
UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Master's Degree in Artificial Intelligence and Data Engineering

# Predictive Modeling of Hotel Booking Cancellations: A Machine Learning Approach

Author:

**Leonardo Ceccarelli**

Project GitHub Repository:

https://github.com/cecco01/DMML_Project

ACADEMIC YEAR 2025/2026

# Contents

# Chapter 1

# Introduction

The global hospitality industry, valued at over $700 billion in 2023, is expected to grow at a Compound Annual Growth Rate (CAGR) of 5.4% through 2027. This growth is driven by increasing travel demand, global tourism, and the rise of digital booking platforms such as Expedia, Booking.com, and Airbnb.
While digital transformation has significantly streamlined hotel operations, it has also introduced new challenges, particularly in handling customer cancellations. Cancellations not only disrupt revenue forecasting but also complicate inventory management and guest satisfaction, impacting both luxury and independent hotels.

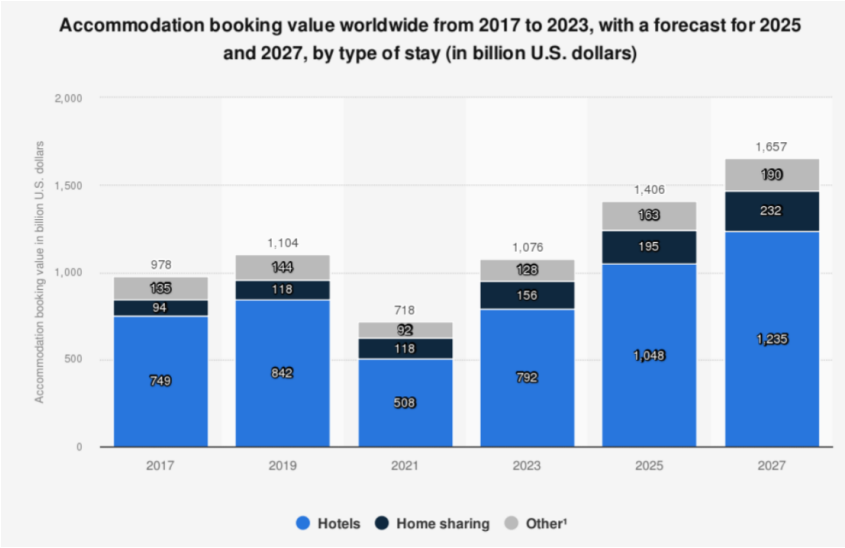


Figure 1.1: Global Accommodation Booking Value by Type. Source: Statista.

Cancellations result in substantial financial losses and operational difficulties. They impair demand forecasting, lead to overbooking, and complicate inventory management. Effective management of cancellations, especially during peak seasons, is essential to optimize room pricing and resource allocation.
Machine learning provides a powerful tool to predict which bookings are likely to be canceled based on historical data. This allows hotels to take proactive steps, such as offering targeted incentives or personalized communication, to reduce cancellations and improve inventory management. Predictive models also help hotels optimize revenue strategies without relying solely on overbooking or stringent cancellation policies.
The objective of this research is to develop a machine learning model that accurately pre-

dicts hotel booking cancellations. Using data mining and machine learning techniques, this project identifies the key factors driving cancellations and constructs a pipeline to mitigate them. The study encompasses exploratory data analysis (EDA), data preprocessing, feature engineering, model selection, and evaluation, delivering actionable insights and practical solutions for the hospitality industry.

# Chapter 2

# Dataset

This project utilizes two hotel booking datasets, each containing records from distinct types of hotels: a resort hotel (H1) and a city hotel (H2). The data spans a period between July 2015 and August 2017, encompassing all bookings made during this time frame, whether they resulted in cancellations or actual stays. The dataset was originally collected as part of a property management system (PMS).

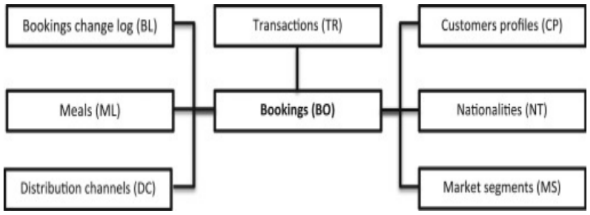


Figure 2.1: PMS database structure

## 2.1 Data Overview

The datasets contain 31 variables, which provide comprehensive information about the booking process, including details on the reservation status, customer demographics, and room allocation. Given the scarcity of real-world data in the hotel industry for research purposes, these datasets serve as valuable resources for studies in revenue management, machine learning, and customer segmentation. The datasets offer a rich set of features that are crucial in understanding customer behavior, booking patterns, and the factors contributing to cancellations.

## 2.2 Data Distribution

The two hotels exhibit slightly different characteristics in terms of booking volume and cancellation rates:

- **H1**: Resort hotel with 40,060 bookings, where 27.8% of the bookings were canceled.
- **H2**: City hotel with 79,330 bookings, where the cancellation rate was higher at 41.7%.

The datasets reflect a diversity of customer types and booking behaviors, ranging from direct bookings to third-party travel agencies and group bookings. Both datasets also

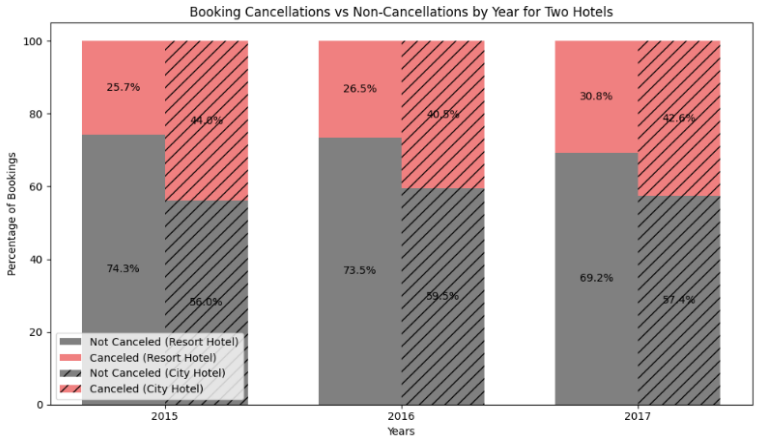


Figure 2.2: Cancellations distribution over the years for H1 and H2

capture key hotel operational data, such as room type assignment, meal plans, and booking modifications, which provide valuable insights into customer preferences and the operational decisions that might influence booking cancellations.

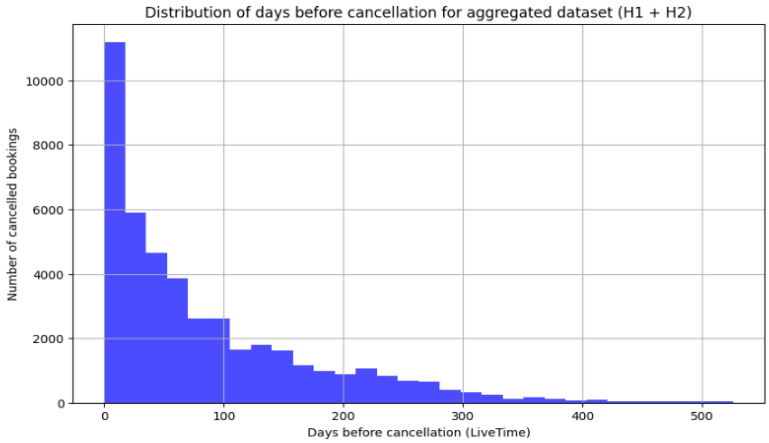


Figure 2.3: Distribution of days before cancellations for both hotels

## 2.3 Dataset Convenience Splitting

To ensure the model generalizes well to new data, a convenience split was used instead of a random or stratified split. This method accounts for changing patterns over time and reduces bias or data leakage. The data was ordered by Arrival Date and split into monthly month-year blocks. Within each block, a 75%-25% stratified split was applied to preserve the distribution of the target variable, IsCanceled.

This time-ordered split addresses non-stationary temporal data, allowing the model to capture evolving booking and cancellation patterns over time. It ensures the model is trained on past data and evaluated on unseen data, providing a more realistic performance assessment.

Despite using a time-ordered split, the problem remains a classification problem, not time-series forecasting. The goal is to predict whether a booking will be canceled based on available features at the time of booking, without considering sequential dependencies. The time-ordered split helps prevent data leakage but does not imply temporal

predictions. All time-related features were removed before training to maintain this independence.

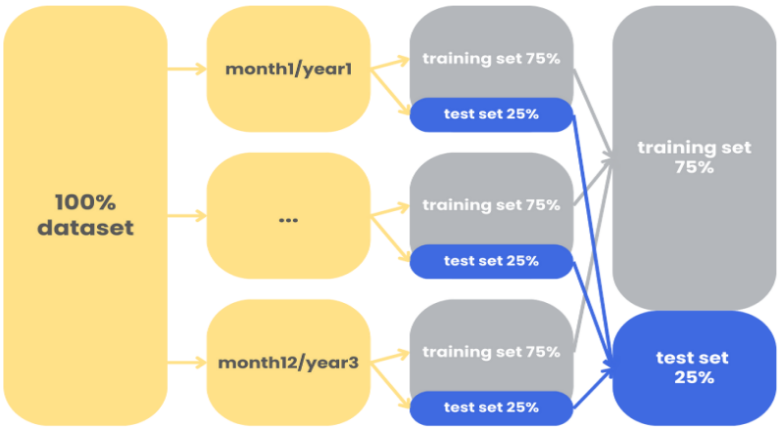


Figure 2.4: Dataset Convenient Splitting

# Chapter 3

# Classification

## 3.1 Preprocessing Phase

To ensure the dataset was suitable for machine learning algorithms, several preprocessing steps were applied. This involved handling missing data, transforming categorical features into a numerical format, scaling numerical variables, and engineering new features to better capture the relationships in the data.

## 3.2 Feature Engineering

In addition to the raw features present in the dataset, an engineered feature was created to enhance the model's performance: ADRThirdQuartileDeviation. This feature was created to account for the variability in hotel pricing. The original ADR (Average Daily Rate) feature only captured the daily price of a room, which could vary significantly based on factors like room type, seasonality, and booking channel. To normalize this, the deviation of each booking's ADR was calculated from the third quartile of ADR values within the same distribution channel, room type, and time period (week/year pair). This transformation ensures that ADR is scaled relative to its peer group, allowing the model to capture the effect of "overpriced" or "underpriced" bookings more effectively. Higher deviations from the third quartile suggest that a booking might be overpriced, which correlates with a higher probability of cancellation.

$$ADRThirdQuartileDeviation = \frac{ADR_i}{Q3_{ADR}(channel, roomtype, week/year)}$$

Where:

- $\mathbf{ADR}_i$: The Average Daily Rate of booking i
- $\mathbf{Q3}_{ADR}$(channel, room type, week/year): The third quartile of ADR values for bookings within the same distribution channel, room type, and time period (week/year pair)

## 3.3 Encoding Categorical Variables

Many machine learning algorithms require categorical variables to be converted into numerical representations. For this dataset, the following encoding techniques were used:

- **One-Hot Encoding:** Categorical variables with a limited number of unique values were transformed using one-hot encoding. This ensures that no ordinal relationship is assumed between categories.

- **Logit-Odds Encoding:** For categorical features with high cardinality, such as Agent and Company, a Logit-Odds encoding was applied to mitigate the overfitting risk and computational inefficiency caused by high cardinality. This method transforms categorical values into two numerical features:
  - **Logit-Odds**: The logit-odds of cancellation for each category, calculated based on the observed likelihood of cancellation. Smoothing was applied to prevent extreme values.
  - **Prevalence Flag**: A binary indicator that flags whether the category level is common, based on a minimum frequency threshold of 2%. Categories meeting the threshold are marked as prevalent (1), while rare levels are marked as non-prevalent (0).

## 3.4 Pipeline Building

The classification model was built using a pipeline to ensure consistent application of preprocessing, feature engineering, and model training. The pipeline consists of the following steps:

1. **Feature Engineering**: The `BookingFeaturesTransformer` computes the engineered feature `ADRThirdQuartileDeviation`, which normalizes the booking price relative to its peer group, capturing pricing variability.
2. **Preprocessing**: Numerical features are standardized using `StandardScaler`, while categorical features are handled using One-Hot Encoding and Logit-Odds Encoding with prevalence flags for high-cardinality features.
3. **Handling Imbalanced Data**: SMOTE (Synthetic Minority Over-sampling Technique) addresses class imbalance by generating synthetic samples for the minority class (cancellations).
4. **Model Integration**: The final step integrates the machine learning model, ensuring all preprocessing steps are applied consistently to both the training and test datasets.

## 3.5 Models Comparison

To evaluate the performance of various machine learning models for predicting hotel booking cancellations, several models were cross-validated using a 5-fold StratifiedKFold. The models tested included:

- Random Forest
- Logistic Regression
- AdaBoost
- Decision Tree
- K-Nearest Neighbors
- Bagging
- Naive Bayesian
- XGBoost

### 3.5.1 Results Summary

The table below presents the performance of each model across the different metrics:

| Model | Accuracy | Precision | Recall | F1 Score | ROC AUC |
|---|---|---|---|---|---|
| Random Forest | 0.8457 | 0.8095 | 0.7635 | 0.7858 | 0.9110 |
| Logistic Regression | 0.7855 | 0.7153 | 0.7001 | 0.7076 | 0.8397 |
| AdaBoost | 0.7924 | 0.7262 | 0.7063 | 0.7161 | 0.8607 |
| Decision Tree | 0.8022 | 0.7198 | 0.7639 | 0.7412 | 0.7962 |
| K-Nearest Neighbors | 0.7939 | 0.7073 | 0.7576 | 0.7316 | 0.8580 |
| Bagging | 0.8379 | 0.7982 | 0.7531 | 0.7750 | 0.8953 |
| Bayesian | 0.6578 | 0.5260 | 0.7765 | 0.6272 | 0.7787 |
| XGBoost | 0.8348 | 0.7958 | 0.7456 | 0.7699 | 0.9000 |

Table 3.1: Comparison of Model Performance Metrics

### 3.5.2 ROC Curves

The ROC curves, in the section below, compare the performance of all models. Random Forest achieved the highest ROC AUC of 0.91, followed by XGBoost and Bagging, both at 0.90.

### 3.5.3 Confusion Matrices

The confusion matrices provide a more detailed insight into the classification performance of each model, showing the distribution of true positive, true negative, false positive, and false negative predictions. Random Forest showed a balanced trade-off between false positives and false negatives, reinforcing its high overall performance.

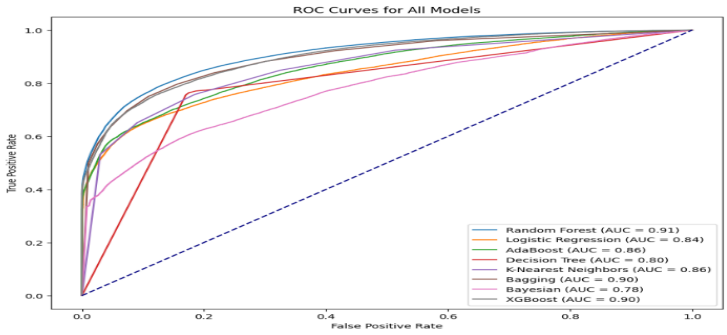


Figure 3.1: ROC Curves for All Models

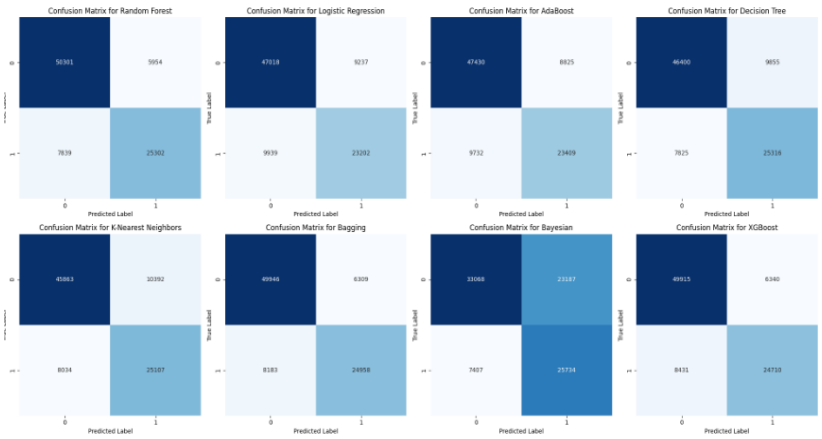


Figure 3.2: Confusion Matrices for All Models

### 3.5.4 Conclusion

Among the models tested, Random Forest performed the best overall, achieving the highest scores across all evaluation metrics, closely followed by XGBoost and Bagging. Naive Bayesian performed the worst, with a particularly low precision and F1 score, making it less suitable for this classification problem. The models with strong ROC AUC scores demonstrate better ability to differentiate between cancellations and non-cancellations, making them more reliable for practical applications in hotel booking systems.

## 3.6 Model Evaluation

### 3.6.1 Fine-tuning using GridSearchCV

To improve the performance of the Random Forest model, a hyperparameter tuning process was conducted using GridSearchCV with a 5-fold Stratified-KFold. The following hyperparameters were optimized:

- `n_estimators`: The number of trees in the forest.
- `max_depth`: The maximum depth of the tree.
- `max_features`: The number of features to consider when looking for the best split.

The grid search was performed with two scoring metrics: ROC AUC and Recall. The best hyperparameters were selected based on the highest recall score, as predicting cancellations (true positives) was a priority. The best parameters found were:

- `n_estimators`: 200
- `max_depth`: None
- `max_features`: sqrt

After fine-tuning, the best model achieved the following cross-validation results:

- Best Recall: 0.7655
- Best ROC AUC: 0.9119

The fine-tuned Random Forest model was then retrained on the entire training set and evaluated on the test set. The evaluation metrics for the test set are as follows:

| Model | Accuracy | Precision | Recall | F1 Score | ROC AUC |
|---|---|---|---|---|---|
| Random Forest | 0.8503 | 0.8150 | 0.7710 | 0.7924 | 0.8340 |

Table 3.2: Evaluation metrics for the fine-tuned Random Forest model

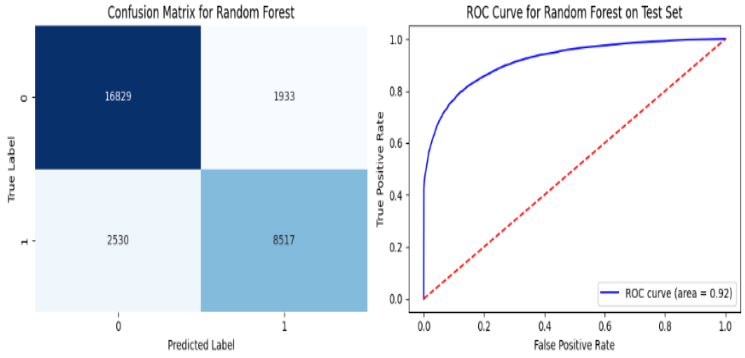


Figure 3.3: Confusion Matrix and ROC Curve for the fine-tuned Random Forest model

### 3.6.2 Feature Importance

To gain insight into the most influential features for predicting booking cancellations, the feature importances of the fine-tuned Random Forest model were extracted. The following table lists the top 10 features contributing to the model's decisions in descending order:

| Feature | Importance |
|---|---|
| LeadTime | 0.216913 |
| ADRThirdQuartileDeviation | 0.172019 |
| TotalOfSpecialRequests | 0.071267 |
| DepositType Non Refund | 0.068034 |
| DepositType No Deposit | 0.065103 |
| StaysInWeekNights | 0.060497 |
| Agent logit odds | 0.056752 |
| PreviousCancellations | 0.035995 |
| StaysInWeekendNights | 0.032009 |
| BookingChanges | 0.027160 |
| Adults | 0.019519 |

Table 3.3: Top 10 Important Features for the fine-tuned Random Forest Model

# Chapter 4

# Graphical User Interface

To facilitate the use of the cancellation prediction model, a graphical user interface (GUI) was developed. This interface allows users to input booking details and receive real-time predictions on whether a booking is likely to be canceled. The GUI is intuitive and easy to use, ensuring that users with little technical knowledge can operate it effectively.
The main interface provides users with input fields to enter booking details such as Lead Time, Arrival Date, Meal Plan, Deposit Type, and more. After filling in the required information, the user can click the "Predict" button to receive the model's prediction. After the user inputs the booking details and clicks the "Predict" button, the system generates the prediction and displays whether the booking is likely to be canceled.

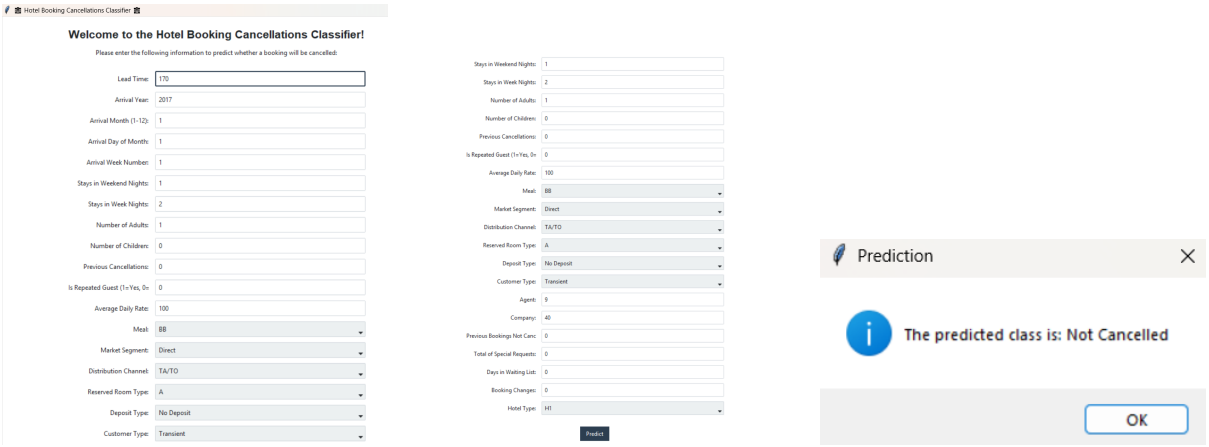


Figure 4.1: Screenshots of the Hotel Booking Cancellation Classifier GUI

# Chapter 5

# Conclusions

This study implemented and evaluated various machine learning models to predict hotel booking cancellations using a comprehensive reservation dataset. The project focused on preprocessing, feature engineering to enhance predictive accuracy, identifying key features influencing cancellations, and fine-tuning top models for improved generalization.
After performing comprehensive preprocessing and feature engineering, including the creation of the ADRThirdQuartileDeviation feature and encoding high-cardinality categorical variables using Logit-Odds Encoding, we applied multiple machine learning algorithms, including Random Forest, XGBoost, Logistic Regression, AdaBoost, and others. The evaluation revealed that the Random Forest model outperformed all other models, achieving the highest recall and ROC AUC scores, making it the most suitable model for predicting cancellations. Hyperparameter tuning via GridSearchCV further improved the Random Forest's performance, leading to a final recall score of 0.77 and a ROC AUC of 0.91 on cross-validation and recall score of 0.77 and ROC AUC of 0.84 on test set. The most important features identified by the model were:

- LeadTime: The number of days between booking and arrival, which is the strongest predictor of cancellations. Longer lead times are associated with higher cancellation rates.
- ADRThirdQuartileDeviation: The deviation of the average daily rate (ADR) from the third quartile, reflecting the influence of pricing on cancellations. Bookings with higher deviations from typical prices are more likely to be canceled.
- TotalOfSpecialRequests: The total number of special requests made by the customer, which can indicate a more engaged customer but also correlate with a higher likelihood of cancellations.
- DepositType: Both non-refundable and no-deposit bookings have a significant impact on cancellations.
- StaysInWeekNights: The number of weekday nights in the booking. Bookings with more weekday nights show different cancellation behaviors, potentially due to varying customer preferences or purposes of travel (business vs. leisure).

These insights suggest that effective cancellation prediction can help hotels make better decisions about overbooking, pricing strategies, and customer segmentation. For example, targeted interventions could be designed for bookings with a high probability of

cancellation based on these features, helping hotels reduce the financial impact of cancellations.

# Chapter 6

# References


- N. Antonio, A. de Almeida and L. Nunes, "Predicting Hotel Bookings Cancellation with a Machine Learning Classification Model, " 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 2017, pp. 1049-1054, doi: 10.1109/ICMLA.2017.00-11.
- Z. A. Andriawan et al., "Prediction of Hotel Booking Cancellation using CRISP-DM, " 2020 4th International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICICoS51170.2020.9299011.
- Nuno Antonio, Ana de Almeida, Luis Nunes, Hotel booking demand datasets, Data in Brief, Volume 22, 2019, Pages 41-49, ISSN 2352-3409, https://doi.org/10.1016/j.dib.2018.11.126