can be used for predicting hotel occupancy.

The paper is organized in the following way. The actualization of the research conducted is given in the introduction part. The “Bibliographic mapping” provides the results of the network visualization based on the bibliometric data obtained from the Web of Science database using two searching keywords, “machine learning” and “hotel occupancy”. The next chapter, “Description of variables used in the research”, explains the variables that significantly impacted tourism demand. A description of the machine learning algorithms used follows this chapter. The measures of model accuracy allow us to compare the models between themselves as well as to assess their forecasting power. It is shown in chapter four. The presentation of the research results is shown in chapter five. The conclusion puts forward the main discoveries related to machine learning algorithms’ forecasting power (accuracy) and the results of testing hypotheses.

## 1. Bibliographic mapping

Figure 1 shows the results of the network visualization based on the bibliometric data obtained from the Web of Science database using two searching keywords, “machine learning” and “hotel occupancy”. In the network visualization, all items related to two keywords in the scientific papers are represented by words (labels) with the highest frequency matched in the articles. The font size of the label and the square of the circle of any item is determined by the item’s weight in the total number of found words. The higher the frequency (weight) of an item, the larger the font size of the label and the square covered by the circle of the item. The different colors of items describe clusters to which the corresponding item belongs. Colorful lines represent the links between the items. In addition, the distance between two items approximately indicates the relatedness of the items, i.e., the closer are two items, the stronger their relatedness is (van Eck & Waltman, 2023).

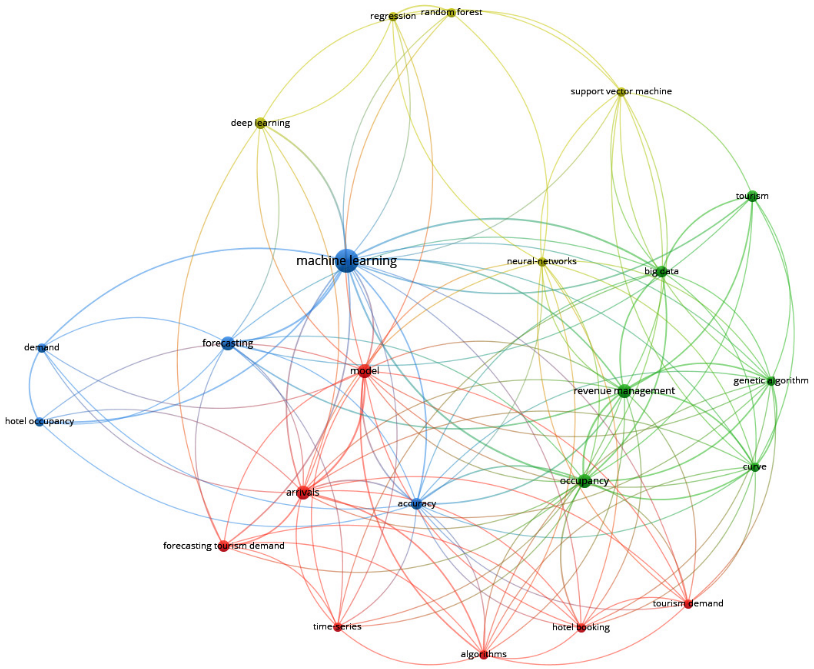


Figure 1. Network visualization

The results shown in Figure 1 can be analyzed by clusters highlighted in different colors:

1. Algorithms. The most frequent words are random forest, deep learning, neural networks, and support vector machine. All these words show the main trends in implementing the corresponding machine learning methods to predict hotel occupancy.
2. Fields of forecasting. There are four main words associated with forecasting: hotel occupancy, demand, accuracy, and forecasting itself.
3. A cluster of words "hotel", "algorithms", "model", "arrivals", "timeseries", "forecasting tourism demand" can be considered as evidence of the usage of different algorithms and models based on time series of data to predict hotel occupancy or demand.
4. The last green cluster links occupancy, revenue management, tourism, big data, and genetic algorithm. Such a relationship between the words can be decoded as the usage of big data in tourism, particularly in occupancy, leads to more effective revenue management, and some genetic algorithms (probably as a class of optimizers) are used in this direction.

Bibliographic mapping through network visualization discovered that machine learning algorithms such as random forest, support vector machine, neural network (recently defined as deep learning algorithms) are widely used in hotel business, including forecasting of hotel occupancy. Following the trend, similar methods were used in this research.

## 2. Description of variables used in the research

Analyzing scientific literature, Lim (1997) showed that over 1961–1994 a various number of variables had a significant impact on tourism demand, such as income, tourism prices, relative prices, transportation costs, exchange rates, sociodemographic characteristics, ethnic or immigration factors, destination marketing, tourism attractiveness, and climate. Kim (2010) expanded the list of variables by discovering the significance of traveler data from luxury hotels in Seoul, South Korea. Yang et al. (2014) conducted one of the most exciting research, showing how Baidu search data allowed us to predict the passenger flow received by tourist attractions in Hainan Province, China.

This paper used five main variables and their derivatives to predict hotel occupancy from July 1, 2017, to Nov 30, 2021, in a daily format. Table 1 shows the description of the variables.

Table 1. A list of five main variables

| # | Variable | Description |
|---|---|---|
| 1 | Hotel occupancy | This is the target of interest. The variable is used as a dependent variable in the models considered in this paper. |
| 2 | Weekend | It is a dummy variable: 1 if Saturday or Sunday and 0, otherwise. |
| 3 | Temperature | It represents the weather conditions by the average temperature in the region. The values are obtained from tianqihoubao.com. |
| 4 | Public holidays | It is a dummy variable: 1 if national holiday and 0, otherwise. The information on the public holidays formulated by the General Office of the National People's Congress (NPC) over the study period was obtained from www.gov.cn. |
| 5 | Baidu search index and 5 search keywords | Consisted of the Baidu index itself and five components: "Guiyang tourism", "Guiyang tourism", "Guiyang food", "Guiyang air ticket", "Guiyang tourist attractions", "Guiyang hotels".<br>These keywords can reflect tourists' travel motives and needs and support conducting subsequent research. |

In addition to the main five variables, the following derivatives were added:

– differenced variables of temperature, Baidu search index, as well as differenced values of five Baidu search index components;

– lagged variables of temperature, Baidu search index, and its five components. The lag is from 1 to 7.

Differenced and lagged variables were added to the set of variables based on the results of the autocorrelation function analysis (see Figures 2–9). Except for hotel occupancy and Guiyang air ticket keyword index value, all other variables show a clear presence of a trend by decaying the ACF values slowly. Based on the wave structure of the ACF of hotel occupancy, it can be concluded that the variable contains a seasonal component (maximal peaks are repeated each seventh day) (see Figure 2). The behavior of Guiyang air ticket keyword index values is similar to hotel occupancy but does not present a clear view of seasonality (see Figure 4).

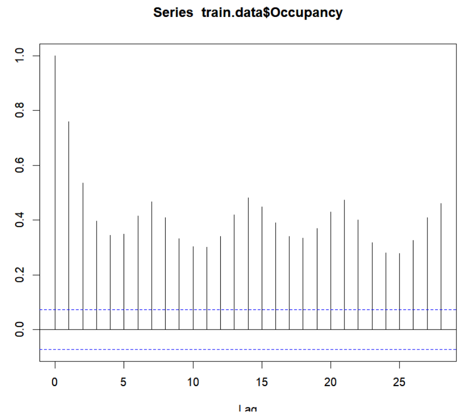


Figure 2. Hotel occupancy

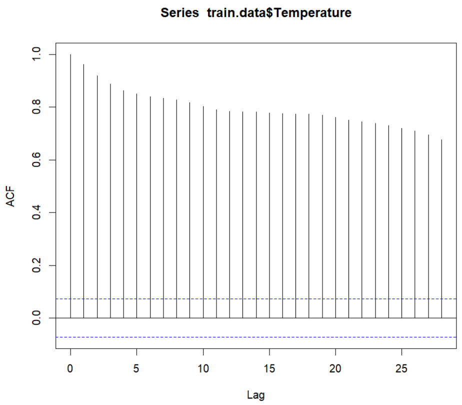


Figure 3. Temperature

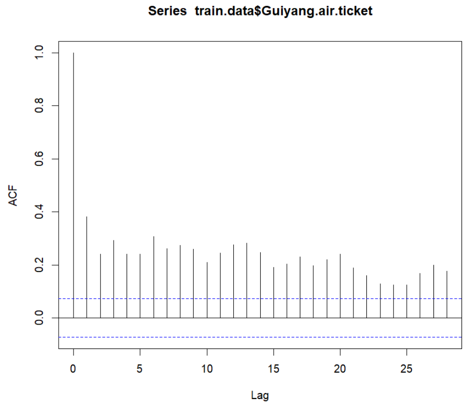


Figure 4. Guiyang air ticket

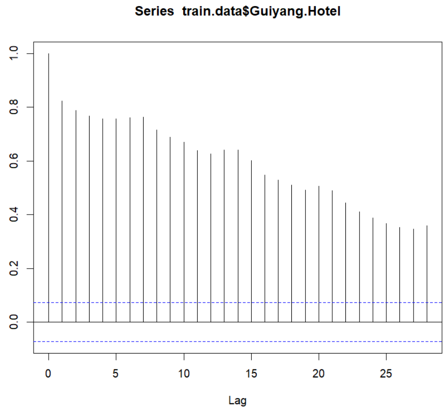


Figure 5. Guiyang Hotel

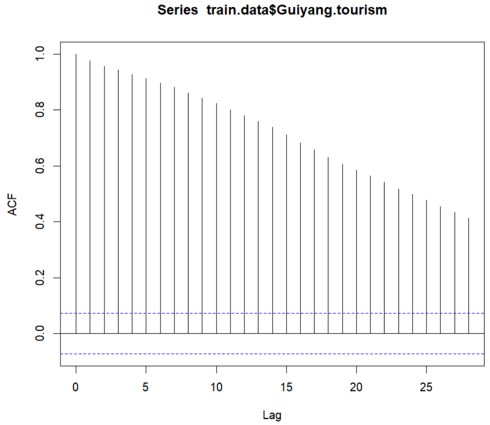


Figure 6. Guiyang tourism

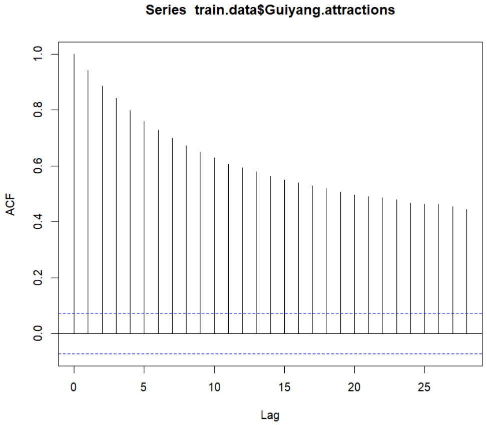


Figure 7. Guiyang attractions

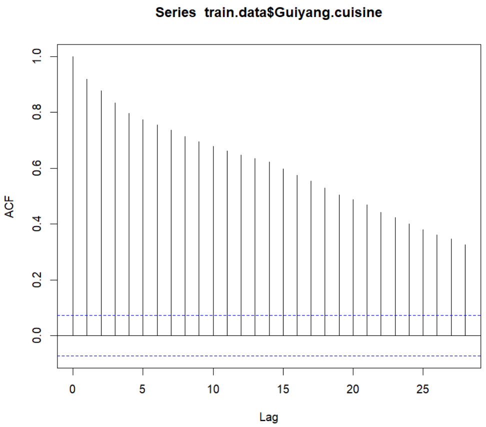


Figure 8. Guiyang cuisine

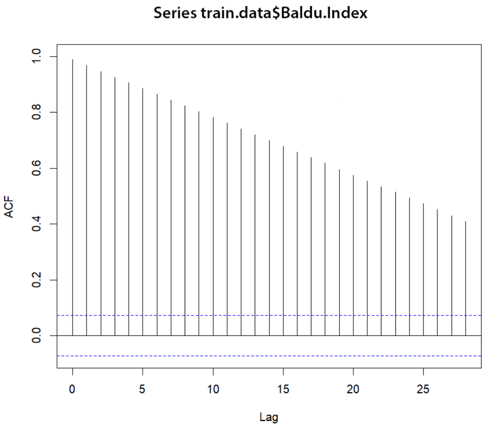


Figure 9. Baidu index

A seasonal component at lag seven also offered to create lagged variables. It should be noted that the necessity to have differenced and lagged variables is not required by tools from the field of machine learning. Nevertheless, an enhanced set of variables can lead to better results.

## 3. Methods and datasets

### 3.1. Machine learning algorithms

In this research, the following machine learning algorithms were used:

1. Bagged CART;
2. Bagged MARS;
3. XGBoost;
4. Random Forest;
5. SVM.

Bagging or bootstrap aggregation is a specific methodology for reducing the forecasting error of learning algorithms. Breiman presented empirical evidence that bagging can reduce prediction error (Buja & Stuetzle, 2006). In general, bagging is realized in several steps:

1. Creating bootstrap samples from the training sample;
2. Applying the corresponding learning algorithm to each bootstrap sample;
3. Predicting by aggregating (usually averaging) the predicted values for test observations.

It was proved that bagging is highly effective for CARTs (classification and regression trees) (Breiman, 1984).

Bagged CART was realized using the *bagging* function from the *ipred* library of R.

Bagged MARS is multivariate adaptive regression splines with bagging and is usually used for solving complex non-linear regression problems. As Friedman wrote in his article, “the model takes the form of an expansion in product spline basis functions, where the number of basis functions as well as the parameters associated with each one (product degree and knot locations) are automatically determined by the data" (Friedman, 1991).

Bagged MARS was realized using the *earth* function from the *earth* library of R.

Extreme gradient boosting package called xgboost is an efficient and scalable implementation of gradient boosting framework referencing Friedman (2001) and Friedman et al. (2000). This package can solve as classification tasks as regression ones (Chen & He, 2023).

XGBoost was realized using the *xgboost* function from the *xgboost* library of R.

Random forest is a supervised machine learning algorithm that can also be used for solving regression tasks based on a group of decision tree models (Afriyie et al., 2023). In practice, random forest is characterized by accurate predictions, estimation of variables' importance, etc. (Prajwala, 2015)

Random forest was realized using the *randomForest* function from the *randomForest* library of R.

Support vector machine, or support vector regression, is the adapted form of SVM when the dependent variable is numerical rather than categorical. The basic principle of SVR is to map the indistinguishable sample data from low dimension to high dimension, in which the indistinguishable sample data become distinguishable using the kernel function. Then, the SVR establishes a decision function based on the theory of structural risk minimization for regression analysis on distinguishable sample data (Sun et al., 2023).

SVM was realized using the *svm* function from the *e1071* library of R.

The period of research covers 1,612 observations from Jul 1, 2017, to Nov 30, 2021. The whole period is divided into three parts called "before", "during", and "after" regarding the situation with COVID-19 in China. The periods of "during" and "after" COVID-19 are defined with the 07/2020 announcement by the Guizhou Provincial Tourism and Culture Department of the full resumption of cross-province (region and city) group travel in Guizhou as the time division point.

All models were trained on the first 80% of data (724 observations) in the before-COVID-19-time-period. After that, all models were tested on the other three sets (see Figure 10).

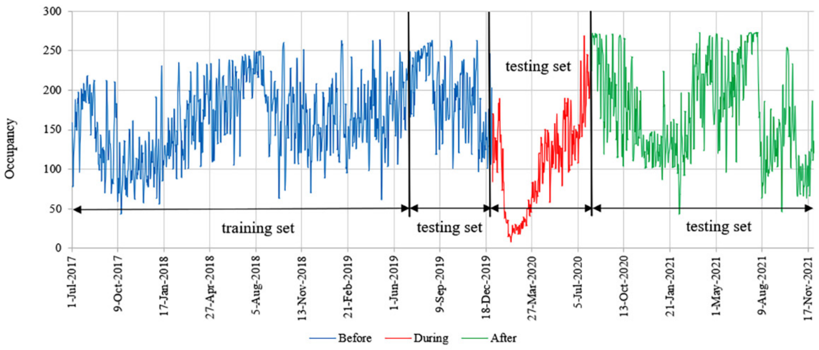
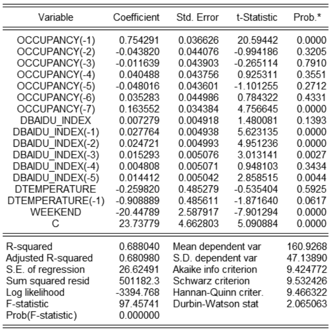


Figure 10. Hotel occupancy over Jul 1, 2017 – Nov 30, 2021

The first testing set belongs to the before-COVID-19-time-period and represents the last 20% of data (183 observations). The next during-COVID-19-time-period (red line in Figure 10) covers 212 observations. The third testing period represents the after-COVID-19-time-period containing 486 observations. Three testing periods were used separately to assess the forecasting power of the models in each period and to understand if there were some changes in the dynamics of hotel occupancy.

### 3.2. Benchmark

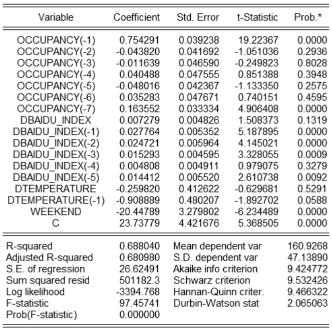
An ARDL model was chosen as a benchmark for comparison with machine learning tools. The ARDL model was built on the same training set as other models. Estimating the ARDL model in eViews 12, the following result was obtained (see Figure 11 and Figure 12) using automatic selection:



| Variable | Coefficient | Std. Error | t-Statistic | Prob.* |
|---|---|---|---|---|
| OCCUPANCY(-1) | 0.754291 | 0.036626 | 20.59442 | 0.0000 |
| OCCUPANCY(-2) | -0.043820 | 0.044076 | -0.994186 | 0.3205 |
| OCCUPANCY(-3) | -0.011639 | 0.043903 | -0.265114 | 0.7910 |
| OCCUPANCY(-4) | 0.040488 | 0.043756 | 0.925311 | 0.3551 |
| OCCUPANCY(-5) | -0.048016 | 0.043601 | -1.101255 | 0.2712 |
| OCCUPANCY(-6) | 0.035283 | 0.044986 | 0.784322 | 0.4331 |
| OCCUPANCY(-7) | 0.163552 | 0.034384 | 4.756645 | 0.0000 |
| DBAIDU_INDEX | 0.007279 | 0.004918 | 1.480081 | 0.1393 |
| DBAIDU_INDEX(-1) | 0.027764 | 0.004938 | 5.623135 | 0.0000 |
| DBAIDU_INDEX(-2) | 0.024721 | 0.004993 | 4.951236 | 0.0000 |
| DBAIDU_INDEX(-3) | 0.015293 | 0.005076 | 3.013141 | 0.0027 |
| DBAIDU_INDEX(-4) | 0.004808 | 0.005071 | 0.948103 | 0.3434 |
| DBAIDU_INDEX(-5) | 0.014412 | 0.005042 | 2.858515 | 0.0044 |
| DTEMPERATURE | -0.259820 | 0.485279 | -0.535404 | 0.5925 |
| DTEMPERATURE(-1) | -0.908889 | 0.485611 | -1.871640 | 0.0617 |
| WEEKEND | -20.44789 | 2.587917 | -7.901294 | 0.0000 |
| C | 23.73779 | 4.662803 | 5.090884 | 0.0000 |

| | | | |
|---|---|---|---|
| R-squared | 0.688040 | Mean dependent var | 160.9268 |
| Adjusted R-squared | 0.680980 | S.D. dependent var | 47.13890 |
| S.E. of regression | 26.62491 | Akaike info criterion | 9.424772 |
| Sum squared resid | 501182.3 | Schwarz criterion | 9.532426 |
| Log likelihood | -3394.768 | Hannan-Quinn criter. | 9.466322 |
| F-statistic | 97.45741 | Durbin-Watson stat | 2.065063 |
| Prob(F-statistic) | 0.000000 | | |

Figure 11. ARDL(7, 5, 1)



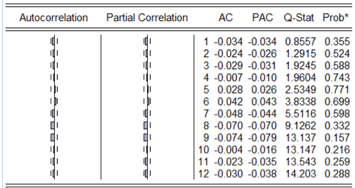
| Variable | Coefficient | Std. Error | t-Statistic | Prob.* |
|---|---|---|---|---|
| OCCUPANCY(-1) | 0.754291 | 0.039238 | 19.22367 | 0.0000 |
| OCCUPANCY(-2) | -0.043820 | 0.041692 | -1.051036 | 0.2936 |
| OCCUPANCY(-3) | -0.011639 | 0.046590 | -0.249823 | 0.8028 |
| OCCUPANCY(-4) | 0.040488 | 0.047555 | 0.851388 | 0.3948 |
| OCCUPANCY(-5) | -0.048016 | 0.042367 | -1.133350 | 0.2575 |
| OCCUPANCY(-6) | 0.035283 | 0.047671 | 0.740151 | 0.4595 |
| OCCUPANCY(-7) | 0.163552 | 0.033334 | 4.906408 | 0.0000 |
| DBAIDU_INDEX | 0.007279 | 0.004826 | 1.508373 | 0.1319 |
| DBAIDU_INDEX(-1) | 0.027764 | 0.005352 | 5.187895 | 0.0000 |
| DBAIDU_INDEX(-2) | 0.024721 | 0.005964 | 4.145021 | 0.0000 |
| DBAIDU_INDEX(-3) | 0.015293 | 0.004595 | 3.328055 | 0.0009 |
| DBAIDU_INDEX(-4) | 0.004808 | 0.004911 | 0.979075 | 0.3279 |
| DBAIDU_INDEX(-5) | 0.014412 | 0.005520 | 2.610738 | 0.0092 |
| DTEMPERATURE | -0.259820 | 0.412622 | -0.629681 | 0.5291 |
| DTEMPERATURE(-1) | -0.908889 | 0.480207 | -1.892702 | 0.0588 |
| WEEKEND | -20.44789 | 3.279802 | -6.234489 | 0.0000 |
| C | 23.73779 | 4.421676 | 5.368505 | 0.0000 |

| | | | |
|---|---|---|---|
| R-squared | 0.688040 | Mean dependent var | 160.9268 |
| Adjusted R-squared | 0.680980 | S.D. dependent var | 47.13890 |
| S.E. of regression | 26.62491 | Akaike info criterion | 9.424772 |
| Sum squared resid | 501182.3 | Schwarz criterion | 9.532426 |
| Log likelihood | -3394.768 | Hannan-Quinn criter. | 9.466322 |
| F-statistic | 97.45741 | Durbin-Watson stat | 2.065063 |
| Prob(F-statistic) | 0.000000 | | |

Figure 12. ARDL(7, 5, 1) under HAC correction

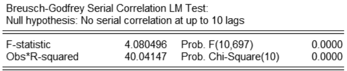
Checking the Gauss-Markov conditions, it was concluded that:

1. The mean of the residuals is –3.75 × 10 – 15, i.e., zero.
2. The existence of serial correlation in the residuals was checked by correlogram Q-statistics (see Figure 13) and the Breusch-Godfrey serial correlation LM test (see Figure 14). Although the Q-statistics did not show any serially correlated residuals ($p$-values > 0), it was decided to remediate serial correlation in the residuals by HAC correction because the second test showed the presence of serial correlation in the model residuals ($p$-value equals zero).



| Autocorrelation | Partial Correlation | | AC | PAC | Q-Stat | Prob* |
|---|---|---|---|---|---|---|
| | | 1 | -0.034 | -0.034 | 0.8557 | 0.355 |
| | | 2 | -0.024 | -0.026 | 1.2915 | 0.524 |
| | | 3 | -0.029 | -0.031 | 1.9245 | 0.588 |
| | | 4 | -0.007 | -0.010 | 1.9604 | 0.743 |
| | | 5 | 0.028 | 0.026 | 2.5349 | 0.771 |
| | | 6 | 0.042 | 0.043 | 3.8338 | 0.699 |
| | | 7 | -0.048 | -0.044 | 5.5116 | 0.598 |
| | | 8 | -0.070 | -0.070 | 9.1262 | 0.332 |
| | | 9 | -0.074 | -0.079 | 13.137 | 0.157 |
| | | 10 | -0.004 | -0.016 | 13.147 | 0.216 |
| | | 11 | -0.023 | -0.035 | 13.543 | 0.259 |
| | | 12 | -0.030 | -0.038 | 14.203 | 0.288 |

Figure 13. ACF of the model residuals

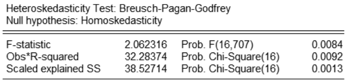


Breusch-Godfrey Serial Correlation LM Test:
Null hypothesis: No serial correlation at up to 10 lags

| | | | |
|---|---|---|---|
| F-statistic | 4.080496 | Prob. F(10,697) | 0.0000 |
| Obs*R-squared | 40.04147 | Prob. Chi-Square(10) | 0.0000 |

Figure 14. ACF of the model residuals

The problem of the heteroskedastic residuals ($p$-value < 0.05) (see Figure 15) was remediated by the HAC correction (see Figure 12).



```
Heteroskedasticity Test: Breusch-Pagan-Godfrey
Null hypothesis: Homoskedasticity

F-statistic            2.062316   Prob. F(16,707)          0.0084
Obs*R-squared          32.28374   Prob. Chi-Square(16)     0.0092
Scaled explained SS    38.52714   Prob. Chi-Square(16)     0.0013
```

Figure 15. Test on homoskedasticity of the residuals

The correlation coefficient between *d(Baidu index)* and *d(Temperature)* is 0.14, with the *p*-value = 0, signalizing about the lack of the multicollinearity problem (the correlation coefficient <0.7).

As a result, the model used as a benchmark can be mathematically expressed as follows:

$$\begin{gathered}Occupancy_t = 0.7543 \times Occupancy_{t-1} - 0.0438 \times Occupancy_{t-2} - \\ 0.0116 \times Occupancy_{t-3} + 0.0405 \times Occupancy_{t-4} - 0.0480 \times Occupancy_{t-5} + \\ 0.0353 \times Occupancy_{t-6} + 0.1636 \times Occupancy_{t-7} + d(Baidu_{index_t}) + 0.0278 \times \\ d(Baidu_{index_{t-1}}) + 0.0247 \times d(Baidu_{index_{t-2}}) + 0.0153 \times d(Baidu_{index_{t-3}}) + \\ 0.0048 \times d(Baidu_{index_{t-4}}) + 0.0144 \times d(Baidu_{index_{t-5}}) - 0.2598 \times d(Temperature_t) - \\ 0.9089 \times d(Temperature_{t-1}) - 20 \times Weekend_t + 23.7378.\end{gathered} \tag{1}$$

The value of hotel occupancy calculated by Equation (1) was rounded down.

### 3.3. Variable importance

The importance of variables was used to check the hypothesis if the Baidu search index values and its components can be utilized to predict hotel occupancy. The values of the assessments are obtained automatically by using the corresponding function of R. The measurement units and the mathematical apparatus used in each case can be found with the help of the corresponding function. All functions that participated in extracting the importance of variables are mentioned in the main text within each machine learning tool. The main focus was paid to the order of variables' importance based on their quantitative assessments provided by the corresponding functions.

## 4. Measures of model accuracy

The measures of model accuracy allow us to compare the models between themselves as well as to assess their forecasting power. Table 2 displays all widely used measures used in this

Table 2. Measures of model accuracy

| # | Metrics | Formula | Comments |
|---|---|---|---|
| 1 | Forecast bias | $\Sigma\left(\hat{y}_t - y_t\right)$ | Because the forecast bias is the difference between the forecasted value and the actual one, it is clearly seen that the positive bias signalizes about overestimated forecasts; if it is negative, then the forecast is underestimated. |
| 2 | R$^2$ | $1 - \frac{\sum_{t=1}^{n}\left(\hat{y}_t - y_t\right)^2}{\sum_{t=1}^{n}\left(\bar{y} - y_t\right)^2}$ | |

paper. In this table, ˆ is the forecasted value in time period *t*, $y_t$ is the actual value in time period *t*, $\overline{y}$ is the mean of *y*.

The primary focus is on the coefficient of determination ($R^2$) as the simplest value for understanding the model's forecasting power.

## 5. Empirical results

As previously mentioned, all models were built on the first 80% of the before-COVID-19 time period covering the first 731 observations from July 1, 2017 to July 1, 2019 and containing 68 variables.

### 5.1. Bagged CART

The bag function contains an internal parameter *minsplit*, which refers to the minimum number of observations required at each node to split further. Conducting some research on the dependence between the values of this parameter and model quality measure (see Figure 16), it was found that the minimal out-of-bag estimate of root mean squared error equaled 31.22 is reached when *minsplit* = 10. Table 3 displays the TOP-10 of variable importance in the model. Except temperature, all variables are related to people's activity in the Baidu search engine.

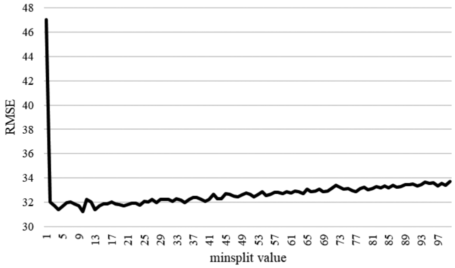


Figure 16. Model RMSE vs. *minsplit* value

The values of importance were calculated automatically by the *varImp* function from the *caret* library. As seen, except temperature all variables are related to people activity in the Baidu search engine.

Table 3. TOP-10 of variable importance (bagged CART)

| # | Variable | Importance |
|---|---|---|
| 1 | Guiyang.attractions | 7.21 |
| 2 | dGuiyang.cuisine | 6.65 |
| 3 | Guiyang.cuisine | 6.64 |
| 4 | Guiyang.Hotel | 6.53 |
| 5 | dGuiyang.attractions | 6.36 |
| 6 | dGuiyang.tourism | 6.21 |

*End of Table 3*

| # | Variable | Importance |
|---|---|---|
| 7 | Temperature | 6.15 |
| 8 | Guiyang.air.ticket | 6.09 |
| 9 | Guiyang.tourism | 6.02 |
| 10 | dGuiyang.Hotel | 5.90 |

Applying the optimal bagged CART model to the 20% of data left in the before-COVID-19 time period as well as on during- and after- COVID-19 time periods, the following results were obtained in the context of assessing model accuracy (see in Table 4).

Table 4. Assessment of model accuracy (bagged CART)

| Period | Before (80%) | Before (20%) | During | After |
|---|---|---|---|---|
| Forecast bias | –229 | –1,452 | 3,729 | –4,952 |
| $R^2$ | 0.92 | 0.53 | 0.54 | 0.68 |

The negative value of forecast bias on the training data is smaller than –1,452 on the last 20% of data in the before-COVID-19 period. The negative value signalizes the underestimation of the forecasts. During the extremal times when the COVID-19 pandemic was officially announced, the value of forecast bias became positive, showing that the model started to overestimate. It is logical because significantly falling demand for hotel rooms and total restrictions seriously decreased hotel occupancy, and the model could not follow the tendencies so quickly. However, in the after-COVID-19 time period the value of forecast bias significantly fell, reaching –4,952, signaling colossal underestimation. Nevertheless, in all time periods, the model could describe more than 50% of the initial variance: 53% on the testing data, 54% in the during-COVID-19 time period, and 68% after.

### 5.2. Bagged MARS

A basic bagged MARS model was built using the earth function from the earth library and optimized by RMSE using a searching grid. Two parameters were changed: *degree* (potential interactions between different hinge functions) and *nprune* (number of terms to retain). The best model has *nprune* = 23 and *degree* = 1. Based on the best model the TOP-10 of variable importance was obtained (see Table 5). The values of importance were obtained using the *vip* function from the *vip* library.

Table 5. TOP-10 of variable importance (bagged MARS)

| # | Variable | Importance |
|---|---|---|
| 1 | Baidu.index_1 | 100 |
| 2 | Guiyang.tourism_6 | 54.8 |

*End of Table 5*

| # | Variable | Importance |
|---|---|---|
| 3 | Temperature_1 | 29.6 |
| 4 | Guiyang.attractions_4 | 24.6 |
| 5 | Guiyang.cuisine_1 | 24.6 |
| 6 | Guiyang.Hotel_6 | 18.1 |
| 7 | Guiyang.cuisine_6 | 14 |
| 8 | Guiyang.attractions_6 | 12.8 |
| 9 | Guiyang.tourism_5 | 5.66 |
| 10 | dTemperature | 0 |

Table 6 represents the model accuracy metrics.

Table 6. Assessment of model accuracy (bagged MARS)

| Period | Before (80%) | Before (20%) | During | After |
|---|---|---|---|---|
| Forecast bias | –360 | –13,355 | 7,076 | –20,444 |
| $R^2$ | 0.58 | –4.92 | –0.07 | –0.36 |

Despite quite advanced mathematics built in this algorithm, the behavior of this model needs to be revised. The coefficient of determination is negative, meaning that the model is worse than the simple average of values.

### 5.3. XGBoost

An XGBoost model was created using the *xgb.train* function from the *xgboost* library. In this function, two parameters can be used in optimization: *max.depth* (maximum depth of a tree) and *nrounds* (max number of boosting iterations). Figure 17 represents the results of optimizing the number of boosting operations vs. the model RMSE. The RMSE value is stabilized at approximately 20 for both lines.

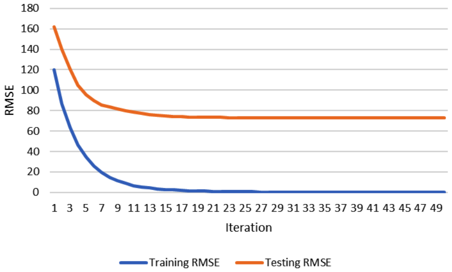


Figure 17. Training and testing RMSE

Assessment of model accuracy and TOP-10 of variable importance are shown in Table 7 and Table 8, respectively.

Table 7. Assessment of model accuracy (bagged MARS)

| Period | Before (80%) | Before (20%) | During | After |
|---|---|---|---|---|
| Forecast bias | –806 | –5,806 | 20 | –16,276 |
| $R^2$ | 0.99 | –0.21 | 0.49 | 0.33 |

The values of importance were provided by the *xgb.importance* function from the *xgboost* library.

Table 8. TOP-10 of variable importance (XGBoost)

| # | Variable | Importance |
|---|---|---|
| 1 | Guiyang.cuisine | 0.01 |
| 2 | Guiyang.attractions | 0.01 |
| 3 | dTemperature | 0.01 |
| 4 | Guiyang.tourism_1 | 0.007 |
| 5 | Guiyang.air.ticket_4 | 0.006 |
| 6 | Guiyang.air.ticket_2 | 0.006 |
| 7 | dGuiyang.Hotel | 0.005 |
| 8 | Baidu.index | 0.005 |
| 9 | Guiyang.air.ticket_7 | 0.005 |
| 10 | dGuiyang.cuisine | 0.005 |

The XGBoost algorithm showed a perfect result on the training data ($R^2$ = 99%), but on the training data the result became unexpectedly negative: the coefficient of determination showed a negative value. However, on new data from the during-COVID-19 and the after-COVID-19 time periods, the model showed positive values of $R^2$, but their values were less than desirable level of 50%.

### 5.4. Random forest

The random forest algorithm requires defining the number of trees. To solve this problem, the relationship between model quality expressed through the mean squared residuals (MSR) and the number of trees was analyzed (see Figure 18). Based on the variance explained and the behavior of MSR, it was decided to set the number of trees equaled 60.

Assessing the model accuracy metrics it was found that the model has the same characteristics as the previous ones (see Table 9). Forecast bias shows the same problems with underestimation (except the during-COVID-19 time period), and the coefficient of determination ($R^2$) is about zero.

Table 9. Assessment of model accuracy (Random Forest)

| Period | Before (80%) | Before (20%) | During | After |
|---|---|---|---|---|
| Forecast bias | –285 | –4,069 | 3,788 | –12,105 |
| $R^2$ | 0.95 | 0.28 | 0.37 | 0.38 |

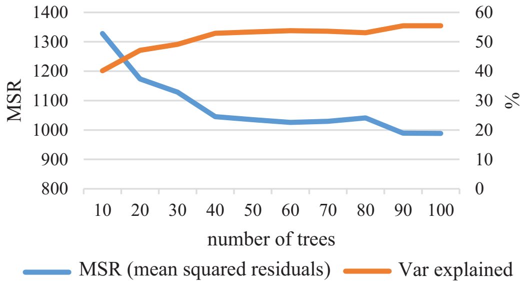


Figure 18. MSR vs. number of trees

Table 10 displays the TOP-10 of variable importance. The values of importance were extracted by the *importance* function from the *randomForest* library.

Table 10. TOP-10 of variable importance (Random Forest)

| # | Variable | Importance |
|---|---|---|
| 1 | Guiyang.Hotel_1 | 4.11 |
| 2 | Guiyang.attractions | 3.90 |
| 3 | Guiyang.cuisine_1 | 3.81 |
| 4 | Baidu.index_1 | 3.75 |
| 5 | Baidu.index | 3.59 |
| 6 | Baidu.index_2 | 3.51 |
| 7 | Guiyang.cuisine | 3.48 |
| 8 | Guiyang.cuisine_5 | 3.39 |
| 9 | Temperature_1 | 3.35 |
| 10 | Guiyang.Hotel | 3.17 |

Summarizing the results produced by the random forest model, it can be concluded that the model cannot predict hotel occupancy appropriately despite significant results obtained from the training data. All $R^2$ under 50% were considered insufficient.

### 5.5. SVM

Implementing a support vector machine technique on the training data, the coefficient of determination ($R^2$) was sufficiently high (0.81). Nevertheless, extending the model on the testing data in each separate time period, these metrics dramatically decreased up to zero (see Table 11), signalizing the inability of the model to forecast hotel occupancy at an appropriate level. In the next two periods, the determination coefficient remained under 50%.

Table 11. Assessment of model accuracy (SVM)

| Period | Before (80%) | Before (20%) | During | After |
|---|---|---|---|---|
| Forecast bias | –225 | –4,429 | 4,470 | –8,093 |
| $R^2$ | 0.81 | 0.01 | 0.36 | 0.48 |

Table 12 displays the TOP-10 of variable importance. The values of importance were calculated by the *Importance* function from the *rminer* library.

Table 12. TOP-10 of variable importance (SVM)

| # | Variable | Importance |
|---|---|---|
| 1 | dGuiyang.attractions | 0.03 |
| 2 | Guiyang.cuisine_6 | 0.02 |
| 3 | Guiyang.cuisine_5 | 0.02 |
| 4 | Weekend | 0.02 |
| 5 | Guiyang.cuisine_7 | 0.02 |
| 6 | Guiyang.cuisine_4 | 0.02 |
| 7 | dGuiyang.Hotel | 0.02 |
| 8 | dTemperature | 0.02 |
| 9 | Guiyang.tourism_6 | 0.02 |
| 10 | Guiyang.tourism_5 | 0.02 |

### 5.6. Benchmark

The following results were obtained by applying the benchmark model to different time periods (see Table 13). In the case of forecast bias, the picture is relatively typical compared with other models: the model tends to underestimate in all time periods except the during-COVID-19 time period.

Table 13. Assessment of model accuracy (benchmark)

| Period | Before (80%) | Before (20%) | During | After |
|---|---|---|---|---|
| Forecast bias | –360 | –842 | 1,227 | –615 |
| $R^2$ | 0.69 | 0.53 | 0.84 | 0.76 |

In the context of the coefficient of determination, the model shows relatively stable results. Based on the training data, the model was able to explain 69% of the initial variance. On the left 20% of the before-COVID-19 time period, the model showed 53% that can be considered appropriate. Notably, in the during-COVID-19 time period the model improved its forecasting power up to $R^2$ = 84%. After the coefficient fell to 76%.

## Conclusions

Considering all results obtained during the research, it can be concluded that:

1. The behavior of all models, including the benchmark, was quite typical: on the testing data the models showed negative bias signalizing the underestimation of hotel occupancy. Mostly all models had forecast bias equaled approximately 300 (except XGBoost where forecast bias = –806). The feature of underestimation remained on

the testing data, showing a significant difference between actual and forecasted data (forecast bias was expressed in negative thousands). Nevertheless, in the during-COVID-19 time period overestimation appeared. This effect can be explained by the significant decrease in hotel occupancy caused by external factors (sufficient restrictions and limitations realized by the Chinese government during the COVID-19 pandemic). When the situation with COVID-19 stabilized, the models again showed more serious negative forecast bias. Thus, bagged CART, bagged MARS, random forest, XGBoost, and SVM tend to underestimate hotel occupancy in a daily format. The benchmark model (ARDL model) increased forecast bias insignificantly compared to other models.

2. The forecasting power (accuracy) of the machine learning algorithms used in this research can be estimated as weak because all coefficients of determination are less than 50% on the testing data (left 20% of the before-COVID-19 time period, during- and after-) except the bagged CART model that could explained 53%, 54%, and 68% of initial variance, respectively. The bagged CART model was the only one that could ensure the $R^2$ coefficient greater than 50%. However, the benchmark model based on the ARDL model class showed better results: 53%, 84%, and 76%, respectively.
3. In the context of variable importance, it was found that the Baidu search index and its components related to hotel booking and visits to the corresponding city can be used to build a model to predict hotel occupancy. Summarizing the frequencies of how often the corresponding variable was used the TOP-10 of them looks as follows:
   - Guiyang.cuisine
   - Baidu.index
   - Guiyang.attractions
   - Guiyang.air.ticket
   - Guiyang.Hotel
   - Guiyang.tourism
   - Temperature
   - dGuiyang.Hotel
   - dTemperature
   - dGuiyang.cuisine
4. The weak forecasting power of the models (excluding bagged CART) shows that the task of regression (not classification) is more typical for the machine learning algorithms used in this research. Nevertheless, at least one algorithm – bagged CART – was able to predict more or less adequately high frequent daily hotel occupancy.

Summarizing all above, the following conclusions on two hypotheses can be done:

1. In this case, the machine learning tools used in this research did not show more effective results than a traditional model based on the ARDL methodology. Thus, the first hypothesis on the higher effectiveness of machine learning tools compared to the traditional forecasting model based on the ARDL class of models is rejected.
2. As the variables' importance showed, the Chinese Baidu search engine index and its components were used in the machine learning models. Thus, the second hypothesis on the possible usage of the Baidu search engines values and its components is accepted.

Despite the weak results of the application of five machine learning algorithms for the task of regression the forecasting power could be improved due to several reasons:

1. All models have a list of different hyperparameters that could be tuned more specifically for the regression task.
2. All models could be tested with retraining each week or any other periodicity.
3. The number of observations could be increased to cover all cycles of hotel occupancy.
4. In addition, other machine learning tools can be chosen instead of the methods used in the research.

## Acknowledgements

This work has been supported by the European Regional Development Fund within the Activity 1.1.1.2 "Post-doctoral Research Aid" of the Specific Aid Objective 1.1.1 "To increase the research and innovative capacity of scientific institutions of Latvia and the ability to attract external financing, investing in human resources and infrastructure" of the Operational Programme "Growth and Employment" (No. 1.1.1.2/VIAA/3/19/479).

## References

Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A., & Eshun, J. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*, *6*, 100163. https://doi.org/10.1016/j.dajour.2023.100163

Ahani, A., Nilashi, M., Ibrahim, O., Sanzogni, L., & Weaven, S. (2019). Market segmentation and travel choice prediction in Spa hotels through TripAdvisor's online reviews. *International Journal of Hospitality Management*, *80*, 52–77. https://doi.org/10.1016/j.ijhm.2019.01.003

Al Shehhi, M., & Karathanasopoulos, A. (2020). Forecasting hotel room prices in selected GCC cities using deep learning. *Journal of Hospitality and Tourism Management*, *42*, 40–50. https://doi.org/10.1016/j.jhtm.2019.11.003

Aryai, V., & Glodsworthy, M. (2023). Day ahead carbon emission forecasting of regional National Electricity Market using machine learning methods. *Engneering Application of Artificial Intelligence*, *123*, 106314. https://doi.org/10.1016/j.engappai.2023.106314

Boriratrit, S., Fuangfoo, P., Srithapon, C., & Chatthaworn, R. (2023). Adaptive meta-learning extreme learning machine with golden eagle optimization and logistic map for forecasting the incomplete data of solar irradiance. *Energy and AI*, *13*, 100243. https://doi.org/10.1016/j.egyai.2023.100243

Breiman, L. (1984). *Classification and regression trees* (1st ed.). Routledge. https://doi.org/10.1201/9781315139470

Buja, A., & Stuetzle, W. (2006). Observations on bagging. *Statistica Sinica*, *16*(2), 323–351. http://www.jstor.org/stable/24307547

Caicedo-Torres, W., & Payares, F. (2016). A machine learning model for occupancy rates and demand forecasting in the hospitality industry. In M. Montes y Gómez, H. Escalante, A. Segura, & J. Murillo (Eds.), *Lecture notes in computer science: Vol. 10022. Advances in Artificial Intelligence – IBERAMIA 2016* (pp. 201–211). Springer. https://doi.org/10.1007/978-3-319-47955-2_17

Calero-Sanz, J., Orea-Giner, A., Villacé-Molinero, T., Muñoz-Mazón, A., & Fuentes-Moraleda, L. (2022). Predicting a new hotel rating system by analysing UGC content from Tripadvisor: Machine learning application to analyse service robots influence, *Procedia Computer Science*, *200*, 1078–1083. https://doi.org/10.1016/j.procs.2022.01.307

Chen, T., & He, T. (2023). *xgboost: eXtreme Gradient Boosting.* R package version 1.7.5.1. https://cran.r-project.org/web/packages/xgboost/vignettes/xgboost.pdf

Divasón, J., Ceniceros, J. F., Sanz-Garcia, A., Pernia-Espinoza, A., & Martinez-de-Pison, F. J. (2023). PSO-PARSIMONY: A method for finding parsimonious and accurate machine learning models with particle swarm optimization. Application for predicting force-displacement curves in T-stub steel connections. *Neurocomputing*, *548*, 126414. https://doi.org/10.1016/j.neucom.2023.126414

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2), 337–407. https://doi.org/10.1214/aos/1016218223

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, *19*(1), 1–67. https://doi.org/10.1214/aos/1176347963

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Gong, Y., Liu, G., Xue, Y., Li, R., & Meng, L. (2023). A survey on dataset quality on machine learning. *Information and Software Technology*, *162*, 107268. https://doi.org/10.1016/j.infsof.2023.107268

Huang, L., & Zheng, W. (2023). Novel deep learning approach for forecasting daily hotel demand with agglomeration effect. *International Journal of Hospitality Management*, *98*, 103038. https://doi.org/10.1016/j.ijhm.2021.103038

Jiang, Y., Tran, T. H., & Williams, L. (2023). Machine learning and mixed reality for smart aviation: Applications and challenges. *Journal of Air Transport Management*, *111*, 102437. https://doi.org/10.1016/j.jairtraman.2023.102437

Kamm, S., Veekati, S. S., Müller, T., Jazdi, N., & Weyrich, M. (2023). A survey on machine learning based analysis of heterogeneous data in industrial automation. *Computers in Industry*, *149*, 103930. https://doi.org/10.1016/j.compind.2023.103930

Kaya, K., Yılmaz, Y., Yaslan, Y., Öğüdücü, S. G., & Çıngı, F. (2022). Demand forecasting model using hotel clustering findings for hospitality industry. *Information Processing and Management*, *59*(1), 102816. https://doi.org/10.1016/j.ipm.2021.102816

Khalil, M., McGough, A. S., Pourmirza, Z., Pazhoohesh, M., & Walker, S. (2022). Machine Learning, Deep Learning and Statistical Analysis for forecasting building energy consumption – A systematic review. *Engineering Applications of Artificial Intelligence*, *115*, 105287. https://doi.org/10.1016/j.engappai.2022.105287

Kim, H. S. (2010). hotel property characteristics and occupancy rate: Examining super deluxe 1st class hotels in Seoul, Korea. *International Journal of Tourism Sciences*, *10*(3), 25–47. https://doi.org/10.1080/15980634.2010.11434630

Kolomoyets, Y., & Dickinger, A. (2023). Understanding value perceptions and propositions: A machine learning approach. *Journal of Business Research*, *154*, 113355. https://doi.org/10.1016/j.jbusres.2022.113355

Koupriouchina, L., van der Rest, J. P., & Schwartz, A. (2014). On revenue management and the use of occupancy forecasting error measures. *International Journal of Hospitality Management*, *41*, 104–114. https://doi.org/10.1016/j.ijhm.2014.05.002

Li, X., Li, H., Pan, B., & Law, R. (2020). Machine learning in internet search query selection for tourism forecasting. *Journal of Travel Research*, *60*(6), 1213–1231. https://doi.org/10.1177/0047287520934871

Lim, C. (1997). Review of international tourism demand models. *Annals of Tourism Research*, *24*(4), 835–849. https://doi.org/10.1016/S0160-7383(97)00049-2

Mehmood, F., Ghani, M. U., Ghafoor, H., Shahzadi, R., Asim, M. N., & Mahmood, W. (2022). EGD-SNet: A computational search engine for predicting an end-to-end machine learning pipeline for Energy Generation & Demand Forecasting. *Applied Energy*, *324*, 119754. https://doi.org/10.1016/j.apenergy.2022.119754

Prajwala, T. R. (2015). A comparative study on decision tree and random forest using R tool. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(1), 196–199. https://doi.org/10.17148/IJARCCE.2015.4142

Qin, Q., Huang, Z., Zhou, Z., Chen, C., & Liu, R. (2023). Crude oil price forecasting with machine learning and Google search data: An accuracy comparison of single-model versus multiple-model. *Engineering Applications of Artificial Intelligence*, *123*, 106266. https://doi.org/10.1016/j.engappai.2023.106266

Sánchez, E. C., Sánchez-Medina, A. J., & Pellejero, M. (2020). Identifying critical hotel cancellations using artificial intelligence. *Tourism Management Perspectives*, *35*, 100718. https://doi.org/10.1016/j.tmp.2020.100718

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

Sayed, Y. A. K., Ibrahim, A. A., Tamrazyan, A. G., & Fahmy, M. F. M. (2023). Machine-learning-based models versus design-oriented models for predicting the axial compressive load of FRP-confined rectangular RC columns. *Engineering Structures*, *285*, 116030. https://doi.org/10.1016/j.engstruct.2023.116030

Strielkowski, W., Vlasov, A., Selivanov, K., Muraviev, K., & Shakhnov, V. (2023). Prospects and challenges of the machine learning and data-driven methods for the predictive analysis of power systems: A review. *Energies*, *16*(10), 4025. https://doi.org/10.3390/en16104025

Sun, C., & Lu, J. (2023). The relative roles of different land-use types in bike-sharing demand: A machine learning-based multiple interpolation fusion method. *Information Fusion*, *95*, 384–400. https://doi.org/10.1016/j.inffus.2023.02.033

Sun, J., Dang, W., Wang, F., Nie, H., Wei, X., Li, P., Zhang, S., Feng, Y., & Li, F. (2023). Prediction of TOC content in organic-rich shale using machine learning algorithms: Comparative study of random forest, support vector machine, and XGBoost. *Energies*, *16*(10), 4159. https://doi.org/10.3390/en16104159

van Eck, N. J., & Waltman, L. (2023). *VOSviewer manual*. https://www.vosviewer.com/documentation/Manual_VOSviewer_1.6.19.pdf

Viverit, L., Heo, C. Y., Pereira, L. N., & Tiana, G. (2023). Application of machine learning to cluster hotel booking curves for hotel demand forecasting. *International Journal of Hospitality Management*, *111*, 103455. https://doi.org/10.1016/j.ijhm.2023.103455

Yang, Y., Pan, B., & Song, H. (2014). Predicting hotel demand using destination marketing organization's web traffic data. *Journal of Travel Research*, *53*(4), 433–447. https://doi.org/10.1177/0047287513500391

Yang, Y., Tang, J., Luo, H., & Law, R. (2015). Hotel location evaluation: A combination of machine learning tools and web GIS. *International Journal of Hospitality Management*, *47*, 14–24. https://doi.org/10.1016/j.ijhm.2015.02.008

Zhai, Q., Tian, Y., Luo, J., & Zhou, J. (2023). Hotel overbooking based on no-show probability forecasts. *Computers & Industrial Engineering*, *180*, 109226. https://doi.org/10.1016/j.cie.2023.109226