Hotel Booking Demand Analysis

Your Name

2025-06-01

Contents

1 Problem Statement 1

1.1 Objective Description . . . 1

1.2 Evaluation Metrics . . . 1

1.3 Baseline Model Performance and Target Metrics . . . 2

2 Dataset Description, Data Preprocessing, and Visualization 2

2.1 Dataset Description, Data Preprocessing, and Visualization . . . 2

3 Exploratory Visualization 3

3.1 Proportion of Hotel Types . . . 3

3.2 Overall Booking Cancellation Proportions . . . 3

3.3 Resort Hotel: Cancellation vs. Non-Cancellation . . . 5

3.4 City Hotel: Cancellation vs. Non-Cancellation . . . 6

3.5 Reservation Counts per Year . . . 8

3.6 Monthly Average Guest Count per Hotel . . . 8

4 Analysis Process Designed for the Problem 9

4.1 Selected Data Mining Methods and Rationale . . . 9

4.2 Model Evaluation Method . . . 10

4.3 Platforms and Tools Used . . . 10

4.4 Research Workflow and Explanation . . . 11

1 Problem Statement

1.1 Objective Description

This study aims to predict whether a hotel booking will be canceled based on customer reservation information. The input data includes detailed booking attributes such as check-in date, number of guests, and length of stay. The research process consists of the following steps: data preprocessing, selection of important features, dataset splitting, model construction, and model performance evaluation. The final output is a prediction indicating whether a customer will cancel their booking.

1.2 Evaluation Metrics

In this study, booking cancellation is defined as the positive class (Positive), while non-cancellation is defined as the negative class (Negative). Based on this classification, the confusion matrix shown in Table 1 illustrates the relationship between actual and predicted outcomes. The evaluation metrics and their definitions are listed below:

- **Accuracy**: The overall correctness of predictions across all bookings

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision**: The proportion of correct “canceled” predictions among all predicted as “canceled”

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**: The model’s ability to correctly identify actual cancellations

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **AUC (Area Under the Curve)**: The area under the Receiver Operating Characteristic (ROC) curve

Table 1: Confusion Matrix for the Prediction Task

Actual.Class	Predicted..Is.Canceled	Predicted..Not.Canceled
Is Canceled	TP	FN
Not Canceled	FP	TN

1.3 Baseline Model Performance and Target Metrics

Table 2 presents the prediction performance reported in a reference study [@antonio2017cancellation] for both resort hotels (H1) and city hotels (H2). Based on these reported performances, we set the following target metrics:

- **Accuracy**: 0.85

- **Precision**: 0.85

- **Recall (Sensitivity)**: 0.80

- **AUC**: 0.90

Table 2: Model Performance from Reference Study

Hotel	Dataset	Acc.	AUC	Prec.	Sensit.
H1	Train	0.846	0.910	0.839	0.626
H1	Test	0.842	0.877	0.811	0.603
H2	Train	0.857	0.934	0.876	0.793
H2	Test	0.849	0.922	0.869	0.779

2 Dataset Description, Data Preprocessing, and Visualization

2.1 Dataset Description, Data Preprocessing, and Visualization

2.1.1 Loading Dataset

```
df <- read.csv("hotel_bookings.csv")
```

2.1.2 Data Source

- Kaggle: https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand
- Original publication[@antonio2019dataset]: https://www.sciencedirect.com/science/article/pii/S2352340918315191

2.1.3 Dataset Overview

- Total records: 119,390
- Time range: 2015 to 2017
- Features: 32 columns

Table 3: Table 1. Dataset Columns and Their Data Types

Columns	Data Type	Columns	Data Type
hotel	object (2)	is_repeated_guest	int64
is_canceled	int64	previous_cancellations	int64
lead_time	int64	previous_bookings_not_canceled	int64
arrival_date_year	int64	reserved_room_type	object (10)
arrival_date_month	object (12)	assigned_room_type	object (12)
arrival_date_week_number	int64	booking_changes	int64
arrival_date_day_of_month	int64	deposit_type	object (3)
stays_in_weekend_nights	int64	agent	float64
stays_in_week_nights	int64	company	float64
adults	int64	days_in_waiting_list	int64
children	float64	customer_type	object (4)
babies	int64	adr	float64
meal	object (5)	required_car_parking_spaces	int64
country	object (178)	total_of_special_requests	int64
market_segment	object (8)	reservation_status	object (3)
distribution_channel	object (5)	reservation_status_date	object

3 Exploratory Visualization

3.1 Proportion of Hotel Types

```r
hotel_prop <- df %>%
  group_by(hotel) %>%
  summarise(count = n()) %>%
  mutate(percent = count / sum(count),
         label = paste0(round(percent * 100, 3), "%"))

ggplot(hotel_prop, aes(x = "", y = percent, fill = hotel)) +
  geom_col(width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Proportion of Resort Hotel and City Hotel") +
  theme_void() +
  geom_text(aes(label = label), position = position_stack(vjust = 0.5), size = 4)
```

3.2 Overall Booking Cancellation Proportions

```r
cancel_prop <- df %>%
  mutate(status = ifelse(is_canceled == 1, "Canceled", "Not Canceled")) %>%
```

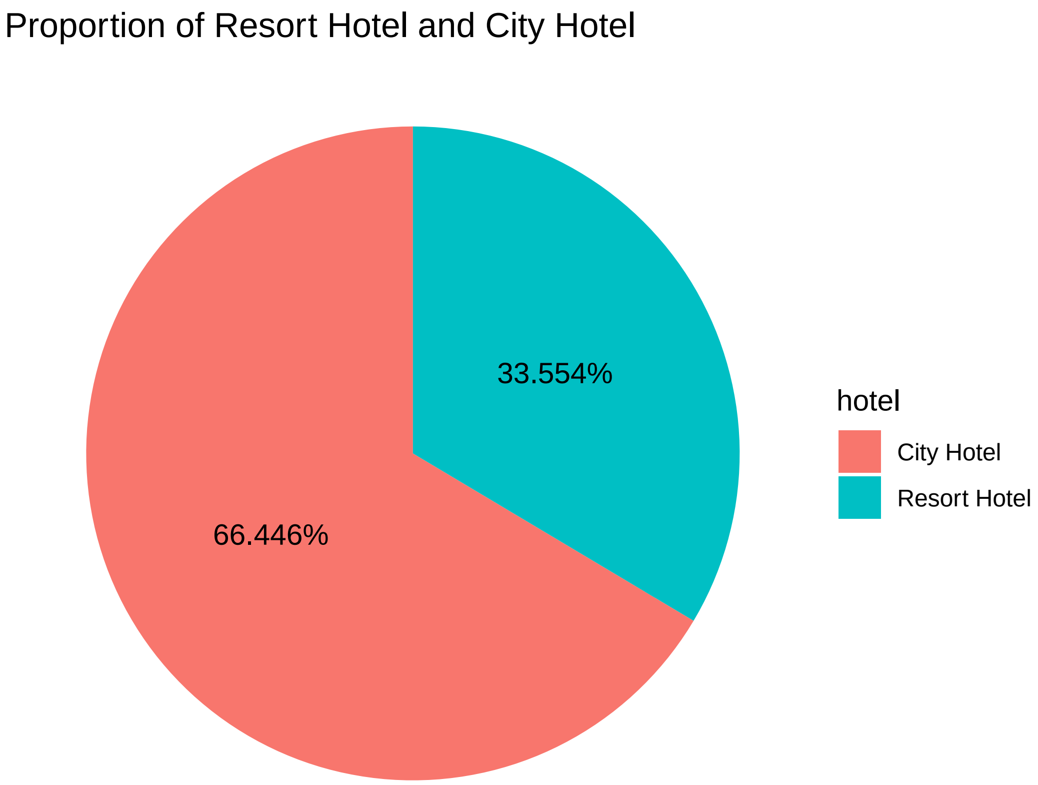

Figure 1: Figure 1. Proportion of Resort Hotel and City Hotel

```
  group_by(status) %>%
  summarise(count = n()) %>%
  mutate(percent = count / sum(count),
         label = paste0(round(percent * 100, 3), "%"))

ggplot(cancel_prop, aes(x = "", y = percent, fill = status)) +
  geom_col(width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Overall Cancellation and Non-Cancellation Proportions") +
  theme_void() +
  geom_text(aes(label = label), position = position_stack(vjust = 0.5), size = 4)
```

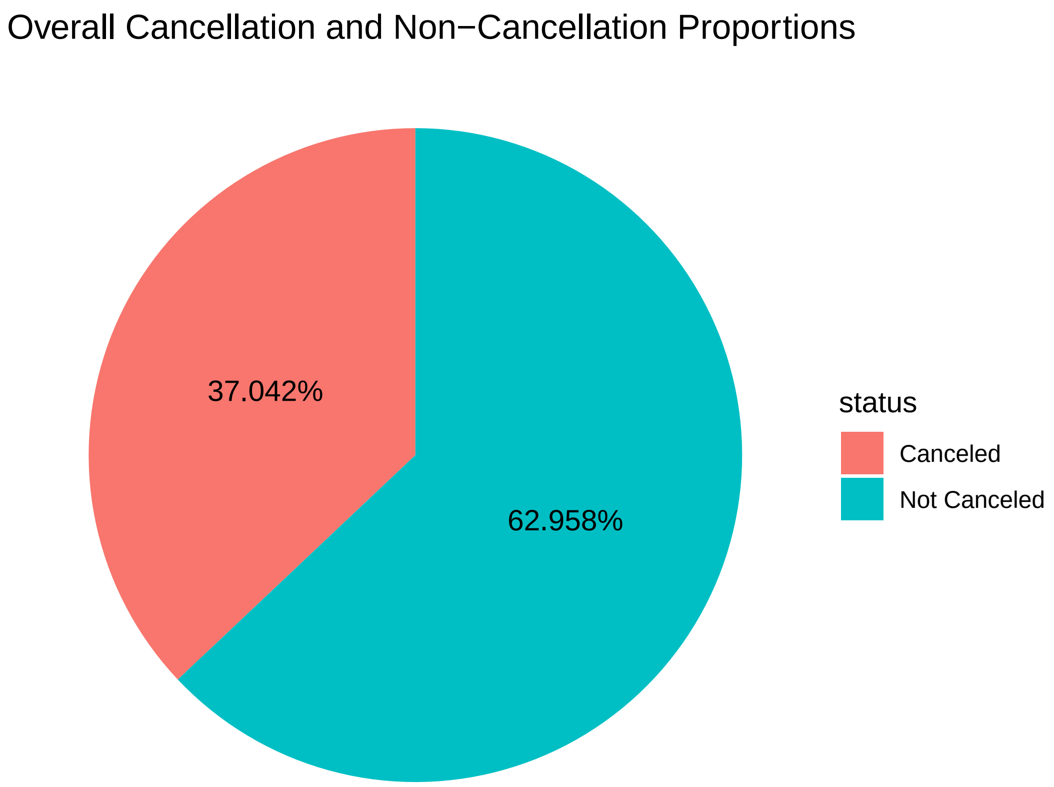

Figure 2: Figure 2. Overall Cancellation and Non-Cancellation Proportions

3.3 Resort Hotel: Cancellation vs. Non-Cancellation

```
resort_cancel <- df %>%
  filter(hotel == "Resort Hotel") %>%
  mutate(status = ifelse(is_canceled == 1, "Canceled", "Not Canceled")) %>%
  count(status) %>%
  mutate(percent = n / sum(n),
         label = paste0(round(percent * 100, 3), "%"))

ggplot(resort_cancel, aes(x = "", y = percent, fill = status)) +
  geom_col(width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Reservation Cancellation Status in Resort Hotel") +
  theme_void() +
  geom_text(aes(label = label), position = position_stack(vjust = 0.5), size = 4)
```

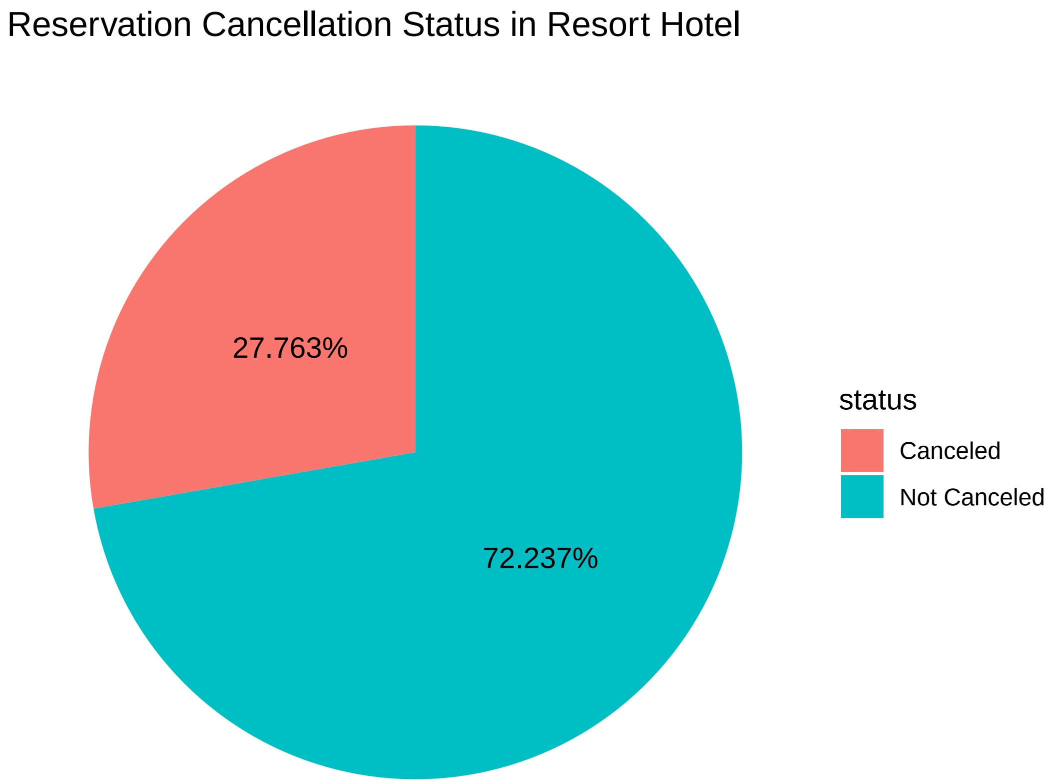

Figure 3: Figure 3. Reservation Cancellation Status in Resort Hotel

3.4 City Hotel: Cancellation vs. Non-Cancellation

```
city_cancel <- df %>%
  filter(hotel == "City Hotel") %>%
  mutate(status = ifelse(is_canceled == 1, "Canceled", "Not Canceled")) %>%
  count(status) %>%
  mutate(percent = n / sum(n),
         label = paste0(round(percent * 100, 3), "%"))

ggplot(city_cancel, aes(x = "", y = percent, fill = status)) +
  geom_col(width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Reservation Cancellation Status in City Hotel") +
  theme_void() +
  geom_text(aes(label = label), position = position_stack(vjust = 0.5), size = 4)
```

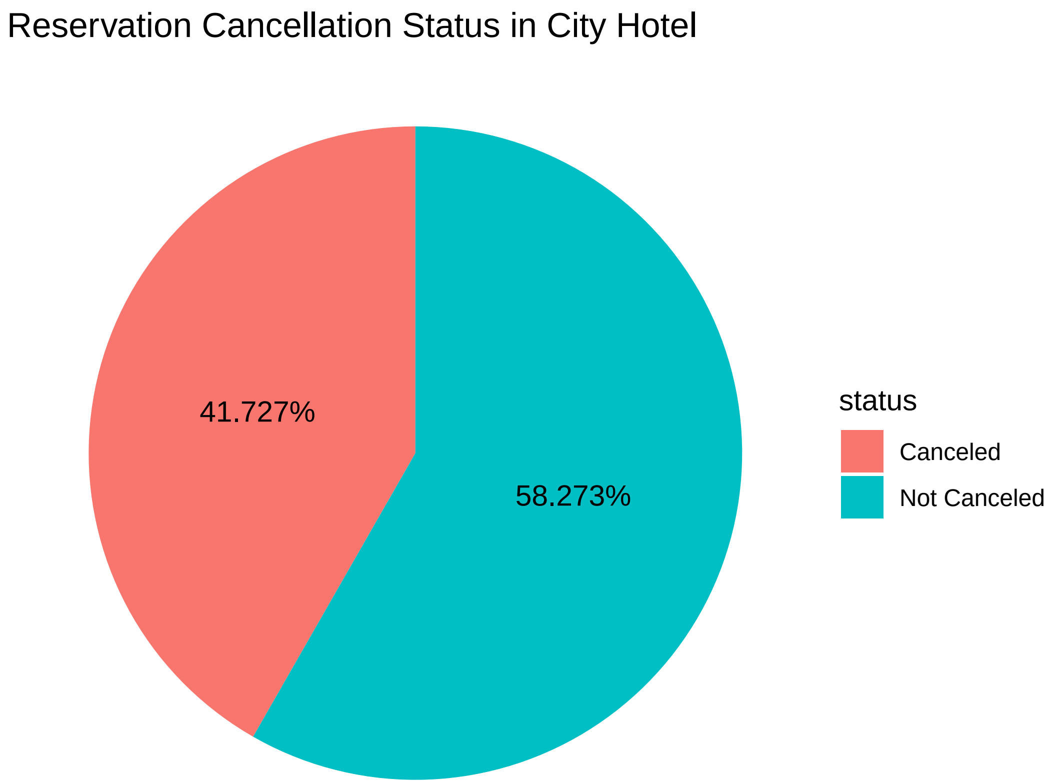

Figure 4: Figure 4. Reservation Cancellation Status in City Hotel

3.5 Reservation Counts per Year

```
yearly_data <- df %>%
  group_by(arrival_date_year) %>%
  summarise(reservations = n())

ggplot(yearly_data, aes(x = factor(arrival_date_year), y = reservations, fill = factor(arrival_date_yea
  geom_col() +
  labs(title = "Number of Reservations per Year",
       x = "Arrival Year",
       y = "Number of Reservations") +
  theme_minimal() +
  theme(legend.position = "none")
```

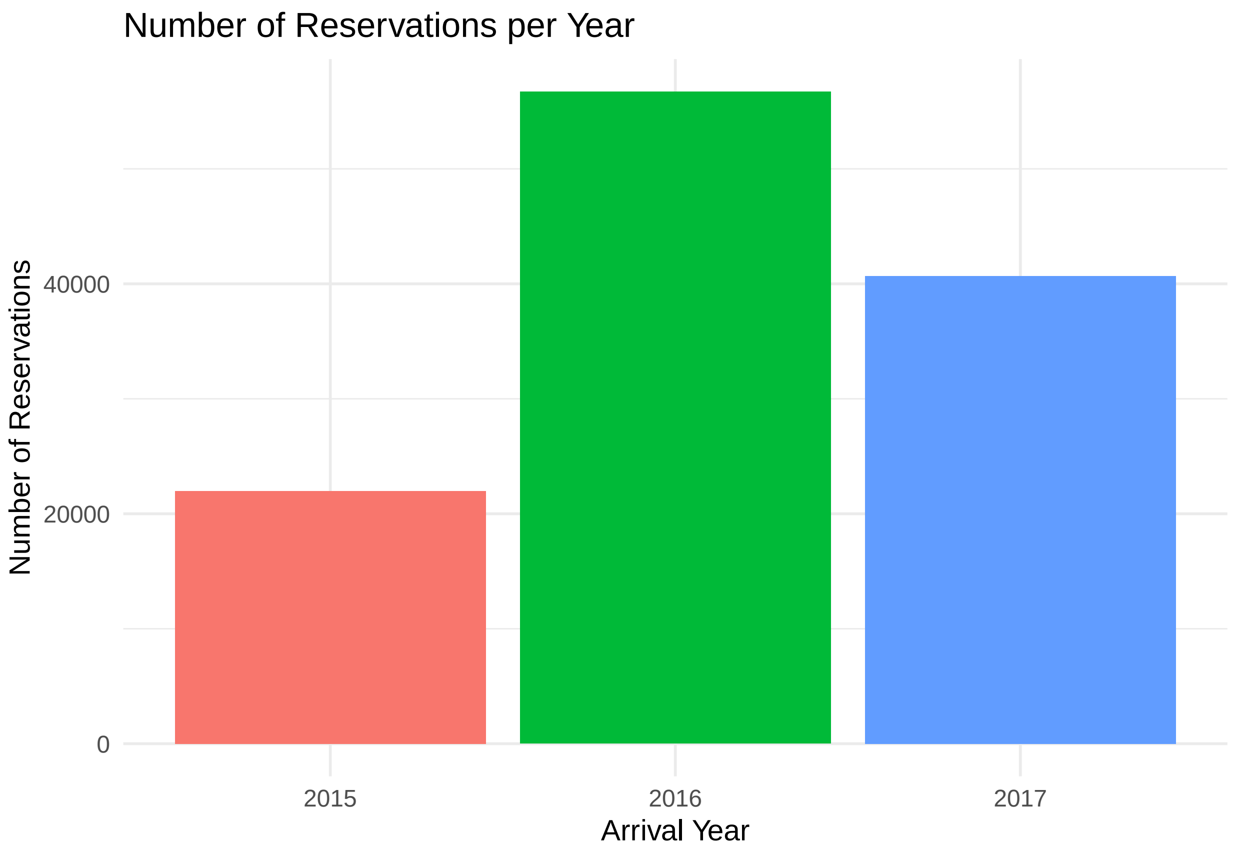

Figure 5: Figure 5. Number of Reservations per Year

3.6 Monthly Average Guest Count per Hotel

```
df$arrival_date_month <- factor(df$arrival_date_month,
                                levels = c("January", "February", "March", "April", "May", "June",
                                           "July", "August", "September", "October", "November", "Decem

monthly_guests <- df %>%
  mutate(total_guests = adults + children + babies) %>%
  group_by(arrival_date_month, hotel) %>%
  summarise(avg_guests = mean(total_guests, na.rm = TRUE)) %>%
  ungroup()

ggplot(monthly_guests, aes(x = arrival_date_month, y = avg_guests, color = hotel, group = hotel)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Average Number of Hotel Guests per Month",
       x = "Month",
       y = "Number of Guests") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

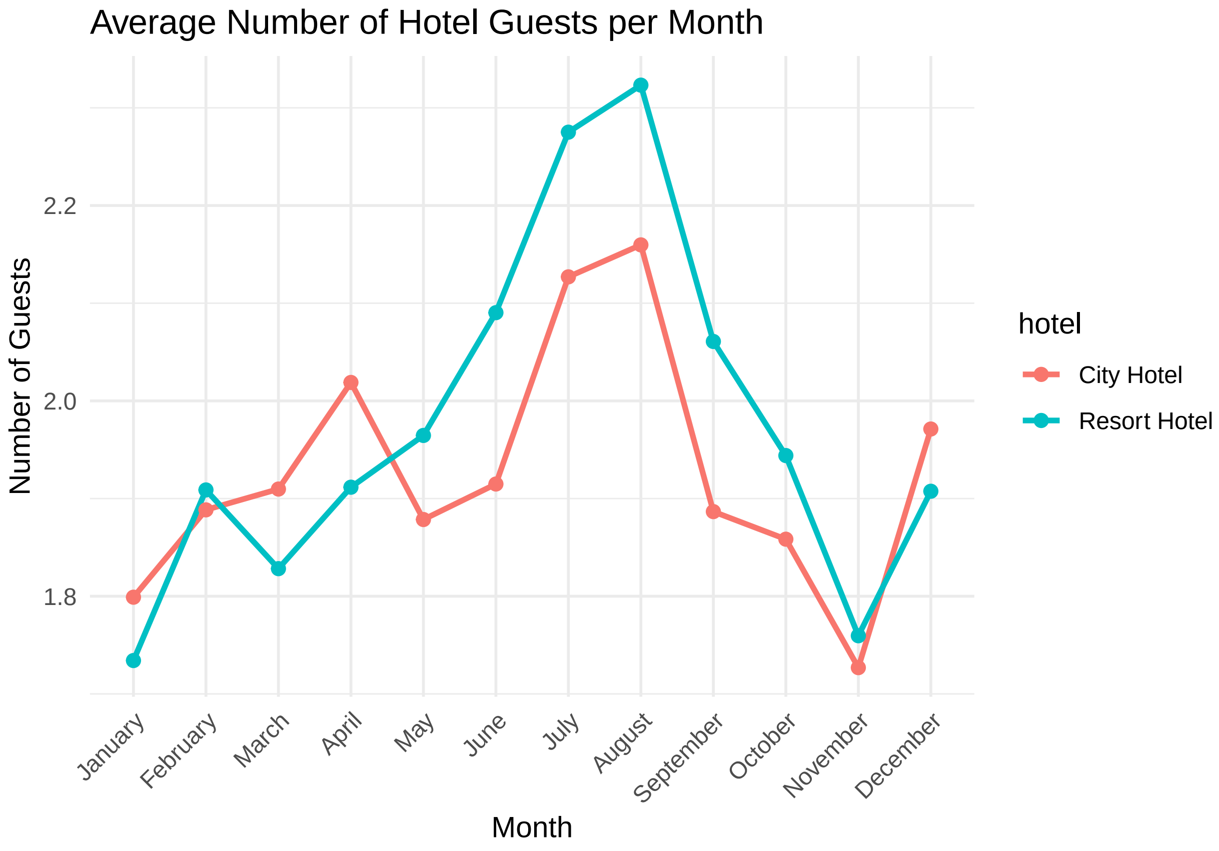

Figure 6: Figure 6. Average Number of Hotel Guests per Month

4 Analysis Process Designed for the Problem

4.1 Selected Data Mining Methods and Rationale

In this study, we utilize **XGBoost** and **CatBoost** as the primary data mining methods.

XGBoost is an algorithm based on gradient boosting decision trees, which improves model performance by sequentially adding new trees to fit the residuals of the previous weak classifiers. As more trees are added, the model gradually converges. For prediction, the model feeds the input data into each tree, determines which leaf node it falls into, and retrieves the corresponding score. The final prediction is the sum of all leaf scores across all trees. Additionally, XGBoost automatically handles missing values, supports parallel computation, and incorporates regularization techniques to reduce the risk of overfitting.

CatBoost is a gradient boosting algorithm optimized for categorical data. Unlike traditional boosting algorithms, CatBoost builds **symmetric trees**, where the split structure is the same across all branches. One of its key advantages is the ability to convert categorical features into numerical ones internally using **target encoding**, which encodes categories based on the conditional probability of the target variable. This allows the model to preserve the relationship between categorical features and the target, significantly reducing preprocessing effort. Furthermore, CatBoost applies **combinatorial category features** to enhance generalization, making it especially effective for datasets with many categorical variables.

We selected **XGBoost** and **CatBoost** primarily for their **accuracy and efficiency**. Both are known for delivering high prediction accuracy and handling class imbalance effectively. They can also extract important information from a large number of features and are highly flexible, adapting well to various data structures. This is particularly valuable for domains such as tourism, where time and seasonal patterns play a critical role in data analysis.

Certainly! Here's the English translation and continuation of section **3.2 Model Evaluation Method**:

4.2 Model Evaluation Method

We employed two data splitting strategies. The **first approach** uses data from the end of 2015 to the end of 2016 as the **training set**, with 25% of that portion further designated as the **validation set**, divided using **monthly time blocks**. The data from **January to June 2017** is used as the **test set**. The **second approach** involves randomly splitting the entire dataset into **75% training data** and **25% testing data**, followed by a **5-fold cross-validation** on the training set to more rigorously evaluate model performance.

Given the strong correlation between travel behavior and both time and seasonality, we segmented the data based on check-in time by creating **"month/year" time blocks**. This temporal structuring enables more accurate trend analysis and forecasting in the travel domain.

To comprehensively evaluate model performance, we used **accuracy**, **recall**, and **precision** as our primary metrics. In addition, we utilized the **ROC curve** to compare model performance, ensuring that the selected models maintain robust predictive capability under various scenarios.

Furthermore, we also analyzed the **confusion matrix** to observe the distribution of true positives, false positives, true negatives, and false negatives. This helped us understand where the models may be prone to errors—such as over-predicting cancellations or failing to identify them. By combining these evaluation methods, we ensured that our final model selection was not only based on overall accuracy but also on its practical utility in minimizing misclassification risks in a real-world hotel booking context.

4.3 Platforms and Tools Used

- **Operating System**: Windows 10, 11
- **Development Environment**: RStudio
- **Programming Language**: R
- **Libraries Used**:
  - `ggplot2` – for data visualization
  - `caret` – for model training and evaluation
  - `dplyr` – for data manipulation
  - `data.table` – for efficient data handling
  - `xgboost`, `catboost` – for model building
  - other supporting libraries as needed (e.g., `readr`, `lubridate`, `pROC`)

These tools provided a flexible and efficient environment for data preprocessing, modeling, visualization, and evaluation within the R ecosystem.

4.4 Research Workflow and Explanation

Figure 7. Workflow of This Study

- **Data Preprocessing**: In this step, we removed columns that could lead to data leakage (e.g., `reservation_status`, `reservation_status_date`). Columns with little meaning or excessive missing values (e.g., `company`, `country`, `agent`) were also excluded. For numerical features, we imputed missing values using the **median**, while categorical features were filled with either the **most frequent category** or a placeholder such as **"Unknown"**, depending on the column context. Date fields were combined and formatted correctly, and outliers were mapped or clipped appropriately to complete the data cleaning process.
- **Feature Selection & Data Visualization**: For feature selection, we employed both **Information Gain** and **SHAP (SHapley Additive exPlanations)** to identify the most important variables. A total of **15 features** were retained for model training. Data visualization was also used to understand feature distributions and correlations.
- **Model Building & Evaluation**: We trained and evaluated our models using **XGBoost** and **CatBoost**. Finally, we visualized the decision-making process of the weak classifiers in the boosting model through **decision tree visualizations**, making the classification basis more interpretable.
- **Conclusion**: Based on the experimental results and the objectives defined earlier, we conducted a comparative analysis to evaluate the effectiveness of our proposed approach in predicting hotel booking cancellations, and proposed suggestions for future improvement.

4.4.1 Workflow Diagram (Recreated for You)

Below is a diagram visually representing the process you described:

```r
library(DiagrammeR)

grViz("
digraph workflow {
  graph [layout = dot, rankdir = LR]

  node [shape = polygon, sides = 6, peripheries = 1, style = filled, fontname = Helvetica, fontsize = 12

  A [label = 'Data Preprocessing', fillcolor = lightblue]
  B [label = 'Feature Selection\n& Data Visualization', fillcolor = lemonchiffon]
  C [label = 'Modelling\n& Evaluation', fillcolor = lightpink]
  D [label = 'Conclusion', fillcolor = palegreen]

  A -> B -> C -> D
}
")
```