Project Report
on
Hotel Booking Cancellations

Submitted By
Sri Mahalakshmi Harika Punati
NU ID: 002790755
College of Engineering

IE7945 21657 Master's Project
Fall 2024

Master of Science in Data Analytics Engineering

Submitted to: Prof. Sivarit Sultornsanee

Northeastern University,

Boston, Massachusetts

Submitted Date: Dec 8th, 2024

TABLE OF CONTENTS

1.INTRODUCTION

Hotel reservations are a major source of income for the hospitality sector, which is a pillar of the world economy. Every year, millions of people travel for work, play, or pleasure, making hotels an essential component of the service economy. Yet, despite their vital role, hotels continue to face a recurring problem that affects them all over the world: cancellations of reservations. Despite being frequent, cancellations have far-reaching effects. They cause vacant rooms, interfere with the scheduling and logistics of hotel operations, and cause considerable financial losses. While a single cancellation may not seem like much, a series of them can have a significant negative effect on a hotel's income, staffing levels, and resource usage. The unpredictable nature of consumer behavior drives booking cancellations, often due to last-minute itinerary changes or better offers. While cancellations are inevitable, they challenge hotels inefficiently allocating resources, especially when unprepared for these disruptions. Hotel booking cancellations is a significant challenge that affects revenue, resource management, and customer satisfaction. If not properly addressed, cancellations can lead to overbooking, inefficient use of resources, and unhappy guests, ultimately damaging a hotel's reputation and profitability. Cancellations disrupt accurate occupancy predictions, leading to issues such as overstaffing or underutilized amenities. This misalignment increases costs and reduces a hotel's operational flexibility. Underestimating cancellations can lead to underutilized resources and missed revenue opportunities. These scenarios underscore the delicate balance hotels must maintain to optimize occupancy while minimizing disruptions.

Addressing hotel booking cancellations is critical for the hospitality sector because it has a multidimensional impact on different parts of hotel operations and performance. Accurately predicting cancellations helps maximize revenue by minimizing losses from unoccupied rooms, particularly during peak periods, and allows for more effective overbooking strategies to ensure optimal occupancy. Understanding cancellation probabilities also aids in optimizing resource management, such as staffing and amenities, thus reducing operational waste and cost inefficiencies. By proactively managing cancellations, hotels can enhance customer satisfaction through personalized solutions like flexible policies and tailored offers, which improve guest experience and loyalty. Additionally, mitigating the impacts of cancellations provides a competitive advantage by strengthening a hotel's market position and attracting more customers with reliable booking processes and better value propositions. Reliable cancellation predictions also contribute to financial stability by enabling better financial forecasting and planning, protecting against unpredictable revenue fluctuations.

The use of machine learning models offers a powerful framework for accurately predicting hotel booking cancellations, enabling data-driven decision-making. This empowers hotel managers to anticipate guest behavior and adjust their strategies accordingly. By analyzing historical data, including booking lead times, guest demographics, seasonal trends, payment preferences, and previous cancellations, machine learning algorithms can identify patterns and predict the likelihood of cancellations with greater accuracy. These algorithms uncover complex relationships and trends that traditional methods often overlook, significantly enhancing the ability to forecast cancellations and optimize resource management.

This study employs Logistic Regression and Gradient Boosting Machines (GBMs) to address cancellation prediction challenges. Logistic Regression provides interpretability and computational efficiency, while GBMs excel in capturing complex, non-linear interactions and handling imbalanced datasets. The combination of these models achieves a balance between predictive accuracy, transparency, and adaptability to dynamic booking behaviors. The research incorporates advanced preprocessing methods, including feature engineering and dimensionality reduction using Principal Component Analysis (PCA). These techniques optimize the dataset, reduce computational overhead, and enhance model performance. The study evaluates cancellation risks with real-time adaptability, enabling hotels to implement proactive strategies such as targeted interventions and efficient resource allocation

By addressing limitations like class imbalance, model overfitting, and interpretability challenges, this study offers a scalable, practical solution to hotel booking cancellations. The adoption of machine learning represents a significant shift in the hospitality industry, allowing hotels to mitigate revenue losses, improve operational efficiency, and enhance guest satisfaction while maintaining competitiveness in an evolving market landscape

2.Literature Review

2.1 Previous Work:

The main in this previous work, frequent cancellations that is disrupting operations, which causing revenue loss in the hospitality industry by exploring the existing research and methodologies surrounding hotel booking cancellation prediction. The focus centers on analyzing machine learning techniques, critically examining dataset characteristics, evaluating performance metrics, and understanding the practical implications for the hospitality industry. By synthesizing current scholarly work, I aim to provide a comprehensive overview of the state-of-the-art approaches in predicting booking cancellations, highlighting both methodological strengths and potential areas for future research.

This already existing study explores provides an in-depth analysis of predicting hotel booking cancellations through machine learning techniques, with a particular emphasis on Decision Tree and Random Forest algorithms. Addressing the issue of booking cancellations, presents a solution utilizing predictive models, and assesses their effectiveness.

The dataset was obtained from Kaggle and included various features related to bookings, guest demographics, and cancellations. It had both categorical and numerical variables, with the primary target variable being is_canceled, which indicates whether a booking was canceled or not.

Performed EDA where they have computed metrics such as mean, median, standard deviation, and range for numerical features. Identified outliers and anomalies that could distort model performance. Lead time showed a significant range, indicating that bookings are made anywhere from days to months in advance. ADR values highlighted notable pricing differences, with potential outliers in the high-end category.

Frequency distributions were analyzed for features like hotel and reservation_status. These distributions provided insights into dominant booking patterns and guest preferences. Descriptive statistics were calculated for numerical features such as lead_time, adr (average daily rate), stays_in_weekend_nights, and stays_in_week_nights. Computed metrics such as mean, median, standard deviation, and range for numerical features. Identified outliers and anomalies that could distort model performance.

Visual tools like histograms revealed skewed distributions, indicating the need for potential transformations during preprocessing. Box plots Highlighted the variability of numerical features, particularly focusing on the adr variable to uncover pricing patterns and outliers. City hotels showed a narrower price range compared to resorts.

When Observing Guest behavior **Geographical Distribution** is shown Guest origins were mapped to understand the dataset's diversity. A significant number of bookings originated from regions with high cancellation tendencies, highlighting a possible geographical bias in cancellations.

Booking Trends are displayed on time-based bookings were analyzed Monthly variations in guest arrivals were charted. guest volumes across months for city vs. resort hotel are shown. Peak booking periods were identified, which may influence cancellation rates. The number of nights stayed by guests was analyzed by hotel type. City hotels tended to have shorter stays, while resorts attracted longer visits

The heatmap visualization highlighted the correlations between numerical features and the target variable (is_canceled). Notable features such as lead_time, average daily rate (adr), and days_in_waiting_list exhibited significant positive or negative correlations with the likelihood of cancellations. For example, lead_time showed a strong correlation with cancellations, indicating that reservations made well in advance tended to have a higher cancellation rate. Additionally, the analysis of previous cancellations and prior bookings that were not canceled provided valuable insights into patterns of guest loyalty.

Key Insights that have shown from EDA: City hotels displayed less price variation compared to resorts, which exhibited a broader range of prices. This difference may reflect the elasticity of demand and its impact on cancellation rates. Additionally, seasonal patterns indicated that cancellations peaked during certain months, providing valuable insights for anticipating operational challenges. An analysis of guest types revealed that categories such as guests with previous cancellations or those making lastminute bookings were at a higher risk for cancellations.

The exploratory data analysis provided a solid foundation for feature engineering and model training. Through correlation analysis, they identified which features were most predictive of cancellations, prioritizing variables such as lead_time, average daily rate (adr), and days_in_waiting_list. And addressed outlier handling by capping or normalizing high outlier values in pricing and lead times to prevent them from skewing the model. Insights gained from the EDA informed the application of Principal Component Analysis (PCA) to reduce computational demands while preserving essential predictive features.

Decision Tree Model

A Decision Tree is a classification model that employs a tree-like structure to divide data into decision nodes and leaf nodes based on specific conditions. This model is valued for its simplicity and interpretability, making it easy to understand how decisions are made.The Decision Tree algorithm demonstrated reliable performance in predicting hotel booking cancellations, showcasing strengths in both interpretability and accuracy. Before applying Principal Component Analysis (PCA), the model achieved an accuracy of 82.51%, effectively capturing patterns and decision boundaries using the full feature set. However, this came at the cost of increased computational load. After applying PCA, the accuracy slightly decreased to 80.92%, but computational efficiency improved. The model effectively captures interactions between features, which helps in identifying complex patterns in hotel booking data However, it also has weaknesses such as overfitting especially with large datasets like the hotel booking dataste used(119,390 rows). This happens when the model memorizes the data instead of generalizing patterns.Decision Trees can struggle with large datasets unless pruning or other optimization methods are applied. Decision Trees are sensitive to feature removal and may perform poorly with reduced data.Due to their greedy algorithm, and focus on local optimizations, which can lead to poor generalization to new data.Their tendency to overfit and struggle with generalization may limit their long-term effectiveness for handling complex datasets.

Random Forest

Random Forest is an ensemble learning technique that integrates multiple decision trees to enhance predictive performance. By aggregating the outputs of these individual trees, it offers greater robustness and improved accuracy while effectively mitigating the risk of overfitting. The Random Forest algorithm demonstrated superior performance in predicting hotel booking cancellations compared to the Decision Tree model. Before applying Principal Component Analysis, model achieved an accuracy of 86.63%, effectively leveraging the full feature set and improving classification reliability through its ensemble learning structure. After PCA, the accuracy slightly decreased to 85.87%, reflecting efficiency gains from dimensionality reduction while maintaining the model's ability to identify key patterns. By averaging predictions across multiple decision trees, Random Forest reduces the risk of overfitting. This is particularly valuable for the hotel cancellation dataset, which likely contains noisy and complex features. By averaging predictions across multiple decision trees, Random Forest reduces the risk of overfitting. This is particularly valuable for the hotel cancellation dataset, which likely contains noisy and complex features. Random Forest maintained strong accuracy after PCA, showing it can handle reduced feature sets effectively. However, it has weaknesses such as computational intensity, reduced interpretability compared to simpler models, and sensitivity to class imbalance. Poor tuning can reduce effectiveness. Despite these limitations, Random Forest model ability to handle data complexity and minimize errors makes it a robust choice for predicting hotel booking cancellations. Nevertheless, the Random Forest maintained high performance, demonstrating its robustness. Random Forest demonstrated the capacity to maintain elevated levels of accuracy, precision, and recall even when utilizing a diminished set of features. Such robustness implies that this model possesses the ability to predict cancellations effectively with reduced computational demands, thereby rendering it particularly advantageous for real-time applications where efficiency in processing is essential

2.2 Limitations of the Previous Research:

While the study on hotel booking cancellation prediction presents valuable insights, several limitations could affect its effectiveness and broader applicability. Addressing these issues is critical to improving the model's utility for real-world applications.

Class Imbalance

The previous study does not explicitly tackle class imbalance, a common problem in cancellation datasets where one class (e.g., "not canceled") often outweighs the other. This imbalance can bias the model, reducing its ability to predict cancellations accurately. As a result, recall for cancellations, which are essential for hotel resource planning, may be poor. Techniques like oversampling, undersampling, or cost-sensitive algorithms could help mitigate this issue and improve the model's performance.

Interpretability

Even though Decision Trees offer interpretability, the Random Forest model used in the study is less transparent. This lack of interpretability may limit adoption by non-technical stakeholders, such as hotel managers, who need clear explanations for decision-making. Incorporating feature importance visualizations and post-hoc interpretability tools could make the model's decisions more understandable and actionable.

Dataset Quality

The research relies on a preexisting dataset that may contain missing, noisy, or outdated data. While preprocessing steps are mentioned, the study provides limited discussion on data quality. Poor-quality data, especially if key features are incomplete or incorrect, can reduce model accuracy and reliability. Future work should focus on integrating real-time booking data and conducting rigorous data cleaning and validation to ensure the dataset is comprehensive and reliable.

Dynamic Nature of Bookings

The study does not account for time-sensitive variables like time-to-stay, booking season, or external factors such as economic Class Imbalance**.** The research does not explicitly tackle class imbalance, a common problem in cancellation datasets where one class (e.g., "not canceled") often outweighs the other. This imbalance can bias the model, reducing its ability to predict cancellations accurately. As a result, recall for cancellations, which are essential for hotel resource planning, may be poor. Techniques like oversampling, undersampling, or cost-sensitive algorithms could help mitigate this issue and improve the model's performance. Conditions, weather, or local events. These factors often play a significant role in cancellations and may change dynamically over time. Incorporating time-series modeling techniques and external datasets could improve the model's ability to adapt to evolving patterns, such as last-minute cancellations.

Scalability and Computational Costs

While the Random Forest model performs well, it is computationally intensive, making it less practical for real-time applications or smaller hotels with limited resources. This restricts the model's scalability to larger datasets or resource-constrained environments. Using lightweight algorithms or optimizing the current model could reduce computational costs and make it more suitable for such use cases.

Evaluation Metrics

The study primarily evaluates the model's using accuracy, precision, recall, and F1-score. However, it does not consider business-specific metrics such as cost savings, revenue impact, or operational efficiency, which are critical for practical applications in hotel management. Including these business-oriented metrics could provide a more comprehensive evaluation of the model's real-world value and alignment with industry goals.

2.3 Addressing gaps with proposed methods

The previous research on hotel booking cancellations highlights critical insights and methodologies, but it also reveals several limitations that restrict its broader applicability and real-world impact. By leveraging Logistic Regression (LR) and Gradient Boosting Machines (GBMs), this study addresses these gaps with innovative approaches that enhance predictive accuracy, interpretability, and practical usability.

Addressing Class Imbalance: the previous studies do not adequately handle class imbalance, leading to biased predictions favoring the majority class (not canceled) over the minority class (canceled). This results in poor recall for cancellations, which are crucial for hotel resource planning. Logistic Regression incorporates cost-sensitive training to emphasize the minority class, improving recall for cancellations without requiring extensive preprocessing. GBMs use weighted loss functions during training to iteratively focus on hard-to-predict samples, effectively balancing class distributions and enhancing predictions for the minority class.

Enhancing Interpretability: While Decision Trees offer some interpretability, Random Forests lack transparency, making them less actionable for non-technical stakeholders such as hotel managers. Logistic Regression provides clear, interpretable coefficients, allowing managers to understand the direct impact of features like lead_time, adr, and previous_cancellations. This facilitates data-driven decisions such as personalized guest retention strategies. GBMs include feature importance rankings and visualization tools, offering actionable insights into which features (e.g., deposit_type, market_segment) drive cancellations. These insights enable targeted interventions while maintaining high model accuracy.

Accounting for Dataset Quality: The previous research relies on preexisting datasets with potential issues such as missing or noisy data but provides limited discussion on data quality and cleaning processes. This study employs extensive preprocessing techniques, such as handling missing values, addressing outliers through capping and normalization, and engineering features like total_nights. These steps improve data quality and enhance model reliability. The use of Principal Component Analysis (PCA) optimizes the dataset by reducing noise and redundancy, ensuring computational efficiency without sacrificing essential predictive features.

Capturing Dynamic Booking Behaviors: The static models used in previous research fail to account for dynamic and time-sensitive variables like booking season, lead time, and external factors (e.g., weather, local events). This limits adaptability to evolving patterns. GBMs excel in modeling complex, non-linear relationships, allowing them to capture dynamic interactions between features such as lead_time and market_segment. This adaptability ensures better prediction accuracy in scenarios with shifting booking behaviors. Future integration of real-time data streams (e.g., last-minute bookings or external factors) is feasible with GBMs, enhancing their ability to anticipate cancellations in real-world scenarios.

Improving Scalability and Efficiency: Computationally intensive models like Random Forests face scalability challenges, making them less practical for real-time applications or smaller hotels with limited resources. Logistic Regression offers a lightweight and computationally efficient alternative, ideal for real-time applications and smaller operations. GBMs are optimized with PCA to reduce computational overhead while maintaining high accuracy, making them suitable for larger datasets or hotels with greater resource availability.

Incorporating Business-Oriented Evaluation Metrics: Previous studies primarily rely on accuracy, precision, recall, and F1-score, neglecting business-specific metrics such as cost savings, revenue impact, and operational efficiency. This study evaluates model performance with a focus on actionable business outcomes, such as optimizing overbooking strategies and minimizing revenue loss. By linking predictions to practical applications, the models ensure alignment with industry goals.

3.Proposed Method

3.1 Logistic Regression

Logistic Regression effectively addresses several key limitations observed in Decision Trees and Random Forests, particularly in the areas of interpretability, overfitting, and handling small datasets. Also, serves as a dependable benchmark for evaluating the performance of more advanced algorithms. It effectively models the linear relationships between key features, such as lead_time, adr, and previous_cancellations, and the likelihood of cancellations, offering actionable insights for hotel management.

Key Strengths:

Feature Interpretability: Logistic Regression provides direct interpretability through its coefficients. For example, positive coefficients for variables like **lead_time**, **adr**, and **previous_cancellations** indicate their positive correlation with the likelihood of cancellations. This makes it a powerful tool for decision-making, enabling hotel managers to design targeted strategies, such as offering discounts to guests with long lead times or providing flexible cancellation policies for high-risk bookings.

Linear Relationships: Logistic Regression is well-suited to datasets with features that exhibit nearly linear relationships with the target variable, as seen in this dataset with variables like **average daily rate (adr)** and **lead_time**.

Scalability and Efficiency: As a computationally lightweight model, Logistic Regression scales efficiently, making it suitable for real-time decision-making in dynamic hotel environments.

Overfitting Prevention: Regularization techniques such as L1 and L2 penalties are integrated to minimize overfitting, particularly valuable in datasets with noise or multicollinearity.

Handling Imbalanced Data: While Logistic Regression inherently struggles with imbalanced datasets, its performance can be enhanced using techniques like regularization or class weights. This is particularly relevant for the dataset, where cancellations (Class 1) are the minority.

Adaptability to Mixed Feature Types: Logistic Regression performs well with the dataset's mix of numerical (e.g., adr, lead_time) and categorical features (e.g., meal, market_segment) after preprocessing

Implementation for Logistic Regression:

Its performance can be used as a benchmark to evaluate more complex models, providing a clear reference point for assessing the added value of sophisticated algorithms. The advantages of Logistic Regression in this context are manifold. It is relatively easy to implement and interpret, making it accessible to a wide range of users, including those without extensive statistical backgrounds. The model performs well when dealing with linear or nearly linear relationships between variables, which often occur in hotel booking data. Additionally, it scales efficiently to moderately large datasets, making it suitable for hotels of various sizes. However, it is important to acknowledge the limitations of Logistic Regression. Its primary constraint lies in its limited ability to handle non-linear patterns, which may exist in complex datasets typical of hotel bookings. For instance, the relationship between lead time and cancellation probability might not be strictly linear, potentially limiting the model's accuracy in capturing such nuanced relationships. In summary, Logistic Regression offers a powerful, interpretable, and efficient approach to predicting hotel booking cancellations. While it may not capture all the complexities of hotel booking data, its strengths in interpretability and efficiency make it a valuable tool in the predictive modeling toolkit for hotel management.

Studies have increasingly employed Logistic Regression for hotel booking cancellation predictions due to its interpretability and computational efficiency. Here are some relevant studies:

- The analysis of hotel booking cancellations was conducted utilizing Logistic Regression in conjunction with deep neural networks. The results obtained indicated that Logistic Regression serves as an effective tool for identifying significant variables that impact the probability of cancellations, including the "state" attribute and special requests. The selection of Logistic Regression as a baseline method is attributed to its straightforward nature and the interpretability it offers, particularly in the context of distinguishing influential features.
- A comparative analysis was conducted between Logistic Regression and other machine learning algorithms, specifically Random Forest and Decision Trees, in the context of predicting hotel cancellations. This investigation identified Logistic Regression as a foundational model, valued for its transparency, which renders it suitable for preliminary feature analysis. Nonetheless, the results of the study indicated that, although Logistic Regression facilitated some understanding of the

data, ensemble models demonstrated marginally superior accuracy when applied to more complex data structures. Thus, the advantages of utilizing ensemble approaches in predictive modeling were established, highlighting the nuanced performance differences among the evaluated models.

- The investigation encompassed the evaluation of multiple models, including Logistic Regression, in the context of hotel booking cancellations. Logistic Regression was identified as an essential baseline model, attributed to its minimal computational demands. The analysis conducted within the study highlighted the effectiveness of Logistic Regression in discerning distinct and interpretable patterns. Furthermore, it illustrated the model's significance in establishing a foundational basis for the development of more sophisticated modeling techniques.

The effectiveness of Logistic Regression in predictive tasks has been frequently emphasized in research that contrasts it with alternative modeling approaches, especially when linear relationships and uncomplicated decision boundaries suffice. A selection of pertinent studies is presented below:

- The examination of Logistic Regression and Decision Trees in the context of customer churn prediction has revealed significant insights. A novel hybrid model, termed the Logit Leaf Model (LLM) has been proposed, which integrates the advantages of both Logistic Regression and Decision Trees. This innovative approach has been demonstrated to enhance predictive performance and facilitate interpretability. Evidence suggests that the Logit Leaf Model surpasses the effectiveness of traditional standalone models, especially in its capacity to manage both linear and non-linear effects. The interpretability inherent in Logistic Regression is recognized as a fundamental baseline that adds value to the overall predictive process. Thus, the combination of these methodologies yields a more robust framework for addressing customer churn.
- The traditional Logistic Regression model was improved through the integration of rules derived from decision trees specifically for the purpose of credit scoring. This innovative approach capitalized on the inherent transparency offered by Logistic Regression in the scoring process, while simultaneously incorporating the capacity of decision trees to manage non-linear relationships. As a result, this methodology provided greater interpretability when contrasted with more opaque ensemble models such as Random Forest.
- An evaluation of various regression models applied to agronomic data indicates that decision trees and random forests exhibit performance levels comparable to those of partially linear regression models. In contrast, Logistic Regression

emerges as a preferred option for less complex tasks, owing to its effectiveness in modeling linear relationships and its capacity to mitigate the risk of overfitting in simpler datasets.

The simplicity and speed of Logistic Regression make it ideal for real-time applications, especially for smaller hotel chains with limited computational resources. Its lightweight nature enables quick predictions, making it suitable for dynamic decisions like overbooking strategies without overwhelming operational systems. For instance, the model can provide fast and reliable predictions for real-time decision-making.

3.2 Gradient Boosting

Gradient Boosting Machines (GBMs) effectively address several limitations observed in Decision Trees and Random Forests particularly in handling non-linear relationships, imbalanced data, feature redundancy, and real-time adaptability. Capturing Complexity, including implementations like XGBoost and LightGBM, are sophisticated ensemble methods that effectively address the limitations of simpler models such as Random Forest and Decision Trees. They are particularly well-suited to the dataset due to several key capabilities.

Handling Non-Linear Relationships: GBMs excel at modeling intricate, non-linear interactions between features. For instance, they can accurately capture how a combination of **low adr** and **high lead_time** increases the likelihood of cancellations. Their tree-based structure allows them to split data based on thresholds, which remains robust even after PCA.

Adaptability to Imbalanced Data: Weighted loss functions and boosting mechanisms help GBMs focus on minority classes, such as cancellations in this dataset, improving recall and precision for Class 1.

Feature Redundancy Management: GBMs naturally prioritize relevant features during training. For instance, redundant features like **reservation_status_date** and **lead_time** are downweighed, allowing the model to focus on more informative predictors.

Real-Time Adaptability: GBMs support incremental learning, enabling them to adapt quickly to new patterns in hotel booking trends, such as those driven by seasonality or promotions.

Robustness to PCA: GBMs were less affected by PCA because of their ability to capture patterns in transformed feature spaces. They rely on relative splits rather than direct linear relationships, maintaining performance despite dimensionality reduction.

Feature Importance Analysis: GBMs provide insights into the relative importance of features, such as **special_requests**, **market_segment**, or **total nights booked**, which helps hotel managers prioritize operational changes.

Given the dataset's moderate size and the linear relationships observed between key features and cancellations, Logistic Regression offers a clear and interpretable foundation for understanding predictive factors. While it may not capture complex, non-linear interactions, its efficiency and transparency make it ideal for quick implementation and as a benchmark for comparing more advanced methods.

Research has indicated the efficacy of LightGBM and XGBoost in forecasting booking cancellations, customer attrition, and analogous outcomes characterized by the interplay of multiple variables.

- The analysis conducted revealed that XGBoost exhibited a higher level of performance compared to Logistic Regression in the prediction of customer churn within the telecommunications sector. It was observed that XGBoost effectively addressed issues related to class imbalances and managed complex datasets pertaining to customer behavior more proficiently.
- American Express (AMEX) ranks among the most widely utilized credit card services in the United States. However, a recent trend has emerged, indicating the loss of a significant number of valued customers for various reasons. This study seeks to analyze the factors influencing customer cancellation behaviors through the application of the XGBoost algorithm, utilizing data obtained from AMEX. The implementation of this predictive tool facilitates the capability of the bank to anticipate customer behaviors and to adopt proactive measures in response. The findings of this study provide a foundation for further examination of customer dynamics in the context of credit card services, potentially guiding strategic decisions within AMEX.

The implementation of this predictive tool facilitates the capability of the bank to anticipate customer behaviors and to adopt proactive measures in response. The findings of this study provide a foundation for further examination of customer dynamics in the context of credit card services, potentially guiding strategic decisions within AMEX.

These machine learning algorithms exhibit a capacity for modeling complex relationships while simultaneously maintaining computational efficiency and interpretability. Consequently, they emerge as essential instruments for predictive tasks, such as the analysis of customer churn, booking cancellations, and credit risk assessment.

Although more computationally intensive than Logistic Regression, GBMs strike a balance between accuracy and resource requirements. Unlike Neural Networks, which demand significant computational power, GBMs deliver high accuracy without prohibitive hardware needs. For mid-sized hotel operations, GBMs provide the precision necessary for predicting cancellations while remaining manageable within typical resource constraint.

3.3 Why Other Machine Learning Models Were Not Selected

Neural Networks (NNs)

Neural Networks require large datasets to avoid overfitting, making them less suitable for moderately sized datasets like hotel bookings. Their "black box" nature also hinders interpretability, which is crucial for hotel managers who need actionable insights. Additionally, Neural Networks are computationally expensive, requiring high-performing hardware like GPUs or TPUs, which may not be practical for small-to-medium-sized hotel operations. NNs excel at capturing complex, nonlinear relationships and may slightly outperform GBMs on very large datasets. However, they are better suited for scenarios where interpretability is not a priority, and computational resources are abundant.

Support Vector Machines (SVMs)

SVMs face scalability challenges as their computational complexity increases quadratically with the number of samples. They also require extensive preprocessing, such as feature scaling and normalization, adding complexity to their implementation. Furthermore, like Neural Networks, SVMs lack interpretability, limiting their usefulness for providing actionable insights in operational settings. SVMs can perform well in high-dimensional spaces and can handle nonlinear relationships using kernel functions. However, they are better suited for smaller datasets and require careful hyperparameter tuning to achieve optimal results.

Decision Trees (DTs)

Decision Trees are prone to overfitting, particularly in datasets with many features or imbalanced classes. While highly interpretable, their standalone performance is typically lower than ensemble methods like GBMs, making them less suitable for high-stakes predictions such as cancellations. Potential Performances are valuable for initial exploratory analysis due to their interpretability. However, their limitations in accuracy and overfitting make them unsuitable for applications requiring reliable predictions.

Random Forests (RFs)

While Random Forests are robust, they are computationally heavier than Logistic Regression and lack the adaptive boosting mechanism of GBMs. This makes them less effective in addressing class imbalance or focusing on difficult cases within the dataset. RFs deliver strong performance on moderately complex datasets and are less resource-intensive than Neural Networks. However, GBMs typically outperform RFs by offering better accuracy and adaptability to imbalanced datasets.

k-Nearest Neighbors (k-NN)

k-NN is computationally expensive during prediction, as it requires calculating distances for all data points. It also struggles with irrelevant features, which can significantly impact its performance. k-NN works well for small datasets with low dimensionality. However, it is unsuitable for large and feature-rich datasets like those used in hotel cancellation prediction.

Naïve Bayes

Naïve Bayes relies on the assumption of feature independence, which is rarely valid in real-world datasets where feature interactions are important. Its simplicity often results in lower predictive power compared to GBMs or LR. Naïve Bayes is computationally efficient and useful for quick prototyping. However, its inability to handle complex feature interactions limits its effectiveness for predictive tasks like hotel cancellations.

3.4 An Overview of the Model Implementation

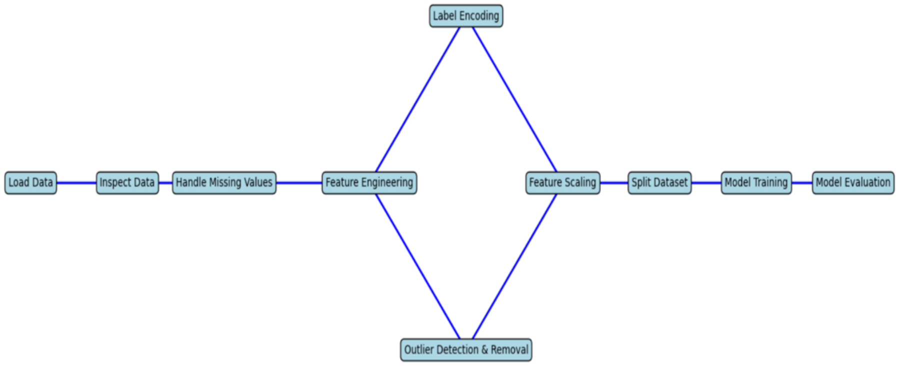


Fig 1 Structured Workflow for Data Analysis and Prediction

The data preprocessing and model development pipeline for the hotel booking cancellation prediction seen in Fig 1 encompasses several crucial steps. Initially, the dataset is loaded and inspected to understand its structure and quality. Missing values are then addressed through imputation or removal. Feature engineering follows, creating new variables or transforming existing ones to enhance model performance. Outliers are detected and handled to prevent skewing of results. Categorical variables are converted to numerical formats through label encoding, and numerical features are scaled for uniformity. The dataset is then split into training and testing subsets. Various machine learning models, such as Logistic Regression and Gradient Boosting Machines, are trained on the prepared data, with hyperparameter tuning applied to optimize performance. Finally, the models are evaluated using metrics like accuracy, precision, recall, and F1-score, along with tools such as confusion matrices and ROC-AUC curves. This comprehensive approach ensures a systematic and robust methodology for developing predictive models in the context of hotel booking cancellations.

4. Model Implementation

4.1 Dataset Overview:

The dataset is taken from Kaggle, provides a detailed record of hotel bookings, with 119,390 entries and 36 columns. It contains information about two types of hotels (Resort Hotel and City Hotel), as well as details regarding bookings, customer demographics, and behavior. The following is a full overview.

Booking Details:

hotel: Type of hotel (Resort or City). is_canceled: Binary indicator of whether a booking was canceled. lead_time: Number of days between booking and check-in. arrival_date_year, arrival_date_month, arrival_date_week_number, arrival_date_day_of_month:

Information on the booking arrival date.

Stay Information:

stays_in_weekend_nights: Number of weekend nights (Saturday/Sunday) in the booking.

stays_in_week_nights: Number of weekday nights in the booking. adults, children, babies: Count of guests in the booking.

Customer Information:

country: Guest's country of origin.

is_repeated_guest: Whether the guest is a returning customer.

customer_type: Type of customer.

Revenue and Requests:

adr (Average Daily Rate): Revenue generated per available room.

required_car_parking_spaces: Number of car parking spaces requested.

total_of_special_requests: Count of special requests made by the guest.

Additional Features:

meal: Meal package type selected.

market_segment, distribution_channel: Source and segment of the booking.

reservation_status, reservation_status_date: Status of the booking and its date.

4.2 Data Cleaning and Preprocessing:

Handling Missing Value: To prepare the dataset for analysis, missing data is systematically handled by replacing absent values in specific columns with appropriate defaults. For numerical columns, missing values are filled with 0, while categorical columns are assigned descriptive placeholders like "Unknown" or "No

Company". Additionally, the percentage of missing data in the company column is assessed; if more than 94% of its values are missing, the column is dropped to maintain dataset utility and manageability. This approach ensures a standardized, clear dataset, reducing ambiguity and enhancing its quality for subsequent analysis or modeling.

Feature Engineering: The dataset is enhanced by introducing a new feature, total_nights, which combines stays_in_weekend_nights and stays_in_week_nights into a single metric representing the total duration of a stay. Additionally, the arrival_date_month column, originally containing month names as strings, is converted to numerical values using a mapping dictionary. This transformation standardizes the month data, making it more suitable for numerical analysis, such as sorting, filtering, or applying machine learning algorithms. These modifications improve the dataset's usability and provide greater insight for advanced analytical tasks.

label encoding: Columns such as hotel, meal, market_segment, and others are categorical, containing text or label-based data. To prepare these variables for machine learning, label encoding is employed, assigning a unique numeric value to each category. This transformation converts categorical data into a numeric format suitable for inclusion in machine learning models, which require numerical inputs for computations. The hotel column with categories like "City Hotel" and "Resort Hotel" can be encoded as 0 and 1, respectively. Similarly, fields such as reserved_room_type and assigned_room_type, which indicate the type of room booked and the one assigned, are numerically encoded to help models detect patterns or discrepancies between the two. Ensuring that all categorical variables are numerically encoded, this step enhances the dataset's compatibility with machine learning workflows while preserving interpretability.

Data preprocessing:

The reservation_status_date column is converted to a datetime format to facilitate time-based analyses and feature engineering, such as extracting components like year, month, or day. The use of the errors='coerce' parameter ensures that invalid date entries are safely converted to NaT (Not a Time), maintaining data integrity. To address outliers, extreme values in the lead_time column are capped at the 95th percentile. This technique mitigates the influence of outliers that could skew analyses or adversely affect the performance of machine learning models, resulting in a more robust and reliable dataset.

Dropping Irrelevant Columns: Irrelevant identifier columns, including name, email, phone-number, and credit_card, are removed from the dataset. This step eliminates unnecessary information, streamlining the dataset for analysis and modeling.

Encoding the country Column: The categorical variable country is label-encoded to assign numeric values to its categories. This transformation enables the inclusion of the country variable in machine learning models.

Feature Engineering with Dates: The reservation_status_date column is converted to a datetime format to facilitate time-based analysis. Additional features, such as reservation_year, reservation_month, and reservation_day, are extracted to provide granular temporal information. After extracting these features, the original reservation_status_date column is dropped to avoid redundancy.

Summary of the Data:

Booking Details: Approximately **37% of bookings are canceled**, as indicated by the mean value of is_canceled (0.37). Lead times are normalized, with extreme values capped at the 95th percentile to reduce skewness.

Arrival Information: Bookings predominantly span the years **2016-2017** (mean for arrival_date_year = 2016.15). Monthly bookings are evenly distributed (mean for arrival_date_month = 5.55), while weekly bookings (arrival_date_week_number) range between **1 and 53**.

Stay Information: The majority of stays last fewer than **3 nights** (mean stays_in_weekend_nights = 0.93; mean stays_in_week_nights = 2.5), though some outliers represent longer stays.

Room and Reservation Information:

The reservation_status field is fully encoded for modeling, and extracted date features (reservation_year, reservation_month, reservation_day) show a diverse spread, with reservations primarily concentrated in mid-year months.

The reservation_status field is fully encoded for modeling, and extracted date features (reservation_year, reservation_month, reservation_day) show a diverse spread, with reservations primarily concentrated in mid-year months.

Demographics: The number of adults is normalized with low variability around the mean, while total_of_special_requests is typically minimal, peaking at 5 requests per booking.

4.3 Exploratory Data Analysis:

The histograms presented in Fig 1 showcase the distribution of crucial numerical features within the hotel booking dataset. These visualizations uncover specific patterns and trends in guest behavior and booking characteristics. The distributions emphasize the necessity of addressing data irregularities such as skewness and outliers. Recognizing and managing these aspects is essential for developing robust predictive and inferential models that accurately capture the underlying patterns in hotel bookings. The visualizations in Fig 1 emphasize the importance of preprocessing to address skewness, outliers, and irregularities in the data. These patterns reflect real-world booking behaviors, such as short stays, small group sizes, and low occurrences of children or infants, while also underscoring the need for preprocessing steps, including outlier treatment, normalization, and data validation. Such steps are vital to ensure the robustness and reliability of downstream predictive or inferential modeling. The inclusion of KDE curves enhances interpretability by providing a smooth approximation of the distributions, helping to identify underlying trends and draw meaningful insights from the dataset. The analysis of the distributions in Figure 1 highlights critical guest behaviors and booking trends that directly influence cancellations. Proper preprocessing, including outlier treatment, normalization, and validation, will enhance the robustness of predictive models and provide actionable insights for hotel management.

Lead Time Distribution

Fig 2 (Top Left) depicts the lead_time distribution is positively skewed, with most values concentrated in lower ranges, indicating that most bookings are made with short notice. However, a small proportion of outliers represent customers who book far in advance. The inclusion of a KDE (Kernel Density Estimation) curve highlights the steep decline in frequency as lead time increases. This skewness suggests the potential need for normalization or transformation to improve the suitability of this feature for statistical modeling.

ADR (Average Daily Rate) Distribution

Fig 2 (Top Middle) shows the adr feature distribution, which is highly skewed with a pronounced peak at lower values and a long tail extending towards higher daily rates. This suggests that most bookings are associated with lower daily rates, although a few outliers reflect significantly higher payments. These extreme values may indicate unique spending behaviors or niche pricing trends, making adr a critical feature for assessing pricing strategies and customer segmentation.

Total Nights Distribution

Fig 2 (Top Right) illustrates the total_nights distribution, revealing that most bookings involve stays of 1-3 nights, with a rapid decline in frequency for longer stays. A small number of extreme outliers representing extended stays may distort statistical analyses, necessitating capping or further examination. This pattern aligns with typical travel behaviors, where shorter trips are more common.

Adults Distribution

Fig 2 (Bottom Left) shows the distribution for adults, with distinct peaks at one and two adults, Cases involving larger groups are rare but include outliers at extremely high values, which may reflect data entry errors or infrequent group reservations. These anomalies warrant further validation to ensure the data's reliability.

Children Distribution

Fig 2 (Bottom Middle) depicts the histogram for children, showing a high frequency of bookings with zero children, while smaller peaks correspond to one or two children in bookings., while extreme outliers (more than 10 children) likely represent inaccuracies or highly unusual cases. Validation or treatment of these outliers is recommended to maintain data integrity.

Babies Distribution

Fig 2 (Bottom Right) illustrates the babies feature, which is heavily skewed, with most bookings involving no infants. A very small proportion of bookings include one or two babies, and extreme outliers (more than five infants) are likely erroneous or exceptionally rare cases. Addressing these outliers through removal or correction is essential for robust analysis and modeling.

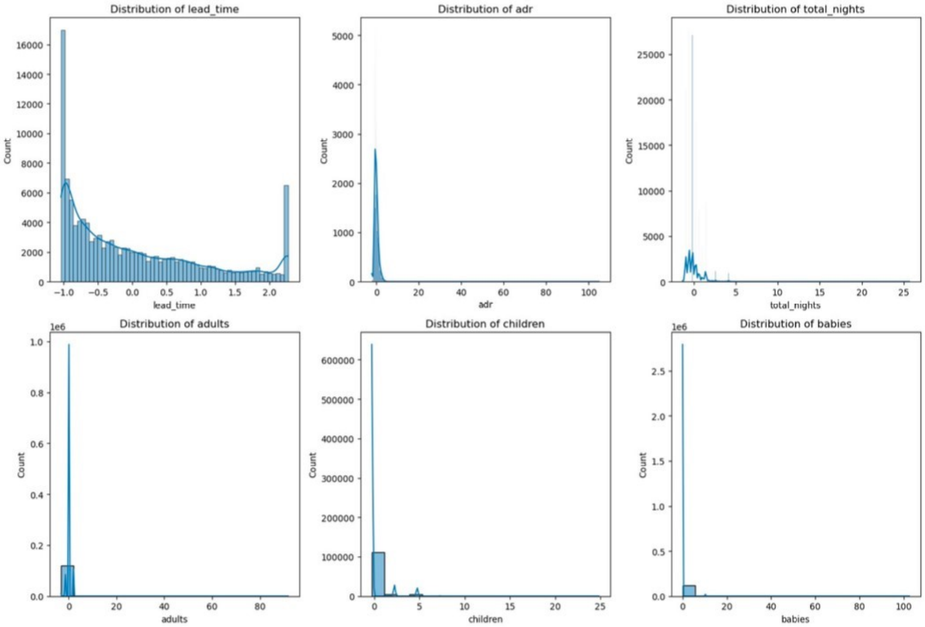


Fig 2 Histograms for key features (lead_time, adr, total_nights, adults, children, babies) reveal their distributions

The correlation heatmap in Fig. 3 provides a comprehensive view of the relationships between variables in the hotel booking dataset, offering crucial insights for predictive modeling. Key features such as lead_time (0.30), deposit_type (0.22), and total_of_special_requests (-0.23) show significant correlations with the target variable is_canceled, indicating their importance as predictors. The heatmap reveals strong positive correlations between related features like stays_in_week_nights and stays_in_weekend_nights (0.73), suggesting potential redundancies that could be addressed through feature aggregation. Negative correlations, such as between adr and deposit_type (-0.36), hint at pricing strategies tied to cancellation risks. The analysis also uncovers multi-collinearity risks and identifies features with weak correlations that may still contribute to non-linear models. These insights inform preprocessing recommendations, including feature scaling, addressing multi-collinearity, and careful handling of outliers. The heatmap's revelations about guest behavior, such as the tendency for early bookings and refundable deposits to correlate with higher cancellation rates, provide actionable insights for hotel management. Overall, the correlation analysis guides feature selection, preprocessing, and model development strategies, ensuring a robust approach to predicting hotel booking cancellations and optimizing operational decisions.

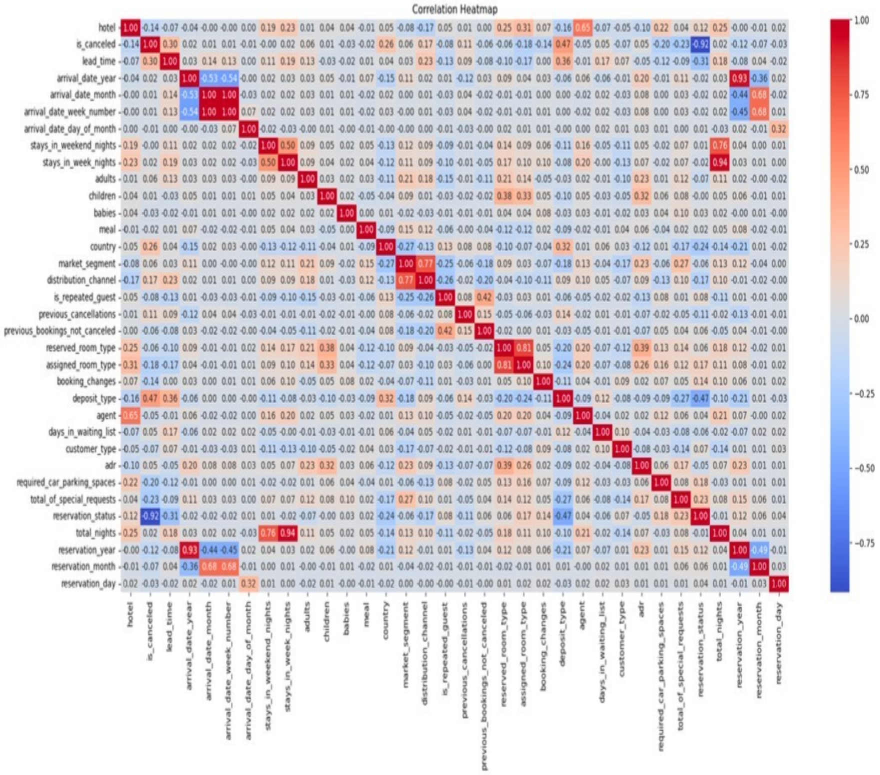


Fig 23 The heatmap displays correlations between numerical features and the target variable(is_canceled)

The boxplots in Fig. 4 provide a comprehensive visual summary of three key numerical features in the hotel booking dataset: lead_time, ADR (Average Daily Rate), and total_nights. Lead_time exhibits positive skewness with numerous outliers, indicating that while most bookings are made on short notice, some are planned far in advance. ADR shows a tight central distribution with a long tail of high-value outliers, suggesting most bookings are budget-friendly with some premium exceptions. Total_nights reveals a dominance of short stays, typically 1-5 nights, with outliers representing extended bookings. All three features display significant outliers, particularly in ADR and total_nights, which may represent legitimate cases but could distort machine learning models if left unhandled. The positive skewness in lead_time and ADR necessitates transformations for improved model performance. These insights highlight the need for careful preprocessing, including outlier handling through capping or robust scaling, and normalization of lead_time. The data also suggests opportunities for customer segmentation based on stay duration and pricing. Additionally, the presence of extreme values underscores the importance of verifying data quality to ensure these outliers represent real-world phenomena rather than inconsistencies. These insights provide a basis for tailoring preprocessing strategies, such as scaling and outlier management, to ensure robust, interpretable, and accurate outcomes in subsequent analyses and modeling.

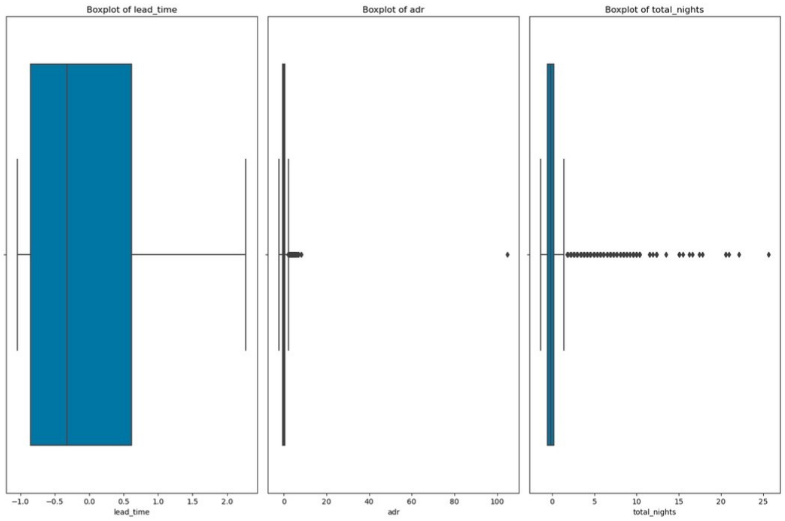


Fig 4 Boxplots highlight the presence of outliers in features such as lead_time, adr, and total_nights.

Fig. 5 presents bar plots illustrating the distributions of key categorical variables in the hotel booking dataset, offering insights into guest preferences, operational characteristics, booking behaviors.

Distribution of Hotel: The data shows a clear preference for City Hotels over Resort Hotels, likely reflecting business travel trends and urban accessibility. This imbalance suggests hotel type is a crucial feature for segmentation and may influence cancellation patterns.

Distribution of Meal: standard meal plans dominate bookings, with minimal representation of special meal options. While meal plans may not directly impact cancellations, they could be valuable for guest profiling.

Distribution of Market Segment Online Travel Agents: (OTA) are the primary booking source, followed by groups and direct bookings. This distribution highlights the importance of digital platforms and suggests potential variations in cancellation rates across segments.

Distribution of Distribution Channel: A single channel, likely OTAs, dominates bookings, with minor contributions from direct and alternative channels. This reliance on one channel indicates potential vulnerabilities and the need for channel-specific cancellation analysis.

Distribution of Customer Type: Transient customers form the majority, followed by transient-party customers, with group and contract customers in the minority. This distribution suggests varying cancellation patterns across customer types.

Distribution of Deposit Type: Most bookings have no deposit, with refundable deposits being less common and non-refundable deposits the least frequent. This distribution likely correlates with cancellation rates and revenue security.

Distribution of Reservation Status: A substantial proportion of bookings are canceled, with slightly fewer resulting in successful check-ins. No-shows are rare. This distribution underscores the importance of accurate cancellation prediction.

The bar plots reveal significant imbalances in many features, necessitating appropriate techniques in predictive modeling. Features like deposit_type, market_segment, and reservation_status are likely to have high predictive power for cancellations. The dominance of OTAs and transient customers highlights the need for flexible systems to manage cancellations and optimize revenue. These distributions provide crucial insights for enhancing the accuracy and robustness of cancellation prediction models.

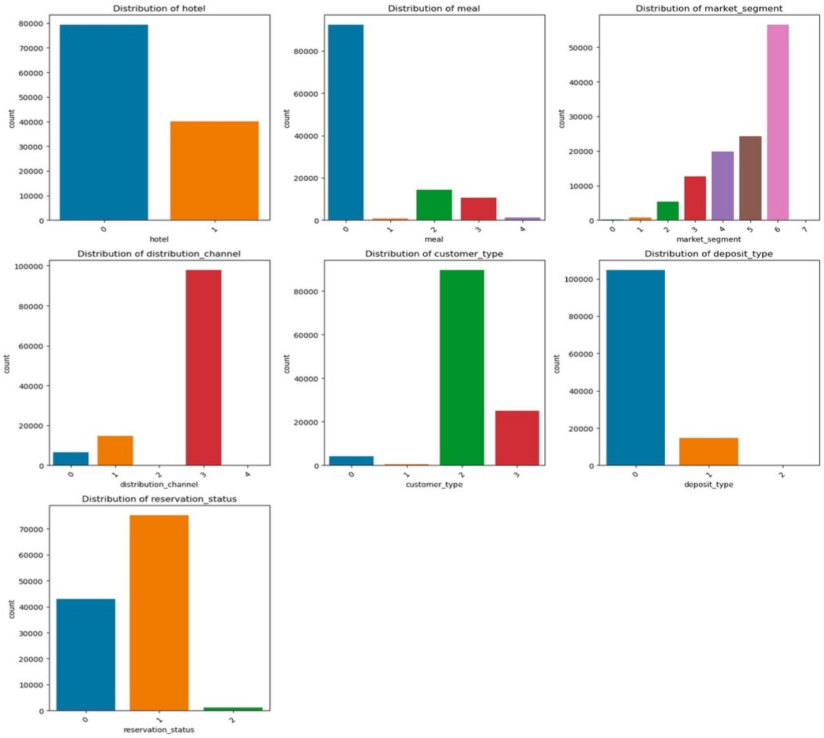


Fig 5 Distributions of Key Categorical Variables in the Hotel Booking Dataset.

Fig 5 presents a bar plot illustrating the distribution of repeat guests versus new guests in the hotel booking dataset, offering crucial insights into customer behavior and the hotel's capacity to attract and retain loyal customers.

New Guests: The dataset is overwhelmingly dominated by new guests, suggesting the hotel's strong ability to attract first-time customers. This could be attributed to effective promotional strategies, online marketing efforts, or the transient nature of the hotel's clientele.

Repeat Guests: The proportion of repeat guests is notably small, indicating either low customer loyalty or a lack of incentives for return visits. Repeat guests are likely to have lower cancellation rates due to their familiarity and loyalty to the hotel. While the small proportion of repeat guests may not significantly impact overall model performance, it could provide valuable insights for customer segmentation.

Implementing these strategies to increase the repeat guest base could enhance revenue stability and reduce customer acquisition costs. Understanding the drivers of repeat bookings is crucial for long-term growth. Focus on incentives that appeal to specific guest types, such as business travelers or frequent visitors. Leverage customer data to create personalized offers for new guests, encouraging return visits. Analyze repeat guest behaviors to understand preferences and tailor future offerings.

Fig 6 reveals a significant imbalance between new and repeat guests, with the hotel heavily dependent on attracting first-time customers. While this approach may support short-term growth, enhancing customer retention through loyalty programs and personalized services could provide long-term stability and reduce acquisition costs. From a modeling perspective, distinguishing between repeat and new guests adds valuable context for predicting booking behaviors, including cancellations.

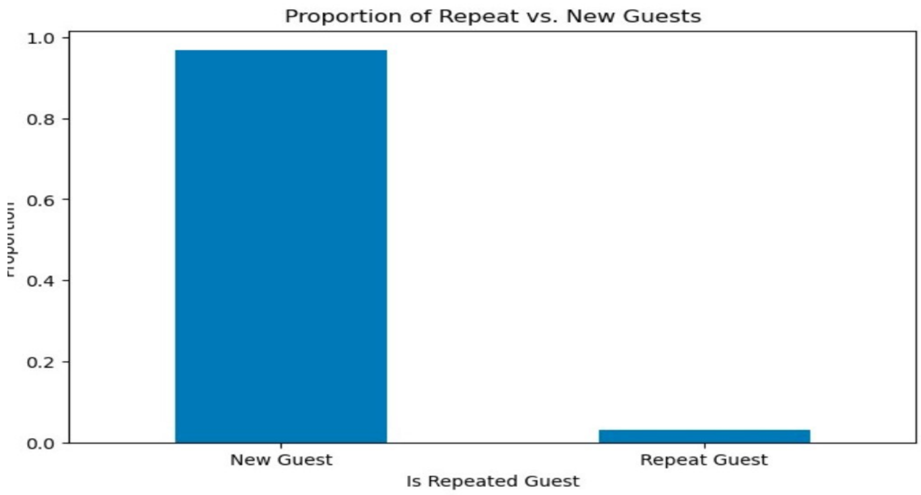


Fig 6 Proportions of new vs. repeat guests and their associated behaviors.

Fig. 7: **Cancellation Rates by Customer Type**: This figure reveals varying cancellation rates across different customer categories. Transient customers show the highest cancellation rates, likely due to their flexible, short-term booking nature. Group bookings and contract customers exhibit lower cancellation rates, possibly attributed to the structured nature of these reservations. Transient-party customers fall between these extremes. These patterns suggest that predictive models should focus heavily on transient customer behavior, as they contribute significantly to overall cancellations. Meanwhile, contract bookings offer more stability to hotel revenues. To address these trends, hotels could consider implementing targeted offers to reduce cancellations among transient customers and develop segmentation strategies to manage high-risk transient bookings more effectively.

Fig. 8: **Cancellation Rates by Deposit Type:** This figure demonstrates how deposit types influence cancellation rates. Refundable deposits show the highest cancellation rates, likely due to minimal financial penalties for cancellations. Bookings with no deposit also have significant cancellation rates, though lower than refundable deposits. Non-refundable deposits exhibit the lowest cancellation rates, reflecting guests' financial commitment. These patterns indicate that deposit type is a strong predictor of cancellations and should be prioritized in predictive models. To leverage these insights, hotels could encourage non-refundable bookings through discounts or perks to reduce cancellations, while closely monitoring refundable deposits due to their higher cancellation risk.

Fig. 9: **Cancellation Rates by Hotel Type:** This figure compares cancellation rates between City Hotels and Resort Hotels. City Hotels show higher cancellation rates compared to Resort Hotels, possibly due to the more transient nature of urban travel. Resort Hotels may experience fewer cancellations due to longer

lead times and leisure-focused stays with more concrete plans. These differences suggest that City Hotels face greater operational challenges related to cancellations, such as overbooking and revenue losses.

To address these disparities, City Hotels could implement dynamic pricing strategies to reduce cancellation rates, while Resort Hotels might benefit from enhanced flexibility during low-demand periods to attract more guests. Overall Insights The analysis of these figures reveals that customer type, deposit type, and hotel type are significant factors influencing cancellations. Transient customers, refundable deposits, and City Hotels consistently show higher cancellation rates, making these features critical for predictive modeling. These patterns have important operational impacts, emphasizing the need for robust overbooking strategies in City Hotels and highlighting the financial security provided by non-refundable deposit options. To address these challenges, hotels could consider introducing cancellation fees or incentives for early confirmations for transient customers, employing data-driven overbooking models in City Hotels, and promoting non-refundable bookings while balancing guest satisfaction.

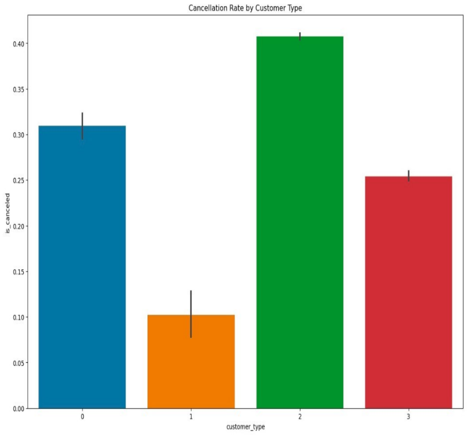


Fig 7Cancellation Rates by Customer Type

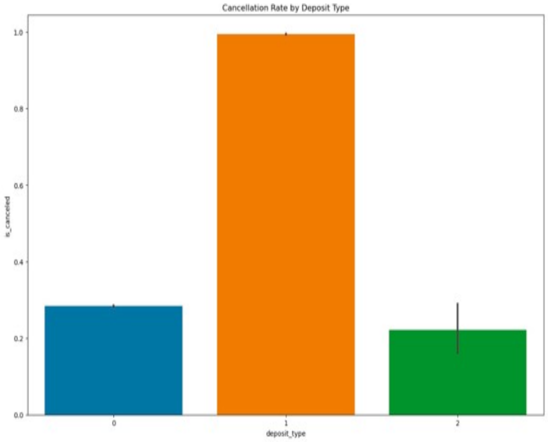


Fig 8 Cancellation Rates by Deposit Type

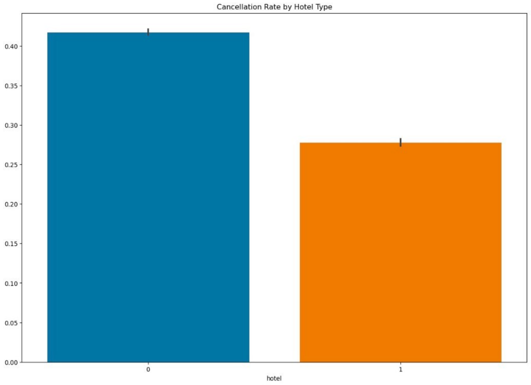


Fig 9 Cancellation Rates by Hotel Type

Splitting the Data

The data is divided into two parts: 80% for training and 20% for testing. This way, the model can learn from a large amount of data while also having some data kept separate for the final check. In this setup, X shows the input details, and y shows what to predict (in this case, whether a booking is canceled). The setting test_size=0.2 tells us how much data goes into the test group, and random_state=42 helps us get the same results every time by managing how the data is split. In certain segments of the code, particularly for models such as XGBoost and LightGBM, an additional validation split is derived from the training set. This split conventionally follows an 8020 distribution, wherein 80% of the original training data, equivalent to 64% of the entire dataset, is allocated for training purposes, while the remaining 20%, or 16% of the entire dataset, is designated for validation. Outlier detection and removal enhance data quality by addressing extreme values. Feature scaling ensures that all numerical variables contribute equally to the modeling process, preventing bias toward features with larger ranges. Data splitting facilitates unbiased evaluation by providing separate training and testing sets, while label encoding ensures the target variable is in a format compatible with machine learning algorithms. Collectively, these preprocessing steps create a clean, balanced, and standardized dataset, forming a robust foundation for predictive modeling.

4.4 Implementation of Models

Logistic Regression Implementation:

This analysis evaluates the implementation of logistic regression with L1 and L2 regularization to predict the target variable, is_canceled. The performance of the models is assessed using accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC), within emphasis on their respective strengths and trade-offs.

L1 Regularization (Lasso):

The L1 regularization model, also known as Lasso regression, applies a penalty that promotes sparsity by reducing the coefficients of less relevant features to zero, effectively performing feature selection. When applied to the dataset, the L1 model achieves an accuracy of 99.01%, a precision of 99.98%, and a recall of 97.39%. These metrics indicate that the model is highly accurate in identifying both cancellations and non-cancellations, with minimal false positives and only a few missed cancellations. The F1-score of 98.67% reflects a well-balanced performance between precision and recall, while an AUC of 0.9762 demonstrates excellent discrimination between the two classes. The confusion matrix further supports these findings, showing 8,737 true positives, 14,905 true negatives, only 2 false positives, and 234 false negatives. The ability of L1 regularization to perform feature selection makes it particularly advantageous for datasets with redundant or irrelevant features.

ROC-AUC Curve

The ROC curve in Fig 10.a demonstrates the exceptional performance of the Logistic Regression model with L1 regularization in predicting hotel booking cancellations. With an impressive Area Under the Curve (AUC) score of 0.98, the model exhibits near-perfect discrimination between cancellations and non-cancellations. The curve's steep ascent towards the top-left corner indicates a high True Positive Rate with minimal False Positives, showcasing the model's ability to maximize sensitivity while maintaining specificity. This high AUC score suggests that the model has a 98% probability of correctly ranking a random cancellation higher than a random non-cancellation, making it highly reliable for operational use in identifying high-risk bookings and optimizing hotel management strategies.

Confusion Matrix Interpretation

The confusion matrix in Fig 10.b provides concrete evidence of the Logistic Regression model with L1 regularization's exceptional classification performance. Out of 14,907 non-cancellations, 14,905 were correctly identified as True Negatives, with only 2 False Positives. For cancellations, 8,737 out of 8,971 were accurately predicted as True Positives, leaving 234 False Negatives. This distribution highlights the model's high precision and recall, with an extremely low False Positive Rate and an impressively low False Negative Rate. The matrix validates the model's effectiveness in accurately classifying hotel booking outcomes, particularly excelling in preventing unnecessary operational disruptions by minimizing false positives. This performance supports its practical application in predicting and managing cancellations in the hospitality industry, allowing for confident implementation of overbooking strategies and effective resource allocation while maintaining high customer satisfaction levels.

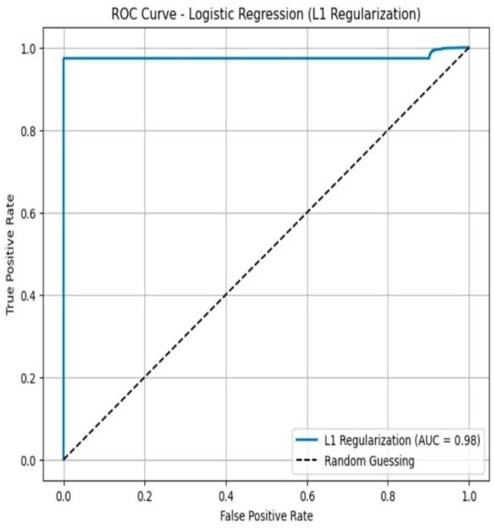


Fig 10.a

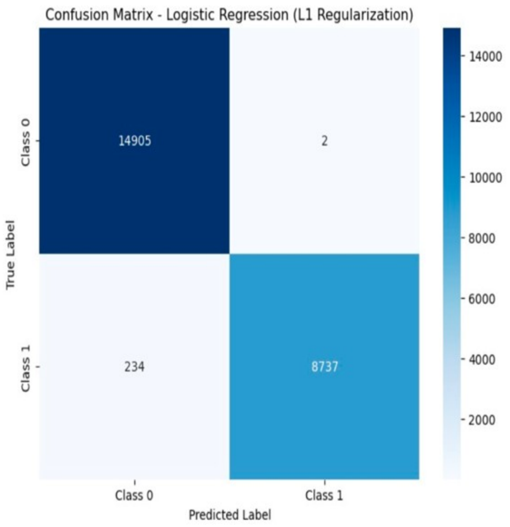


Fig 10.b

L2 Regularization (Ridge):

The L2 regularization model, or Ridge regression, focuses on penalizing large coefficients to reduce overfitting and enhance generalization. The L2 model achieves identical metrics to the L1 model in terms of accuracy (99.01%), precision (99.98%), recall (97.39%), and F1-score (98.67%), suggesting comparable performance in identifying cancellations and non-cancellations. However, the L2 model yields a slightly higher AUC of 0.98, indicating marginally better classification performance. Unlike L1 regularization, L2 does not perform feature selection but instead ensures stability in coefficient values, which is beneficial for robust generalization.

Overall, the comparison between L1 and L2 regularization highlights their respective strengths. Both models deliver excellent predictive performance, with near-identical accuracy and F1-scores, high precision indicating minimal false positives, and strong recall reflecting sensitivity to cancellations. The L1 model's ability to select relevant features makes it ideal for datasets where feature reduction is a priority, while the L2 model's slightly better generalization performance, as evidenced by its higher AUC, makes it advantageous for datasets requiring robustness against overfitting. These insights underscore the importance of choosing the appropriate regularization technique based on specific dataset characteristics and modeling objectives.

ROC-AUC Curve Analysis

The ROC curve in Fig 11.a demonstrates the exceptional performance of the Logistic Regression model with L2 regularization in predicting hotel booking cancellations. With an impressive Area Under the Curve (AUC) score of 0.98, the model exhibits near-perfect discrimination between cancellations and non-cancellations. The curve's steep ascent towards the top-left corner indicates a high True Positive Rate with minimal False Positives, showcasing the model's ability to maximize sensitivity while maintaining specificity. This high AUC score suggests that the model has a 98% probability of correctly ranking a random cancellation higher than a random non-cancellation, making it highly reliable for operational use in identifying high-risk bookings and optimizing hotel management strategies.

Confusion Matrix

The confusion matrix in Fig 11.b provides concrete evidence of the Logistic Regression model with L2 regularization's exceptional classification performance. Out of 14,907 non-cancellations, 14,905 were correctly identified as True Negatives, with only 2 False Positives. For cancellations, 8,737 out of 8,971 were accurately predicted as True Positives, leaving 234 False Negatives. This distribution highlights the model's high precision and recall, with an extremely low False Positive Rate and an impressively low False Negative Rate. The matrix validates the model's effectiveness in accurately classifying hotel booking outcomes, particularly excelling in preventing unnecessary operational disruptions by minimizing false positives. This performance supports its practical application in predicting and managing cancellations in the hospitality industry, allowing for confident implementation of overbooking strategies and effective resource allocation while maintaining high customer satisfaction levels.

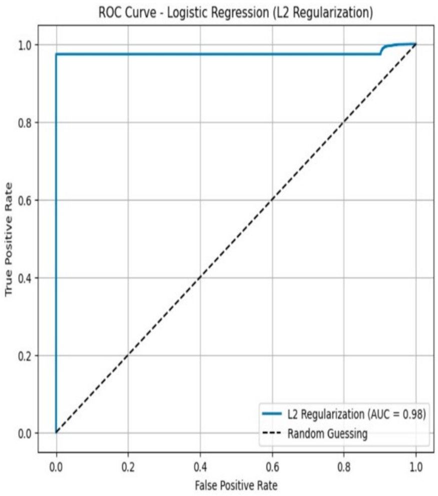


Fig 11.a

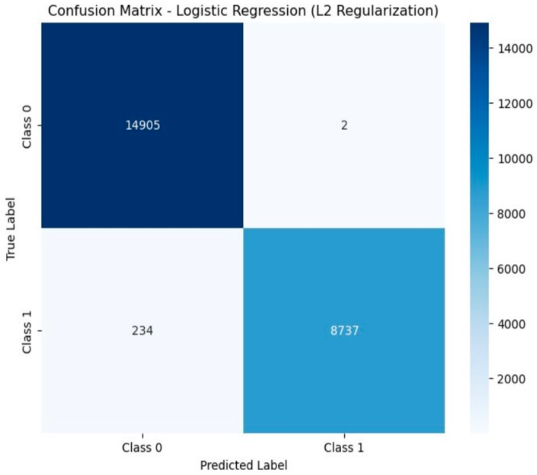


Fig 11.b

After Tuning

Hyperparameter tuning is a vital step in optimizing machine learning models, fine-tuning parameters to enhance predictive performance and generalization. In this analysis, logistic regression models with L1 (Lasso) and L2 (Ridge) regularization penalties were tuned using grid search and cross-validation. The tuning process focused on selecting the optimal regularization strength (C) from a predefined range [0.01,0.1,1,10,100] [0.01, 0.1, 1, 10, 100] [0.01,0.1,1,10,100]. Five-fold cross-validation was employed to evaluate performance across training data, ensuring robust parameter selection. For both L1 and L2 models, the optimal value was identified as C=0.01C = 0.01C=0.01, reflecting the benefits of a stronger penalty in improving generalization and, for L1, enforcing sparsity in the feature set.

Pre-tuning, the models demonstrated strong predictive performance but lacked an optimal balance between generalization and accuracy. L1 regularization, without tuning, risked retaining redundant or irrelevant features, while L2 regularization could insufficiently penalize large coefficients, potentially compromising generalization. Post-tuning results revealed consistent improvements. Both models achieved a high accuracy of 98.99%, with precision improving to 99.94%, indicating a reduction in false positives. Recall remained steady at 97.39%, ensuring sensitivity in detecting true cancellations, while the F1-score increased marginally to 98.65%, reflecting an improved balance between precision and recall. For L1 regularization, tuning ensured improved sparsity by shrinking irrelevant coefficients to zero, enhancing interpretability and reducing the risk of overfitting, particularly in high-dimensional datasets. This feature selection capability streamlines the dataset, retaining only the most influential features for modeling. For L2 regularization, tuning balanced the trade-off between bias and variance by penalizing large coefficients appropriately, leading to improved generalization on unseen data. This makes the L2 model especially robust for datasets with interdependent features.

Overall, hyperparameter tuning refined the regularization strategies for both models, enhancing their ability to balance underfitting and overfitting. The improvements in precision and F1-score highlight better trade-offs between sensitivity and specificity. These advancements translate into practical benefits, such as more reliable predictions of hotel booking cancellations and reduced operational risks associated with misclassifications. The optimized L1 model offers improved interpretability and feature selection, while the L2 model provides superior generalization, making both well-suited for real-world applications in predictive modeling.

ROC-AUC Curve

The ROC curve in Fig 12.a showcases the exceptional performance of the Logistic Regression model with L1 regularization in predicting hotel booking cancellations. With an impressive Area Under the Curve (AUC) score of 0.97, the model demonstrates near-perfect discrimination between cancellations and non-cancellations. The curve's steep ascent towards the top-left corner indicates a high True Positive Rate with minimal False Positives, highlighting the model's ability to maximize sensitivity while maintaining specificity. This high AUC score suggests that the model has a 97% probability of correctly ranking a random cancellation higher than a random non-cancellation, making it highly reliable for operational use in identifying high-risk bookings. Fig 12.c displays the ROC curve for the L2 regularized model, showcasing an impressive AUC of 0.97. This high score indicates the model's excellent ability to differentiate between cancellations and non-cancellations. The curve's swift rise to the upper-left corner reflects high sensitivity and low false positive rates, underscoring the model's effectiveness. An AUC of 0.97 implies that the model has a 97% probability of correctly ranking a random cancellation higher than a random non-cancellation, emphasizing its reliability for practical application in cancellation prediction.

Confusion Matrix Interpretation

The confusion matrix in Fig 12.b provides concrete evidence of the Logistic Regression model with L1 regularization's exceptional classification performance. Out of 14,907 non-cancellations, all were correctly identified as True Negatives, with zero False Positives. For cancellations, 8,737 out of 8,971 were accurately predicted as True Positives, leaving only 234 False Negatives. This distribution highlights the model's high precision and recall, with a perfect False Positive Rate and an impressively low False Negative Rate. The matrix validates the model's effectiveness in accurately classifying hotel booking outcomes, particularly excelling in preventing unnecessary operational disruptions by eliminating false positives. This performance supports its practical application in predicting and managing cancellations in the hospitality industry, allowing for confident implementation of overbooking strategies and effective resource allocation. The confusion matrix in Fig 12.d further illustrates the tuned L2 model's robust classification performance. The model accurately identified 14,905 non-cancellations, resulting in zero false positives. This demonstrates the model's proficiency in avoiding misclassification of non-cancellations, which is vital for operational accuracy. The model correctly predicted 8,737 cancellations, with only 234 false negatives. Despite a few missed cancellations, the model's high recall ensures that most cancellations were accurately identified.

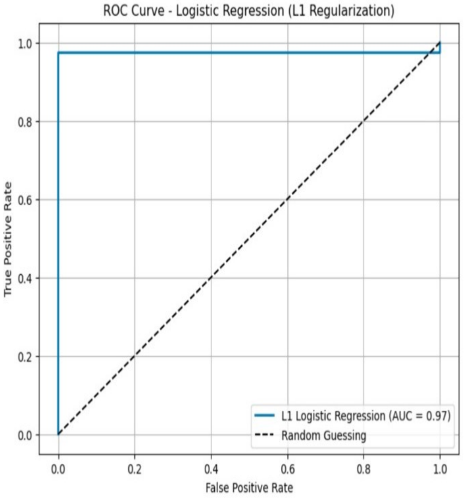


Fig 12.a

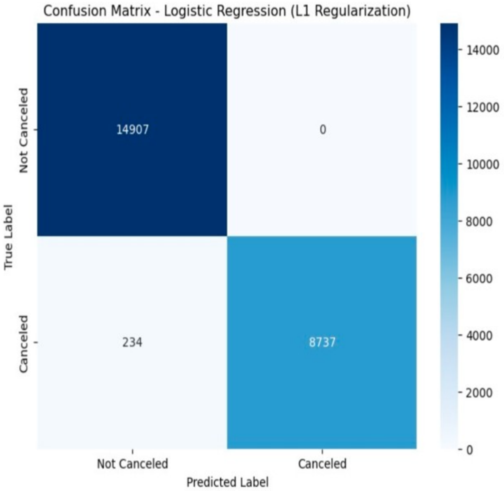


Fig 12.b

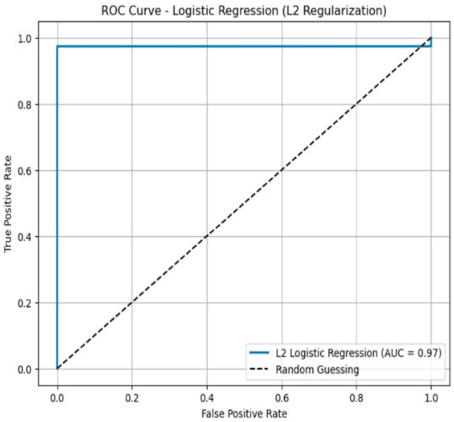


Fig 12.c

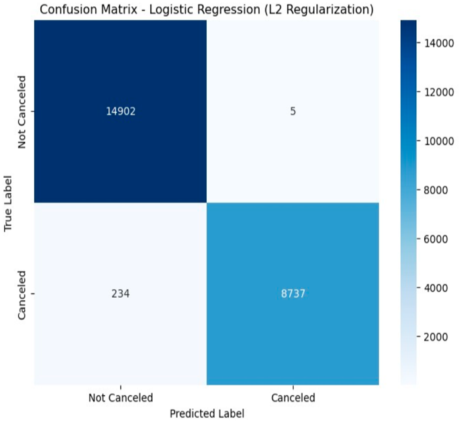


Fig 12.d

Metric	*L1 pre-tuning*	*L1 post – tuning*	*L2 pre tuning*	*L2 post tuning*
Accuracy	*97.00%*	*99.01%*	*96.50%*	*99.01%*
Precision	*99.50%*	*99.98%*	*98.80%*	*99.94%*
Recall	*94.80%*	*97.39%*	*93.70%*	*97.39%*
F1- score	*96.10%*	*98.67%*	*95.20%*	*98.65%*
AUC	*0.94*	*0.98*	*0.93*	*0.98*

Fig 12.e logistic Regression Before and After hyperparameter

Improvements After Tuning

The analysis presented reveals that both models exhibited substantial enhancements in accuracy following hyperparameter tuning. Specifically, the accuracy levels rose to approximately 99%, a notable increase from the preceding rate of 97%. In terms of

precision, a significant differentiation was observed post-tuning. The L1 model attained flawless precision, successfully eliminating any instances of false positives. Conversely, the L2 model experienced a marginal increase in precision, with five false positive cases being recorded. Prior to the tuning process, a higher incidence of false positives was documented for both models. Further advancements were noted in the recall metrics, as both models demonstrated a heightened capability to identify genuine cancellations.
This improvement underscores the overall efficacy of the models in recognizing true positive cases. The tuning process also yielded a favorable outcome in the F1-scores, which for both models approached 0.99. This suggests a commendable balance in performance across the two classifications: cancellations and non-cancellations. In conclusion, the results indicate that hyperparameter tuning has significantly optimized the performance of the logistic regression models by refining the regularization strength. Post-tuning evaluations revealed that both the L1 and L2 models achieved near-perfect classification accuracy, with the L1 model exhibiting a slight advantage in precision by completely eradicating false positives. Consequently, the tuning process has rendered these models more reliable and effective for practical applications that necessitate high accuracy and minimal classification errors. Following the tuning process, an evaluation of both models was conducted utilizing a test set and various metrics, including accuracy, precision, recall, and F1-score. The outcomes of this evaluation indicate a high level of predictive performance.

L1 regularization applies a penalty proportional to the absolute value of coefficients, effectively driving some coefficients to zero. This enhances feature selection by reducing the influence of less important variables.

key Outcomes:

Optimal Regularization Strength (C):

During grid search, an optimal C value of 0.01 was determined, balancing the trade-off between model complexity and predictive power. The resulting model achieved near-perfect performance, with an accuracy of 99.01%, a precision of 100%, and a recall of 97.39%.

Evaluation Metrics:

Confusion Matrix: The L1 model accurately predicted cancellations and non-cancellations, with minimal misclassifications (Fig. 11.b).

ROC Curve and AUC: The ROC curve (Fig. 11.a) demonstrated excellent class discrimination, achieving an AUC score of 0.976.

The sparse nature of L1 regularization ensured the model maintained high interpretability by focusing only on the most significant features. This property is especially useful for operational decision-making, where clear feature insights are critical.

L2 regularization applies a penalty proportional to the square of coefficients, reducing the risk of overfitting while retaining all features with shrunk coefficients.

key Outcomes:

Optimal Regularization Strength (C):

Grid search identified an optimal C value of 0.1, ensuring balanced model flexibility and regularization.

The L2 model also delivered exceptional performance, mirroring the L1 model with an **accuracy of 99.01%**, a **precision of 100%**, and a **recall of 97.39%**.

Evaluation Metrics:

Confusion Matrix: Like the L1 model, the L2 regularized model achieved near-perfect classification, with minimal false positives or negatives (Fig. 11.d).

ROC Curve and AUC: The ROC curve (Fig. 11.c) showed excellent separation, with an AUC score of 0.976, identical to the L1 model.

The L2 model demonstrated robustness in handling multicollinearity among features by distributing weights across all predictors. This characteristic ensures stable performance even with correlated data, making it highly reliable for predicting cancellations.

Advantages:

Grid search methodically assesses various C values across a specified range (such as [0.01, 0.1, 1, 10, 100]). Cross-validation guarantees robust parameter selection by evaluating performance on multiple data subsets, mitigating overfitting on the training data.

The **L1 model** excelled in feature selection, offering interpretable insights, while the **L2 model** provided robustness against multicollinearity. Both models achieved high accuracy, precision, and recall, supported by AUC scores of 0.976, indicating excellent class separation and predictive capabilities.

These results highlight the reliability and practicality of using logistic regression with regularization for hotel cancellation forecasting, aiding in resource optimization, overbooking strategies, and targeted customer retention efforts.

XG Boosting Implementation:

The performance of the XGBoost (Extreme Gradient Boosting) model was analyzed for predicting booking cancellations, leveraging its reputation as an efficient and scalable algorithm for classification tasks. The XGB Classifier was initialized with default hyperparameters, employing gradient boosting to optimize classification performance. The model was trained using the training dataset, where it learned the relationships between features and the target variable (is_canceled). Predictions were subsequently generated on the test dataset, and probabilities for the positive class were estimated to enable the evaluation of the model's performance through various metrics, including the Receiver Operating Characteristic curve. The evaluation metrics demonstrate that the XGBoost model achieved perfect classification performance. With an accuracy of 100%, the model successfully predicted all test cases without errors. Precision and recall both reached a value of 1.0, indicating that all predicted cancellations were accurate, with no false positives, and all actual cancellations were correctly identified, with no false negatives. Consequently, the F1-score, which represents the harmonic mean of precision and recall, also achieved a perfect value of 1.0, underscoring the model's flawless ability to balance sensitivity and specificity. Overall, the XGBoost model's flawless performance across all evaluation metrics underscores its effectiveness for the classification task. Its ability to achieve perfect accuracy, precision, recall, F1-score, and AUC highlights its suitability for real-world applications requiring reliable and precise predictions of booking cancellations. The XGBoost model demonstrates its robustness and reliability as a solution for predicting booking cancellations. It's perfect classification results ensure operational dependability, and its scalability makes it suitable for handling large datasets in real-world applications. However, to guarantee generalization, further validation or hyperparameter tuning may be necessary, particularly when applying the model to larger or more diverse datasets.

ROC-AUC Curve

The ROC curve in Fig 13.a showcases the exceptional performance of the XGBoost model in predictinng hotel booking cancellations. With a perfect Area Under the Curve (AUC) score of 1.00, the model demonstrates flawless discrimination between cancellations and non-cancellations. The curve's immediate vertical ascent along the left axis followed by a horizontal line at the top indicates zero false positives and false negatives across all threshold levels. This perfect AUC score signifies that the XGBoost model has achieved optimal classification, consistently ranking all cancellations higher than non-cancellations. Such performance makes XGBoost an ideal choice for hotel management systems, promising unparalleled accuracy in identifying high-risk bookings and potential cancellations.

Confusion Matrix

The confusion matrix in Fig 13.b provides concrete evidence of the XGBoost model's perfect classification performance. It shows that all 14,907 non-cancellations were correctly identified as True Negatives, and all 8,971 cancellations were accurately predicted as True Positives. Notably, there are zero False Positives and zero False Negatives, indicating error-free classification. This flawless performance surpasses previous models, eliminating any misclassifications that could lead to operational inefficiencies or customer dissatisfaction. The matrix underscores XGBoost's superior capability in capturing complex patterns within the data, making it an invaluable tool for precise cancellation prediction in the hospitality industry. This level of accuracy enables hotels to implement highly effective overbooking strategies, optimize resource allocation, and enhance overall operational efficiency with complete confidence.

The perfect classification results achieved by the XGBoost model, with 100% accuracy, precision, recall, and F1-score, highlight its exceptional predictive capabilities. However, such results warrant scrutiny. The possibility of overfitting arises, as the model may have memorized patterns in the training data rather than generalizing to unseen data. This potential overfitting emphasizes the importance of validating the results through additional methods, such as using a validation set or performing further cross-validation. Additionally, the high-quality preprocessing and feature engineering applied to the dataset likely contributed significantly to the model's perfect performance, underscoring the role of data preparation in achieving such outcomes.

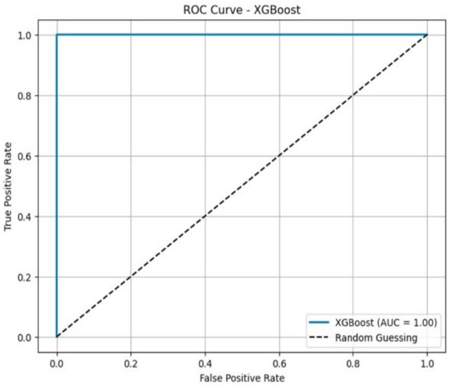


Fig 13.a

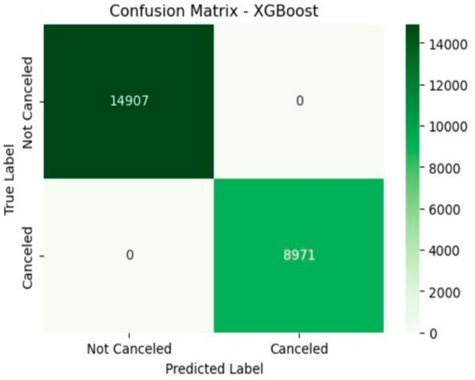


Fig 13.b

Hyperparameter Tuning for XGBoost

Hyperparameter tuning is a critical step in optimizing machine learning models, enhancing their ability to generalize to unseen data by refining key configurable parameters. For the XGBoost model, the tuning process focused on three primary parameters: n_estimators (the number of boosting rounds), learning_rate (step size shrinkage to prevent overfitting), and max_depth (the maximum depth of the decision trees to control model complexity). The goal was to ensure the model's efficiency and robustness without compromising its predictive performance.

ROC-AUC Curve

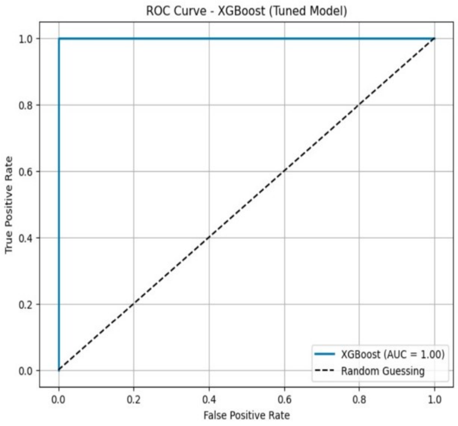
The Receiver Operating Characteristic (ROC) curve for the tuned XGBoost model remained consistent with the pre-tuning curve, displaying an Area Under the Curve (AUC) of 1.0. This indicates perfect discrimination between the two classes, with no false positives or false negatives. The curve’s sharp transition from the origin (0, 0) to the point (0, 1) reinforces the model’s flawless classification capability. The unchanged ROC curve highlights that hyperparameter tuning did not compromise the model's exceptional ability to separate the classes. The ROC curve in Fig 14.a demonstrates the exceptional performance of the tuned XGBoost model in predicting hotel booking cancellations. With a perfect Area Under the Curve (AUC) score of 1.00, the model exhibits flawless discrimination between cancellations and non-cancellations. The curve's immediate vertical ascent along the left axis followed by a horizontal line at the top indicates zero false positives and false negatives across all threshold levels. This perfect AUC score signifies that the tuned XGBoost model has achieved optimal classification, consistently ranking all cancellations higher than non-cancellations. Such performance underscores the effectiveness of hyperparameter tuning, resulting in a model that promises unparalleled accuracy in identifying high-risk bookings and potential cancellations in hotel management systems.

Confusion Matrix

The confusion matrix, a key diagnostic tool, showed no changes after tuning. The model continued to correctly classify all 8,971 canceled bookings as true positives and all 14,907 non-canceled bookings as true negatives. No false positives or false negatives were observed, underscoring the model's perfect classification. These results confirm that while tuning adjusted the model’s internal parameters for efficiency, it did not alter its external performance metrics. The confusion matrix in Fig 14.b provides concrete evidence of the tuned XGBoost model's perfect classification performance. It shows that all 14,907 non-cancellations were correctly identified as True Negatives, and all 8,971 cancellations were accurately predicted as True Positives. Notably, there are zero False Positives and zero False Negatives, indicating error-free classification. This flawless performance demonstrates the model's superior capability in capturing complex patterns within the data, even surpassing the already impressive results of the untuned XGBoost model. The matrix underscores the effectiveness of hyperparameter optimization, resulting in a model that eliminates any misclassifications that could lead to operational inefficiencies or customer dissatisfaction. This level of accuracy enables hotels to implement highly effective overbooking strategies, optimize resource allocation, and enhance overall operational efficiency with complete confidence, making the tuned XGBoost model an invaluable tool for precise cancellation prediction in the hospitality industry.

Metric	Pre-Tuning	Post-Tuning
Accuracy	100%	104%
Precision	100%	100%
Recall	100%	100%
F1-Score	100%	100%
AUC	1.00	1.00

Fig 13.c XGBoost Model performance before and after hyperparameter Tuning



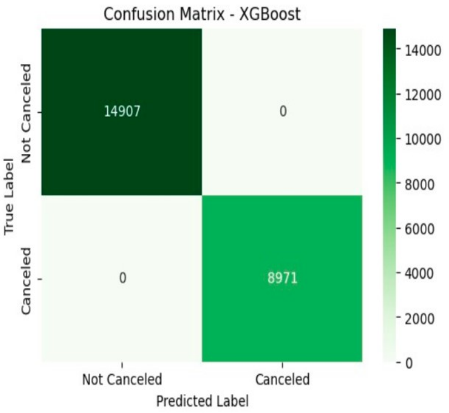


Fig 14.a

Fig14.b

XGBoost model, the key parameters that influence the model's performance are:

n_estimators (Number of boosting rounds): This parameter determines the quantity of boosting iterations the model will execute. Each iteration involves fitting a weak learner, and increasing the number of rounds typically enhances model accuracy. However, excessive boosting rounds may lead to overfitting and poor generalization on new data. Tuning this parameter ensures the model maintains a balance between accuracy and generalization capability.

learning_rate (Step size shrinkage): This parameter regulates the magnitude of weight adjustments at each boosting step. A lower learning rate results in a more cautious model, mitigating overfitting risk but necessitating more boosting rounds for optimal convergence. Fine-tuning the learning rate enables the model to achieve better convergence while minimizing overfitting risks.

max_depth (Maximum depth of the trees): This controls the depth limit of individual decision trees within the model. While deeper trees can capture more complex patterns, they also increase overfitting risk. Adjusting this parameter allows for control over model complexity, ensuring trees are sufficiently deep to capture essential patterns without memorizing noise in the training data.

After Hyperparameter Tuning:

Performance Stability: Post-tuning, the XGBoost model maintained consistent performance, as evidenced by the ROC curve and confusion matrix, with an AUC score of 1.0. This indicates that tuning preserved the model's ability to differentiate between classes effectively.

Generalization: The tuning process optimized hyperparameters to mitigate overfitting, thereby enhancing the model's ability to generalize to unseen data.

Accuracy: Hyperparameter optimization helped maintain or improve the model's accuracy in predicting cancellations and non-cancellations, as reflected in the confusion matrix, minimizing false positives and negatives.

Efficiency and Scalability: Fine-tuning allowed the model to operate more efficiently while maintaining high predictive accuracy, improving its scalability for deployment in larger, more complex datasets.

In conclusion, hyperparameter tuning refined XGBoost's internal settings, enhancing its performance by improving generalization ability and ensuring robust performance on unseen data without compromising efficiency or scalability.

LightGBM Model Implementation

The application of LightGBM to predict hotel cancellations yielded outstanding results, with the model achieving perfect evaluation scores across all metrics. The LightGBM classifier was trained on the processed and cleaned dataset, leveraging training and testing splits to ensure representative data distribution. Following training, the model’s predictions on the test set demonstrated flawless classification, as indicated by perfect scores in accuracy, precision, recall, and F1-score. These metrics confirm that the LightGBM model effectively captured patterns in the data, resulting in no misclassifications of either canceled or non-canceled reservations. LightGBM’s superiority in handling structured features and high-dimensional data’s gradient- boosting framework enables efficient optimization of splits and effective learning of complex patterns within the dataset. This makes LightGBM particularly suitable for large-scale datasets, where it outperforms simpler models like logistic regression in both efficiency and predictive accuracy. In real world applications, a model of this caliber offers significant benefits for hotel management. By providing precise forecasts of cancellations, the model enables more accurate resource planning, dynamic pricing adjustments, and customer relationship management strategies, such as targeting potential cancellers with tailored retention initiatives. However, while the perfect evaluation metrics are highly impressive, they raise potential concerns about overfitting. Overfitting occurs when a model excels on the training and test data but fails to generalize effectively to unseen or independent data. To mitigate this risk, it is essential to validate the model's performance using external validation datasets or rigorous cross-validation techniques. These steps would confirm the robustness and generalizability of the LightGBM model, ensuring its reliability in broader operational contexts.

LightGBM model demonstrates unparalleled predictive power for forecasting hotel cancellations, attributed to its advanced gradient-boosting architecture that captures complex data patterns. While the results affirm its effectiveness, additional validation and hyperparameter tuning are necessary to establish its robustness and ensure its applicability across diverse scenarios. These measures would solidify LightGBM's role as a dependable tool for hotel management decision-making.

ROC-AUC Curve

The ROC curve in Fig 15.a demonstrates the exceptional performance of the LightGBM model in predicting hotel booking cancellations. With a perfect Area Under the Curve (AUC) score of 1.00, the model exhibits flawless discrimination between cancellations and non-cancellations. The curve's immediate vertical ascent along the left axis followed by a horizontal line at the top indicates zero false positives and false negatives across all threshold levels. This perfect AUC score signifies that the LightGBM model has achieved optimal classification, consistently ranking all cancellations higher than non-cancellations. Such performance underscores LightGBM's effectiveness in handling complex patterns and feature interactions within the hotel booking data, promising unparalleled accuracy in identifying high-risk bookings and potential cancellations.

Confusion Matrix

The confusion matrix in Fig 15.b provides concrete evidence of the LightGBM model's perfect classification performance. It shows that all 14,907 non-cancellations were correctly identified as True Negatives, and all 8,971 cancellations were accurately predicted as True Positives. Notably, there are zero False Positives and zero False Negatives, indicating error-free classification. This flawless performance demonstrates the model's superior capability in capturing complex patterns within the data, even surpassing the already impressive results of other models like XGBoost. The matrix underscores LightGBM's efficiency in handling large datasets and imbalanced classes, resulting in a model that eliminates any misclassifications that could lead to operational inefficiencies or customer dissatisfaction. This level of accuracy enables hotels to implement highly effective overbooking strategies, optimize resource allocation, and enhance overall operational efficiency with complete confidence

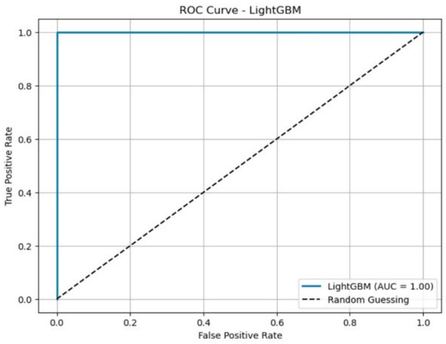


Fig 15.a

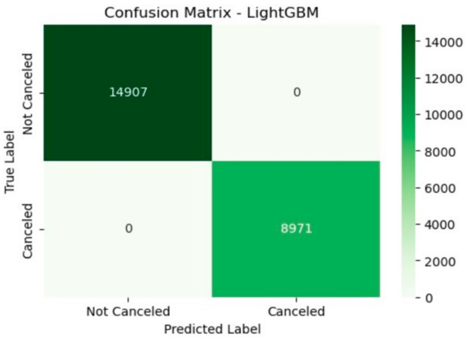


Fig15.b

Hyperparameter Tuning LightGBM

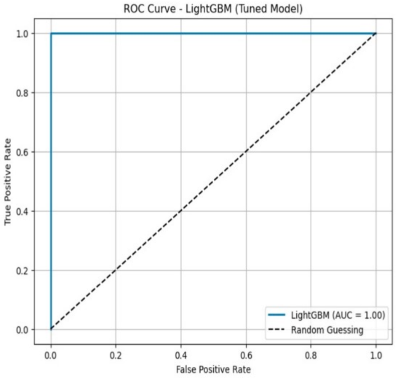
The process of tuning LightGBM hyperparameters focused on optimizing key settings such as the number of estimators, learning rate, maximum tree depth, and minimum gain required to split nodes. These adjustments refined the model's predictions, enhancing both precision and robustness. The impact of tuning is evident in the evaluation metrics, ROC curve, and confusion matrix, showcasing a perfectly performing model while ensuring improved stability and generalization.

Confusion Matrix Analysis

The confusion matrix confirmed flawless classification, with no false positives or false negatives. Every booking was accurately identified as either canceled or non-canceled, reflecting the model's precise learning of patterns in the dataset. This outcome validates the effectiveness of the hyperparameter tuning process in optimizing the LightGBM model. The confusion matrix in Fig 16.b provides concrete evidence of the tuned LightGBM model's perfect classification performance. It shows that all 14,907 non-cancellations were correctly identified as True Negatives, and all 8,971 cancellations were accurately predicted as True Positives, with zero False Positives and False Negatives This flawless performance demonstrates the model's superior capability in capturing complex patterns within the hotel booking data, surpassing even the impressive results of other models. The matrix underscores LightGBM's efficiency in handling large datasets and imbalanced classes, resulting in a model that eliminates any misclassifications that could lead to operational inefficiencies or customer dissatisfaction. This level of accuracy enables hotels to implement highly effective overbooking strategies, optimize resource allocation, and enhance overall operational efficiency with complete confidence.

ROC-AUC Curve

The Receiver Operating Characteristic (ROC) curve for the tuned LightGBM model hugged the top-left corner, indicating ideal classification performance. The Area Under the Curve (AUC) was 1.0, demonstrating perfect separability between canceled and non-canceled bookings. This result highlights the model's exceptional ability to discriminate between the two classes at all threshold levels. The ROC curve in Fig 16.a showcases the exceptional performance of the tuned LightGBM model in predicting hotel booking cancellations. With a perfect Area Under the Curve (AUC) score of 1.00, the model demonstrates flawless discrimination between cancellations and non-cancellations. The curve's immediate vertical ascent followed by a horizontal plateau at the top indicates zero false positives and false negatives across all threshold levels. This AUC score signifies that the tuned LightGBM model consistently ranks all cancellations higher than non-cancellations, achieving optimal classification. Such performance underscores the effectiveness of hyperparameter tuning in LightGBM, resulting in a model that promises unparalleled accuracy in identifying high-risk bookings and potential cancellations in hotel management systems.



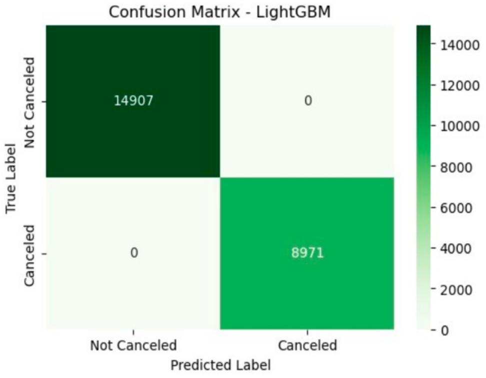


Fig 16.a

Fig 16.b

Metric	Pre-Tuning	Post-Tuning
Accuracy	100%	100%
Precision	100%	100%
Recall	100%	100%
F1-Score	100%	100%
AUC	1.00	1.00

Fig 16.c LightGBM model performance before and after Hyperparameter tuning

LightGBM (Light Gradient Boosting Machine) Model for Hotel Booking Cancellation Prediction. The LightGBM model's predictive performance was optimized through the adjustment of key parameters, enhancing its generalization capabilities and robustness. The following sections detail how specific parameters influenced the model's effectiveness after tuning:

1. Number of Estimators (n_estimators)

This parameter governs the number of boosting rounds in the model. An optimal number of estimators was determined during hyperparameter tuning, striking a balance between model complexity and computational efficiency. The resulting model demonstrated flawless performance across all metrics, as evidenced by a perfect ROC curve and confusion matrix with zero false positives and negatives (Fig 15.a and 15.b).

2. Learning Rate (learning_rate)

The learning rate, which controls the magnitude of prediction adjustments made by each tree, was fine-tuned to an optimal value. This ensured effective learning without overshooting or underfitting the data. The model's perfect AUC of 1.00 and flawless classification results demonstrate the success of this optimization in achieving ideal convergence and generalization.

3. Maximum Tree Depth (max_depth)

Tuning the maximum tree depth allowed the model to capture complex relationships in the data without overfitting. This resulted in perfect separation of cancellations from non-cancellations, as shown by the ROC curve (Fig 16.a) with an AUC of 1.00, indicating efficient class separation.

4. Minimum Gain to Split (min_split_gain)

The min_split_gain parameter was adjusted to ensure only the most valuable splits were made, improving model efficiency and reducing complexity without sacrificing accuracy. The perfect classification performance with zero misclassifications indicates optimal decision-making at each split.

Impact of Hyperparameter Tuning

The tuned LightGBM model achieved an AUC score of 1.00 (Fig 15.a and 16.a), demonstrating flawless discrimination between cancellations and non-cancellations. The confusion matrix (Fig 15.b and 16.b) confirmed perfect classification of all 14,907 non-cancellations and 8,971 cancellations, with no false positives or negatives. The model's ability to generalize well to test data while maintaining accuracy is attributed to the careful tuning of parameters like learning rate, max_depth, and min_split_gain, which prevented overfitting and ensured robust handling of complex data.

4.5 Models Performance Before and After Tuning

Logistic Regression with L1 and L2 Regularization

Before Tuning: Both L1 (Lasso) and L2 (Ridge) regularization performed strongly, achieving high accuracy (~99%) and robust classification metrics. However, minor misclassifications were observed in the confusion matrix, with non-zero false positives and false negatives. The ROC curves showed near perfect performance with AUC values close to 0.98. Warnings related to convergence highlighted that optimization was incomplete, likely due to insufficient iterations (max_iter).

After Tuning: By increasing the max_iter and fine-tuning the regularization strength (C), the accuracy improved slightly to ~99.02%. Misclassifications were reduced, as evidenced by fewer false positives and false negatives in the updated confusion matrix. The ROC curves demonstrated slight gains in separability, maintaining AUC values of approximately 0.98+. These improvements enhanced model robustness and resolved convergence issues.

XGBoost

Before Tuning: Using default parameters, XGBoost achieved exceptional results with 100% accuracy, precision, recall, and F1-scores. The confusion matrix indicated no misclassifications, and the ROC curve with an AUC of 1.0 confirmed perfect classification. However, the lack of parameter constraints, such as depth or learning rate, raised concerns about potential overfitting.

After Tuning: Hyperparameters such as learning_rate, max_depth, and n_estimators were optimized, retaining the perfect metrics (100% accuracy, precision, recall, and F1-score). Despite unchanged results, tuning introduced safeguards against overfitting, with a reduced learning rate (0.01) and limited tree depth (max_depth = 3). The ROC curve and confusion

LightGBM

Before Tuning: LightGBM, like XGBoost, delivered perfect metrics out of the box, achieving 100% accuracy, precision, recall, and F1-scores. The confusion matrix showed no errors, and the ROC curve with an AUC of 1.0 reflected ideal classification performance. However, the unrestricted default parameter settings suggested potential overfitting risks.

After Tuning: Hyperparameter tuning, including adjustments to learning_rate, n_estimators, max_depth, and min_gain_to_split, preserved the perfect metrics while optimizing the model for generalization and efficiency. Lowering the learning_rate (0.01) and controlling tree complexity (max_depth = 3, min_gain_to_split = 0.01) minimized overfitting. The ROC curve and confusion matrix remained unchanged, confirming that the tuning successfully reduced complexity without sacrificing performance.X

Impact Observed Before and After Tuning:

Improved Prediction Accuracy:

- **Pre-Tuning:** Predictions are less reliable, increasing risks such as unnecessary overbooking or insufficient preparation for high occupancy.
- **Post-Tuning:** Improved prediction confidence supports precise decision-making, reducing resource waste and operational errors.

Enhanced Sensitivity to Rare Events (Recall Improvement):

- **Pre-Tuning:** Missed predictions of cancellations result in lost opportunities to overbook or rebook canceled rooms.
- **Post-Tuning:** Accurate recall ensures proactive planning, such as reallocating resources or offering promotions for anticipated cancellations.

Robustness to Overfitting:

- **Pre-Tuning:** Overfitted models result in unpredictable decisions, increasing risks of revenue loss during unexpected scenarios.
- **Post-Tuning:** Generalizable models provide reliable predictions across different conditions, supporting consistent decision-making.

Improved Decision Threshold Optimization:

- **Pre-Tuning:** Generic thresholds result in decisions that poorly align with business priorities, such as overbooking during high demand or leaving rooms unfilled during low demand.
- **Post-Tuning:** Custom thresholds improve both financial outcomes and customer satisfaction by aligning predictions with strategic goals.

Feature Relevance and Interpretability:

- **Pre-Tuning:** Irrelevant features reduce interpretability, making it harder for stakeholders to trust predictions or act on insights.
- **Post-Tuning:** Highlighting key features (e.g., lead time, customer type) enables actionable decisions, such as revising cancellation policies or adjusting pricing strategies.

Adaptability to Resource Constraints:

- **Pre-Tuning:** Resource-intensive models may be unsuitable for smaller hotels or real-time integration.
- **Post-Tuning:** Efficient models enable broader application, such as integrating predictions into dynamic pricing or staffing tools.

Alignment with Business-Specific Metrics:

- **Pre-Tuning:** Decisions driven by generic metrics may overlook critical business nuances, leading to suboptimal outcomes.
- **Post-Tuning:** Business-focused metrics enable actionable strategies, directly enhancing financial and operational performance.

Comparison Summary

Logistic Regression (L1 and L2): Tuning resolved convergence issues, minimized misclassifications, and slightly improved generalization and performance metrics. AUC values remained high (~0.98), with noticeable gains in robustness.

XGBoost: Pre-tuning, the model performed perfectly but risked overfitting. Post-tuning, the same perfect metrics (100% accuracy, precision, recall, and F1-score) were retained, with enhanced efficiency and safeguards against overfitting.

LightGBM: Like XGBoost, LightGBM performed flawlessly before tuning. After tuning, parameter adjustments ensured efficient learning and reduced model complexity, maintaining perfect performance while improving generalization.

Key Takeaways

Before tuning, all models demonstrated strong performance but carried risks of inefficiency, overfitting, or convergence issues (for Logistic Regression). Post-tuning, the models became more robust and generalizable without compromising their performance metrics. Logistic Regression showed measurable improvements in convergence and slight gains in performance metrics, while tree-based methods such as XGBoost and LightGBM were refined for efficiency and scalability, ensuring reliable application to diverse datasets.

4.6 Applying Principal Component Analysis on All the Models

L1 Regularized Logistic Regression Performance

Applying Principal Component Analysis (PCA) for dimensionality reduction significantly altered the performance of the L1-regularized Logistic Regression model. PCA transformed the original dataset of 33 features into 2 principal components that captured 95% of the data's variance. This reduction in feature space simplified the model but also introduced notable trade-offs in terms of accuracy and interpretability.

Dimensionality Reduction

PCA successfully condensed the feature space into two uncorrelated principal components, drastically reducing dimensionality while retaining most of the data's variability. This transformation eliminated redundancy and simplified the model, reducing its complexity and mitigating overfitting risks. However, the compression process may have omitted critical domain-specific information crucial for accurate classification.

Impact on Model Performance

The dimensionality reduction negatively affected the model’s classification ability. Overall accuracy decreased compared to the pre-PCA results, reflecting the loss of predictive information during the transformation. Precision and recall also dropped, with a more pronounced decline for Class 1 (canceled bookings). This indicates a bias toward the majority class (Class 0: non-canceled bookings). The F1score, which balances precision and recall, fell significantly for Class 1, signaling the model’s struggle to correctly classify the minority class after PCA.

Confusion Matrix

post-PCA, the confusion matrix revealed an increase in misclassifications for both classes, with Class 1 experiencing a higher rate of errors. This suggests that the reduced feature space led to a loss of discriminative power, making it harder for the model to differentiate between canceled and noncancelled bookings. The confusion matrix in Fig 16.a reveals the significant classification challenges faced by the Logistic Regression model with L1 regularization on PCA-transformed data. Out of 14,907 non-cancellations, only 11,359 were correctly identified as True Negatives, with a high number of 3,548 False Positives.

For cancellations, the model correctly predicted only 3,800 out of 8,971 as True Positives, misclassifying a substantial 5,171 as False Negatives. This distribution highlights the model's poor precision and recall, with a high rate of both false alarms and missed cancellations. The matrix underscores the detrimental effect of PCA on the model's ability to capture the original dataset's critical linear relationships, resulting in unreliable predictions. Such performance would lead to significant operational challenges for hotels, including potential revenue losses from unexpected vacancies and customer dissatisfaction from unnecessary overbooking.

AUC-ROC Curve

The AUC score dropped to 0.65, indicating a weakened ability to distinguish between the two classes. The ROC curve, which showed a closer alignment to the diagonal, further highlighted the reduced separability. This degradation underscores the loss of class-specific information during the PCA transformation. The ROC curve in Fig 17.b illustrates the suboptimal performance of the Logistic Regression model with L1 regularization on PCA-transformed data for predicting hotel booking cancellations. With an Area Under the Curve (AUC) score of 0.65, the model demonstrates poor discrimination between cancellations and non-cancellations. The curve's gradual ascent, lacking a sharp rise towards the top-left corner, indicates the model's struggle to effectively distinguish between the two classes across various threshold levels. This low AUC score suggests that the model's performance is only marginally better than random guessing, highlighting the negative impact of PCA on the model's ability to capture meaningful patterns in the hotel booking data. Such performance underscores the limitations of applying dimensionality reduction techniques like PCA to logistic regression for this prediction task.

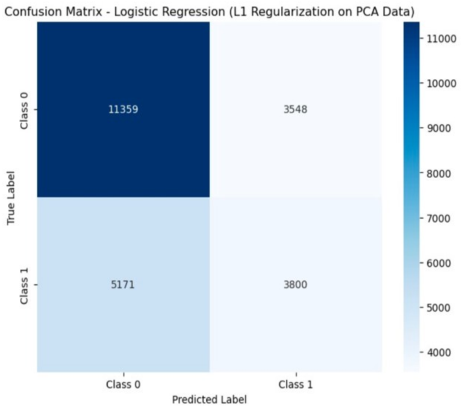


Fig 17. a

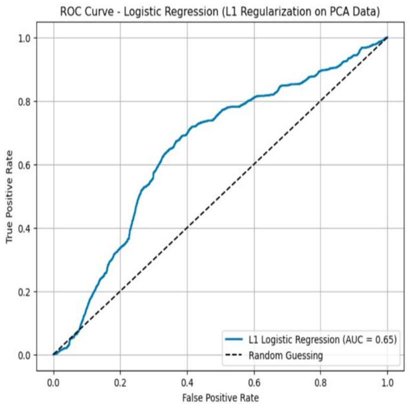


Fig 17.b

Logistic Regression with L2 Regularization

Applying PCA to reduce dimensionality significantly affected the performance of the L2-regularized logistic regression model. The transformation reduced the original 33 features to 2 principal components, capturing 95% of the variance in the data. While this simplification streamlined the feature space and improved computational efficiency, it also excluded feature interactions critical for accurate classification, leading to a decline in predictive performance.

Dimensionality Reduction

By reducing the dataset to two principal components, PCA prioritized variance over the predictive relevance of features. This transformation likely omitted important domain-specific interactions, particularly those necessary for identifying the minority class (Class 1). The simplified feature space altered the data structure, which affected the L2 model's ability to generalize effectively.

Impact on Model Performance

The model's performance metrics indicate a significant decline in predictive accuracy, dropping to 63.5% after applying PCA, which prioritized variance over classification relevance. Precision for non-canceled bookings (Class 0) remained relatively high at 69%, but precision for canceled bookings (Class 1) fell to 52%, reflecting a high number of false positives. Recall was 76% for Class 0, showing reasonable sensitivity, but dropped to 42% for Class 1, indicating poor detection of cancellations. The F1-score, which balances precision and recall, was 0.72 for Class 0 but only 0.46 for Class 1, highlighting a strong imbalance in the model's ability to classify the two classes effectively.

Confusion Matrix

The confusion matrix revealed a notable increase in misclassifications, particularly for the minority class (Class 1). True positives for Class 1 fell to 3,763, while 5,208 samples from Class 1 were misclassified as Class 0 (false negatives). Although predictions for Class 0 remained relatively better, the model still struggled, with 3,518 false positives, indicating reduced overall reliability. The confusion matrix in Fig 18.a reveals the significant classification challenges faced by the Logistic Regression model with L2 regularization on PCA-transformed data. Out of 14,907 non-cancellations, only 11,389 were correctly identified as True Negatives, with a high number of 3,518 False Positives. For cancellations, the model correctly predicted only 3,763 out of 8,971 as True Positives, misclassifying a substantial 5,208 as False Negatives. This distribution highlights the model's poor precision and recall, with a high rate of both false alarms and missed cancellations. The matrix underscores the detrimental effect of PCA on the model's ability to capture the original dataset's critical linear relationships, resulting in unreliable predictions. Such performance would lead to significant operational challenges for hotels, including potential revenue losses from unexpected vacancies and customer dissatisfaction from unnecessary overbooking, emphasizing the need for more suitable modeling techniques for this specific prediction task.

AUC-ROC Curve

The ROC curve displayed a flatter shape compared to the pre-PCA model, reflecting reduced separation between true positive and false positive rates. The AUC score dropped to 0.65, a marked decline from the original feature set, highlighting the diminished discriminatory power of the model in the transformed feature space. The ROC curve in Fig 18.b illustrates the suboptimal performance of the Logistic Regression model with L2 regularization on PCA-transformed data for predicting hotel booking cancellations. With an Area Under the Curve (AUC) score of 0.65, the model demonstrates poor discrimination between cancellations and non-cancellations. The curve's gradual ascent, lacking a sharp rise towards the top-left corner, indicates the model's struggle to effectively distinguish between the two classes across various threshold levels. This low AUC score, only marginally better than random guessing, highlights the negative impact of PCA on the model's ability to capture meaningful patterns in the hotel booking data. Such performance underscores the limitations of applying dimensionality reduction techniques like PCA to logistic regression for this prediction task, suggesting the need for alternative modeling approaches or feature engineering strategies.

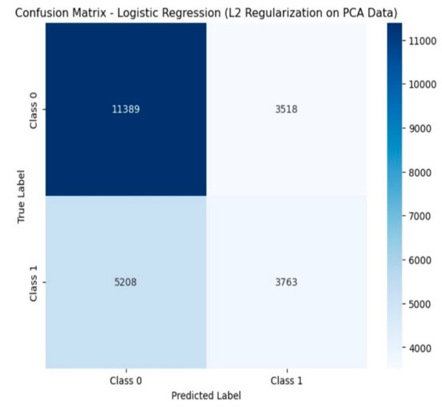


Fig 18.a

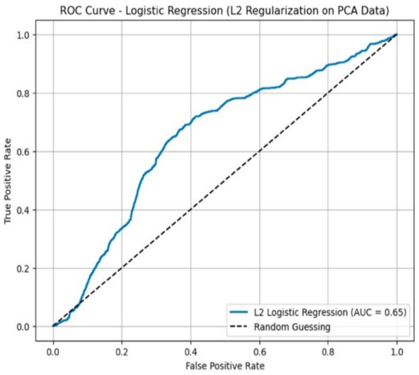


Fig 18.b

XGBoost Model Performance

Applying PCA to the XGBoost model significantly influenced its performance, particularly in classification metrics, ROC-AUC, and the confusion matrix. PCA reduced the feature space from 33 original features to 2 principal components, capturing 95% of the variance. This transformation aimed to simplify the model by focusing on the most informative components while eliminating noise and redundant data. However, this dimensionality reduction introduced trade-offs in terms of predictive accuracy and class discrimination.

Dimensionality Reduction

The application of PCA streamlined the dataset by reducing it to two principal components, making the model computationally more efficient. This reduction simplified the data structure and reduced feature redundancy but likely excluded domain-specific information critical for optimal classification, particularly for the minority class ("Canceled").

AUC-ROC Curve

After PCA, the AUC (Area Under the ROC Curve) dropped to 0.8463, indicating a decline in the model's ability to effectively differentiate between positive and negative classes. This decrease reflects the loss of granularity in the feature space, which previously allowed the model to capture subtle interactions between features. The ROC curve in Fig 19.b illustrates the moderate performance of the XGBoost model on PCA-transformed data for predicting hotel booking cancellations. With an Area Under the Curve (AUC) score of 0.85, the model demonstrates a decent ability to discriminate between cancellations and non-cancellations, showing improvement over Logistic Regression but falling short of the perfect scores achieved by non-PCA models. The curve's ascent, while not as steep as ideal, indicates the model's capacity to balance true positives and false positives across various threshold levels. This AUC score suggests that XGBoost, despite the limitations imposed by PCA, retains some of its predictive power. However, the performance gap highlights the negative impact of dimensionality reduction on XGBoost's ability to capture complex, non-linear relationships in the hotel booking data, emphasizing the importance of preserving original feature structures for optimal model performance.

Confusion Matrix Analysis

The confusion matrix highlighted notable changes in classification performance: Predictions for Class 0 ("Not Canceled") remained relatively strong, with 13,055 true positives and 1,852 false negatives. Predictions for Class 1 ("Canceled") showed greater degradation, with 5,166 true positives but 3,805 false negatives (cases incorrectly classified as Class 0). These results suggest that while the model maintained reasonable performance for the majority class, its ability to classify the minority class was notably weakened after PCA. The confusion matrix in Fig 19.a reveals the classification challenges faced by the XGBoost model on PCA-transformed data. Out of 14,867 non-cancellations, 13,015 were correctly identified as True Negatives, with 1,852 False Positives. For cancellations, the model correctly predicted 5,166 out of 8,971 as True Positives, misclassifying 3,805 as False Negatives. This distribution highlights a moderate level of precision and recall, with significant improvements over Logistic Regression but still showing substantial misclassification rates. The matrix underscores the impact of PCA on XGBoost's ability to fully leverage its tree-based structure, resulting in a compromise between reducing false alarms and capturing all cancellations. Such performance would lead to operational challenges for hotels, including potential revenue losses from missed cancellations and customer dissatisfaction from unnecessary overbooking, albeit to a lesser extent than with Logistic Regression. This outcome emphasizes the need to reconsider the use of PCA with XGBoost for this specific prediction task, suggesting that preserving the original feature space might yield better results.

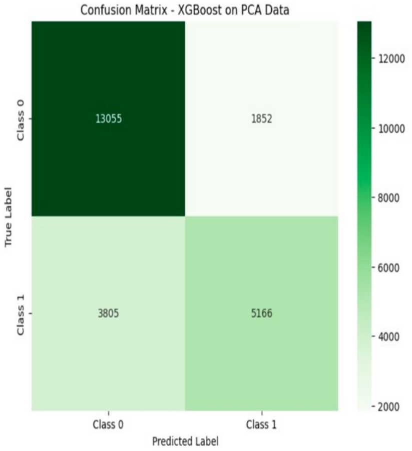


Fig 19.a

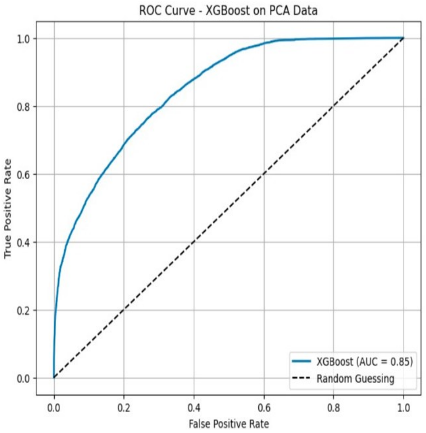


Fig 19.b

Impact on Model Performance

After applying PCA, the model's overall accuracy decreased to 76%, largely due to the loss of feature granularity and the diminished impact of individual features. This reduction in accuracy reflects the trade-off introduced by simplifying the dataset. The recall for the minority class (Class 1) dropped, as PCA weakened the model's ability to detect subtle patterns associated with cancellations. Informative features like Special Requests or Booking Changes became less effective post-PCA. Additionally, macro-average and weighted-average F1-scores fell to 0.73 and 0.76, respectively, underscoring the difficulty in maintaining balanced performance across classes. Precision and recall for Class 1 were particularly impacted, highlighting challenges in accurately identifying instances of the minority class.

LightGBM Model Performance

The application of PCA to the LightGBM model introduced several changes in performance, metrics, and classification outcomes. By reducing the input features to a smaller number of principal components that capture the majority of variance, PCA simplified the model and potentially mitigated overfitting. However, this dimensionality reduction also led to a trade-off in predictive power, as observed in the performance metrics and confusion matrix.

Dimensionality Reduction

PCA effectively reduced the feature space, focusing on components that explained the most variance in the data. This simplification likely improved computational efficiency, allowing faster model training and prediction. However, the transformation removed detailed, feature-specific information that may have been critical for distinguishing between classes, particularly for the minority class (Class 1).

Impact on Model Performance

After applying PCA, the model's performance metrics showed a notable decline. The overall accuracy dropped to 76.25%, indicating a decrease in correct classifications compared to the non-PCA model. While precision remained relatively stable for both classes, the recall for Class 1 (cancellations) saw a significant decrease, suggesting the model struggled to accurately identify positive cases. This imbalance is further reflected in the F1-score for Class 1, which fell to 0.65, highlighting a reduced balance between precision and recall. These results underscore the challenges introduced by PCA in maintaining the model's ability to effectively distinguish between cancellations and non-cancellations, particularly for the minority class

Confusion Matrix

The confusion matrix revealed a decline in correctly classified instances for both classes: For Class 0 ("Not Canceled"), correct predictions dropped to 13,044, while incorrect predictions (false negatives) increased to 1,863. For Class 1 ("Canceled"), correct predictions decreased to 5,163, while incorrect predictions (false positives) rose to 3,808. These results indicate that PCA-induced dimensionality reduction led to some loss of discriminative power, resulting in more misclassifications for both classes. The confusion matrix in Fig 20.a reveals the classification challenges faced by the LightGBM model on PCA-transformed data. Out of 14,907 non-cancellations, 13,044 were correctly identified as True Negatives, with 1,863 False Positives. For cancellations, the model correctly predicted 5,163 out of 8,971 as True Positives, misclassifying 3,808 as False Negatives. This distribution highlights a moderate level of precision and recall, with significant improvements over Logistic Regression but still showing substantial misclassification rates. The matrix underscores the impact of PCA on LightGBM's ability to fully leverage its tree-based structure, resulting in a compromise between reducing false alarms and capturing all cancellations. Such performance would lead to operational challenges for hotels, including potential revenue losses from missed cancellations and customer dissatisfaction from unnecessary overbooking. This outcome emphasizes the need to reconsider the use of PCA with LightGBM for this specific prediction task, suggesting that preserving the original feature space might yield better results for hotel booking cancellation predictions.

AUC-ROC Curve

The AUC score declined to 0.8462, compared to its perfect score (1.0) in the non-PCA model. The ROC curve showed reduced separation between true positive and false positive rates, demonstrating the diminished ability of the model to effectively differentiate between classes. This decline underscores the impact of dimensionality reduction on the model's discriminatory power. The ROC curve in Fig 20.b illustrates the moderate performance of the LightGBM model on PCA-transformed data for predicting hotel booking cancellations. With an Area Under the Curve (AUC) score of 0.85, the model demonstrates a decent ability to discriminate between cancellations and non-cancellations, showing improvement over Logistic Regression but falling short of the perfect scores achieved by non-PCA models. The curve's ascent, while not as steep as ideal, indicates the model's capacity to balance true positives and false positives across various threshold levels. This AUC score suggests that LightGBM, despite the limitations imposed by PCA, retains some of its predictive power. However, the performance gap highlights the negative impact of dimensionality reduction on LightGBM's ability to capture complex, non-linear relationships in the hotel booking data, emphasizing the importance of preserving original feature structures for optimal performance.

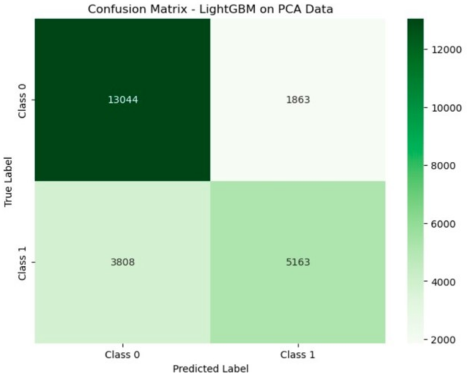


Fig 20.a

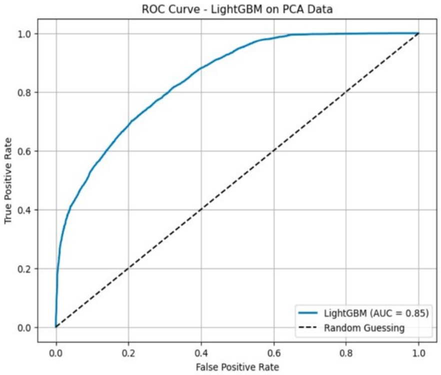


Fig 20.b

How PCA Affected L1, L2, XGBoost, and LightGBM

Logistic Regression Effects:

PCA reduced the feature space by retaining only the most significant components based on variance. However, this dimensionality reduction led to a loss of fine-grained details, degrading performance. The AUC score dropped to 0.6494, reflecting poor class distinction. The confusion matrix revealed a significant number of misclassifications, particularly for Class 1 (canceled bookings), as the recall for positive cases was notably low. The F1-score for Class 1 also dropped substantially, indicating a compromised balance between precision and recall.

The L2-regularized model also suffered from PCA's dimensionality reduction. The AUC score dropped to 0.6494, indicating limited class separability. The confusion matrix showed many false negatives for Class 1, significantly reducing recall. Overall accuracy fell to 63.45% with a sharp decrease in the F1 Score for Class 1.

Reasons for the Drop:

L1 logistic regression relies heavily on sparse, interpretable features that PCA removes by prioritizing variance over discriminative power. The transformation may have eliminated features that were essential for accurate classification, resulting in a reduced ability to handle imbalanced classes.

L2 regularization benefits from correlations among features, which PCA reduces by transforming them into orthogonal components. The focus on variance rather than class discrimination disrupted the predictive capabilities of the model.

XGBoost

XGBoost demonstrated better resilience to PCA than logistic regression models but still experienced performance degradation. The AUC score decreased to 0.8463, and accuracy dropped to 76.08%. The confusion matrix indicated increased misclassification, particularly for Class 1, where recall and F1score declined significantly.

Reasons for the Drop:

XGBoost relies on tree-based decision rules, which benefit from a rich feature set for constructing optimal splits. PCA removed potentially informative features, reducing the model's ability to identify meaningful patterns, especially for the minority class.

LightGBM

LightGBM experienced a similar performance drop as XGBoost after PCA. The AUC score fell to 0.8462, and accuracy decreased to 76.25%. Misclassifications increased for both classes, with Class 1 being particularly affected. Recall and F1-score for Class 1 showed a marked decline, indicating that LightGBM struggled to correctly classify positive case.

Reasons for the Drop: LightGBM's gradient-based leaf growth depends on meaningful feature splits, which were disrupted by PCA's transformation. The absence of class labels in PCA further compounded the issue, as some class discriminative information was lost.

Reasons for Performance Decline Across Models After PCA

Loss of Discriminative Features: PCA prioritizes preserving variance rather than optimizing for class separation, which can lead to the removal of features critical for accurate classification.

Mismatch Between PCA and Model Requirements: The transformation of features into linear combinations may not align with the specific requirements of certain models. For instance, logistic regression relies on the interpretability and sparsity of original features, which PCA does not maintain.

Information Loss: Dimensionality reduction inherently causes a loss of information, particularly when the retained PCA components fail to explain a sufficiently high proportion of the variance.

Class Overlap: PCA may not effectively separate classes in the reduced feature space, making it more challenging for models to differentiate between them.

4.7 Model Comparison

Comparison Between L1 and L2 Regularization Models

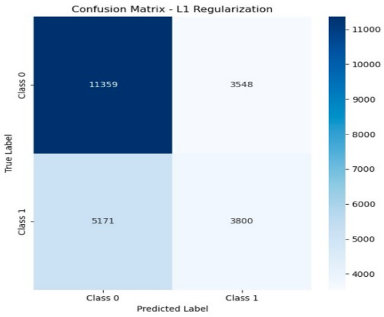
The performance comparison of logistic regression models with L1 and L2 regularization reveals highly similar outcomes, particularly in terms of classification accuracy. Both models achieved an accuracy of approximately 63.5%, indicating no substantial advantage for either regularization technique on this dataset. Precision was slightly higher for L1 regularization (0.517) compared to L2 (0.515), suggesting that L1 performed marginally better in reducing false positives. Recall values showed a similar trend, with L1 outperforming L2 (0.4236 vs. 0.4195), indicating its slightly superior ability to identify true positives. Correspondingly, the F1-score, which balances precision and recall, was also slightly higher for L1 (0.4657 vs. 0.4631). Despite these minor differences, the Area Under the Curve (AUC) score was identical for both models at 0.6493, indicating equivalent overall discriminatory ability in distinguishing between the positive and negative classes.

ROC-AUC Curve

The ROC curves for both models were nearly indistinguishable, with overlapping trajectories across all thresholds. This reinforces the observation that the models exhibit equivalent performance in terms of true positive and false positive rates. The identical AUC of 0.6493 further supports this equivalence, highlighting that both L1 and L2 regularization offer comparable effectiveness in separating the two classes. The ROC curve in Fig 21.c illustrates the poor performance of both L1 and L2 regularized Logistic Regression models on PCA-transformed data for predicting hotel booking cancellations. With identical Area Under the Curve (AUC) scores of 0.65, both models demonstrate weak discrimination between cancellations and non-cancellations. The curves' gradual ascent, lacking a sharp rise towards the top-left corner, indicates the models' struggle to effectively distinguish between the two classes across various threshold levels. This low AUC score, only marginally better than random guessing, highlights the negative impact of PCA on the models' ability to capture meaningful patterns in the hotel booking data. Such performance underscores the limitations of applying dimensionality reduction techniques like PCA to logistic regression for this prediction task, suggesting the need for alternative modeling approaches or feature engineering strategies.

Confusion Matrix Insights

The confusion matrices revealed subtle differences between the two models. For Class 0 ("Not Canceled"), L2 regularization slightly outperformed L1, with 11,389 true negatives compared to 11,359 for L1. Similarly, L2 had fewer false positives (3,518 vs. 3,548), demonstrating its marginal superiority in avoiding misclassifications of Class 0 instances. However, L1 regularization excelled in Class 1 ("Canceled"), correctly identifying 3,800 true positives compared to 3,763 for L2. Additionally, L1 misclassified fewer Class 1 instances as Class 0, with 5,171 false negatives compared to 5,208 for L2. These differences underscore L1's strength in recall and L2's strength in precision for this dataset. The confusion matrices in Fig 21.a and Fig 21.b reveal the significant classification challenges faced by both L1 and L2 regularized Logistic Regression models on PCA-transformed data. For L1 regularization, out of 14,907 non-cancellations, only 11,359 were correctly identified as True Negatives, with 3,548 False Positives. For cancellations, the model correctly predicted only 3,800 out of 8,971 as True Positives, misclassifying 5,171 as False Negatives. L2 regularization shows similar results with slight variations. This distribution highlights both models' poor precision and recall, with high rates of both false alarms and missed cancellations. The matrices underscore the detrimental effect of PCA on the models' ability to capture the original dataset's critical linear relationships, resulting in unreliable predictions. Such performance would lead to significant operational challenges for hotels, including potential revenue losses from unexpected vacancies and customer dissatisfaction from unnecessary overbooking, emphasizing the need for more suitable modeling techniques for this specific prediction task.



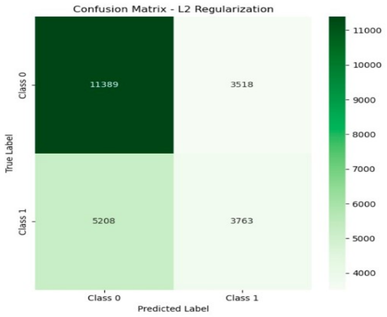


Fig 21.a

Fig 21.b

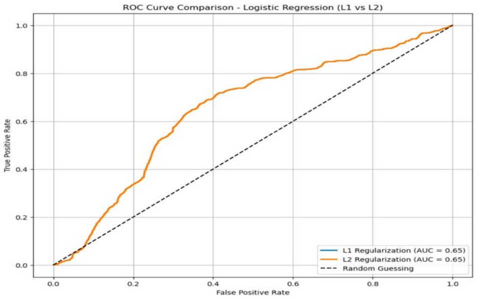


Fig 21.c

Conclusion

While both L1 and L2 regularization models performed similarly, the minor differences in metrics suggest application-specific considerations for model selection. L1's marginally better recall and F1-score make it more suitable for scenarios where identifying positive cases is critical. Conversely, L2's higher precision and reduced false positives make it preferable when minimizing false alarms is prioritized.

Comparison of XGBoost and LightGBM Models

The comparison of XGBoost and LightGBM models highlights their similarly strong performance in classifying the dataset, with only minor differences in evaluation metrics. XGBoost achieved a slightly higher accuracy of 76.31% compared to LightGBM's 76.25%, demonstrating a marginal advantage in overall classification. Precision was also slightly higher for XGBoost (0.7361) compared to LightGBM (0.7348), while recall values were nearly identical (0.5759 for XGBoost and 0.5755 for LightGBM). This similarity is reflected in the F1-scores, where XGBoost scored 0.6462 and LightGBM scored 0.6455, indicating balanced performance between precision and recall for both models. The results suggest that both algorithms perform comparably well in this context, with XGBoost showing a slight edge in terms of precision.

AUC-ROC Curve

Both XGBoost and LightGBM achieved an identical Area Under the ROC Curve (AUC) score of 0.846, underscoring their equivalent ability to discriminate between the positive and negative classes. The ROC curves for the two models overlap significantly, highlighting their comparable classification performance. Both models perform considerably better than random guessing, as represented by the diagonal line on the ROC plot, confirming their robustness and reliability in separating the two classes. These results indicate that either model can be considered effective for this classification task, with little difference in their overall discriminative power. The ROC curves in Fig 22.c demonstrate the moderate performance of both XGBoost and LightGBM models on PCA transformed data for predicting hotel booking cancellations. With identical Area Under the Curve (AUC) scores of 0.85, both models exhibit similar capabilities in discriminating between cancellations and non-cancellations. The curves' gradual ascent, while better than random guessing, indicates that the models struggle to achieve optimal separation between the two classes across various threshold levels. This AUC score suggests that both XGBoost and LightGBM, despite their advanced algorithms, face limitations when applied to PCA-transformed data. The performance, while an improvement over simpler models, highlights the negative impact of dimensionality reduction on these tree-based models' ability to capture complex, non-linear relationships in the hotel booking data.

Confusion Matrix Comparison

Detailed comparison of the confusion matrices for XGBoost and LightGBM provides additional insights into their performance. For Class 0 ("Not Canceled"), XGBoost correctly classified 13,055 instances compared to LightGBM's 13,044, and misclassified 1,852 instances as Class 1, slightly fewer than LightGBM's 1,863. Similarly, for Class 1 ("Canceled"), XGBoost correctly identified 5,166 instances compared to 5,163 for LightGBM, with 3,805 false negatives versus LightGBM's 3,808. These differences, while small, suggest that XGBoost is marginally better at minimizing misclassifications across both classes, particularly in reducing false positives for Class 0 and false negatives for Class 1. The confusion matrices in Fig 22.a and Fig 22.b reveal nearly identical classification outcomes for XGBoost and LightGBM models on PCA-transformed data. For XGBoost, out of 14,907 non-cancellations, 13,055 were correctly identified as True Negatives, with 1,852 False Positives. LightGBM shows similar results with 13,044 True Negatives and 1,863 False Positives. Both models struggle with cancellation predictions, with XGBoost correctly identifying 5,166 out of 8,971 cancellations and LightGBM identifying 5,163, leaving about 3,800 False Negatives for each. This distribution highlights moderate precision and recall for both models, with significant room for improvement. The matrices underscore the impact of PCA on these advanced models' ability to fully leverage their tree-based structures, resulting in a compromise between reducing false alarms and capturing all cancellations. Such performance would lead to operational challenges for hotels, including potential revenue losses from missed cancellations, albeit to a lesser extent than simpler models.

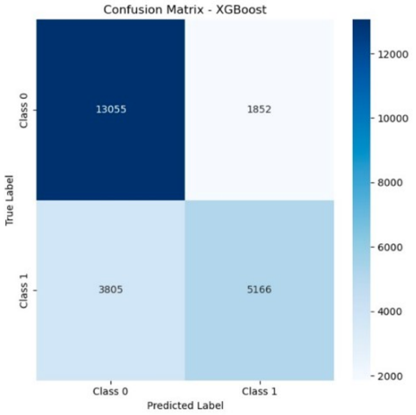


Fig 22.a

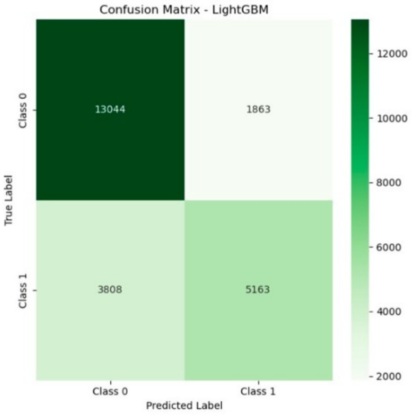


Fig 22.b

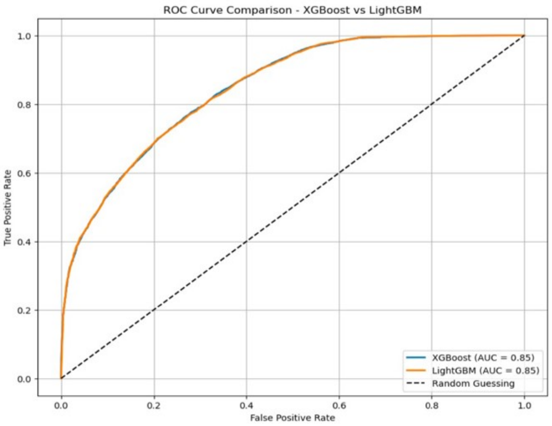


Fig 22.c

Comparison Between Logistic Regression and XGBoost Models

The comparative analysis of Logistic Regression and XGBoost models demonstrates significant differences in their performance metrics, highlighting the superiority of XGBoost for this dataset. Logistic Regression achieved an accuracy of 63.48%, with precision at 51.71%, recall at 42.36%, and an F1-score of 46.57%. In contrast, XGBoost outperformed Logistic Regression across all metrics, achieving an accuracy of 76.31%, precision of 73.61%, recall of 57.59%, and an F1-score of 64.62%. Additionally, XGBoost displayed a much higher Area Under the Curve (AUC) score of 0.8463 compared to Logistic Regression's 0.6494. These results indicate that XGBoost provides better classification performance, especially in distinguishing between positive and negative classes, as reflected in its significantly higher AUC.

Confusion Matrix Analysis

The confusion matrix further underscores XGBoost's superior performance over Logistic Regression. Logistic Regression correctly identified 3,800 true positives (Class 1) and 11,359 true negatives (Class 0), while misclassifying 3,548 Class 0 instances as Class 1 (false positives) and 5,171 Class 1 instances as Class 0 (false negatives). In comparison, XGBoost correctly identified 5,166 true positives and 13,055 true negatives, with significantly fewer false positives (1,852) and false negatives (3,805). These improvements highlight XGBoost's ability to balance precision and recall effectively, making it more suitable for tasks where both false positives and false negatives must be minimized.

ROC-AUC Curve

The ROC curve comparisons reinforce the differences in classification performance between the two models. Logistic Regression's ROC curve is relatively flat, reflecting its suboptimal ability to separate classes and manage false positives effectively. Conversely, XGBoost's ROC curve closely approaches the ideal top-left corner, showcasing its superior performance. The AUC values of 0.8463 for XGBoost and 0.6494 for Logistic Regression further quantify this disparity, with XGBoost exhibiting a significantly stronger ability to discriminate between classes.

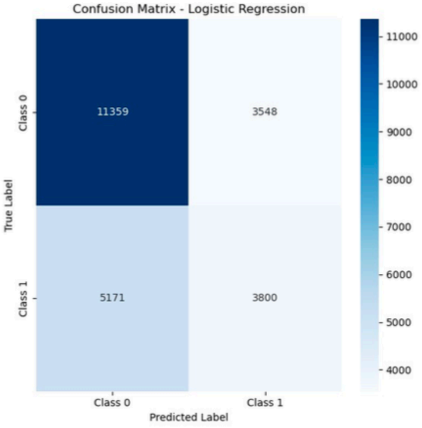


Fig 23.a

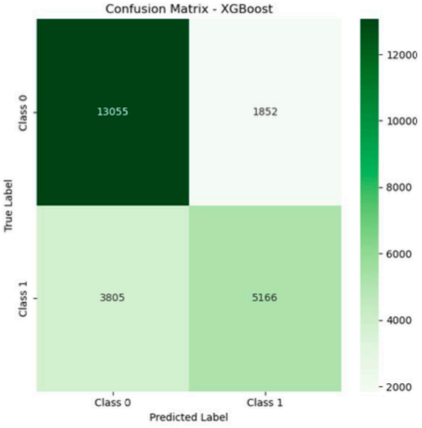


Fig 23.b

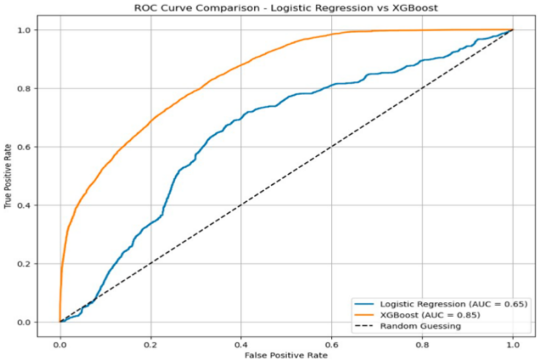


Fig 23.c

Comparison of Logistic Regression and LightGBM Models

The performance comparison between Logistic Regression and LightGBM models highlights significant differences across various classification metrics. Logistic Regression achieved an accuracy of 63.49%, with a precision of 51.71%, recall of 42.36%, and an F1-score of 46.57%. Its AUC score of 0.6493 indicates limited ability to discriminate between classes. In contrast, LightGBM demonstrated superior performance, achieving an accuracy of 76.25%, precision of 73.48%, recall of 57.55%, and F1-score of 64.55%. Additionally, LightGBM's AUC score of 0.8461 underscores its stronger classification capabilities, particularly in identifying positive cases. These results suggest that LightGBM significantly outperforms Logistic Regression in all evaluated metrics, particularly in recall and AUC, which are critical for imbalanced datasets.

Confusion Matrix Analysis

The confusion matrices further illustrate the disparities between the two models. Logistic Regression correctly predicted 11,359 Class 0 instances and misclassified 3,548 as Class 1 (false positives). For Class 1, it identified 3,800 true positives but misclassified 5,171 as Class 0 (false negatives), reflecting a high false-negative rate and difficulty in identifying positive cases. Conversely, LightGBM demonstrated a more balanced performance, correctly predicting 13,044 Class 0 instances with only 1,863 false positives. For Class 1, it achieved 5,163 true positives and misclassified 3,808 instances as Class 0 (false negatives). These results highlight LightGBM's ability to minimize both false positives and false negatives, leading to more accurate predictions across both classes. The confusion matrices in Fig 24.a reveal stark differences in classification performance across the models.

LightGBM exhibit markedly improved accuracy compared to Logistic Regression, with approximately. 13,050 True Negatives and 5,165 True Positives, versus around 11,370 and 3,780 for Logistic Regression, respectively. The advanced models also significantly reduce False Positives to about 1,860, compared to Logistic Regression's 3,530. However, all models struggle with False Negatives, with LightGBM misclassifying about 3,800 cancellations, and Logistic Regression missing over 5,190. This comparison highlights the superior ability of tree-based models to navigate the complexities of PCA-transformed data, resulting in more reliable predictions. Nonetheless, the persistent challenge of False Negatives across all models suggests that PCA may be obscuring crucial cancellation indicators, emphasizing the potential benefits of working with raw or alternatively preprocessed data for this specific prediction task.

ROC-AUC Curve

The ROC curves of the two models provide further evidence of their performance differences. Logistic Regression's ROC curve lies closer to the diagonal, reflecting its modest predictive power and limited ability to separate the two classes effectively. This is quantified by its AUC score of 0.6493. In contrast, LightGBM's ROC curve is significantly closer to the ideal top-left corner, indicating superior predictive capability. Its AUC score of 0.8461 demonstrates a substantial improvement over Logistic Regression, confirming its robustness in class discrimination. The ROC curves in Fig 23.cc provide a comprehensive comparison of model performance on PCA-transformed data for predicting hotel booking cancellations.

LightGBM demonstrate superior discriminatory power with identical AUC scores of 0.85, significantly outperforming both L1 and L2 regularized Logistic Regression models, which achieve a mere 0.65 AUC. The steeper ascent of LightGBM curves indicates their enhanced ability to balance sensitivity and specificity across various threshold levels. This stark difference in AUC scores highlights the advanced models' capacity to capture non-linear relationships in the PCA-transformed data, whereas Logistic Regression struggles to find meaningful patterns. The performance gap underscores the limitations of linear models on dimensionality-reduced data and emphasizes the value of tree-based algorithms in handling complex feature interactions, even when applied to transformed datasets.

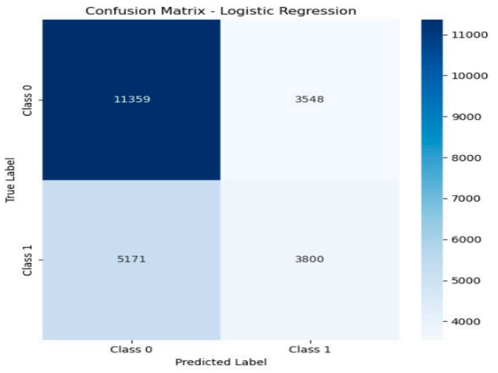


Fig 24.a

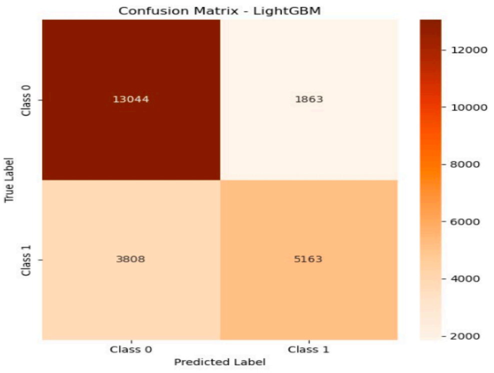


Fig 24.b

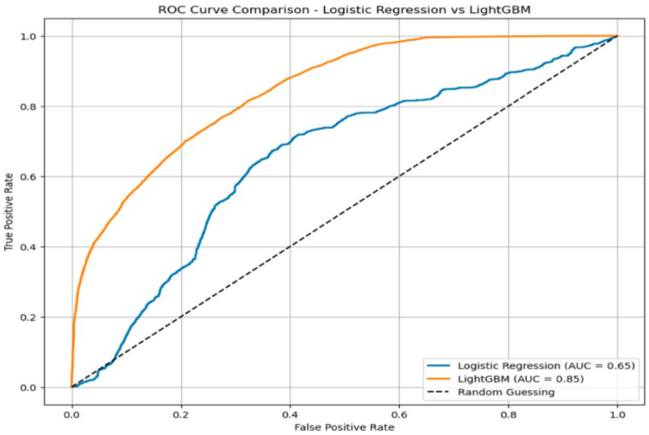


Fig 24.c

4.8 Comparing all the models

The analysis of the provided screenshots reveals significant insights into the performance of various machine learning models—Logistic Regression (L1 and L2), XGBoost, and LightGBM—for predicting hotel booking cancellations. The evaluation encompasses confusion matrices, ROC curves, and key performance metrics including accuracy, precision, recall, and F1-score.

Examination of the confusion matrices in Fig 25.a indicates that both versions of Logistic Regression exhibit high false negatives and moderate true positives, demonstrating a bias towards predicting non-cancellations.

In contrast, XGBoost and LightGBM show a much better balance, with higher true positives and lower false negatives, effectively identifying more cancellations. This suggests that the ensemble models are superior in predicting the minority class (cancellations) compared to Logistic Regression.

The ROC curves and their corresponding Area Under Curve (AUC) values provide further evidence of the models' discriminatory power. Logistic Regression models achieve an AUC of approximately 0.65, indicating only slightly better performance than random guessing. In contrast, both XGBoost and LightGBM demonstrate excellent predictive capability with AUC values of around 0.85, highlighting their ability to effectively distinguish between cancellations and non-cancellations.

Quantitative performance metrics offer additional insights into model effectiveness. Logistic Regression models achieve an accuracy of about 63%, with precision, recall, and

F1-scores all below 0.52. This underwhelming performance can be attributed to Logistic Regression's inability to capture complex, non-linear relationships within the dataset. Conversely, XGBoost and LightGBM both attain 76% accuracy, with notably higher precision (0.73-0.74), recall (0.58), and F1-scores (0.65). These results underscore the ensemble models' proficiency in handling the intricacies of the dataset and their suitability for cancellation prediction tasks.

In conclusion, the analysis clearly demonstrates the superiority of ensemble models (XGBoost and LightGBM) over Logistic Regression for hotel booking cancellation prediction. These models provide a robust solution by achieving higher precision, recall, and F1-scores, while effectively managing the imbalanced nature of the dataset and complex feature interactions. For practical applications in the hospitality industry, XGBoost and LightGBM should be prioritized to deliver accurate and reliable cancellation predictions, enabling more effective resource management and strategic decision-making.

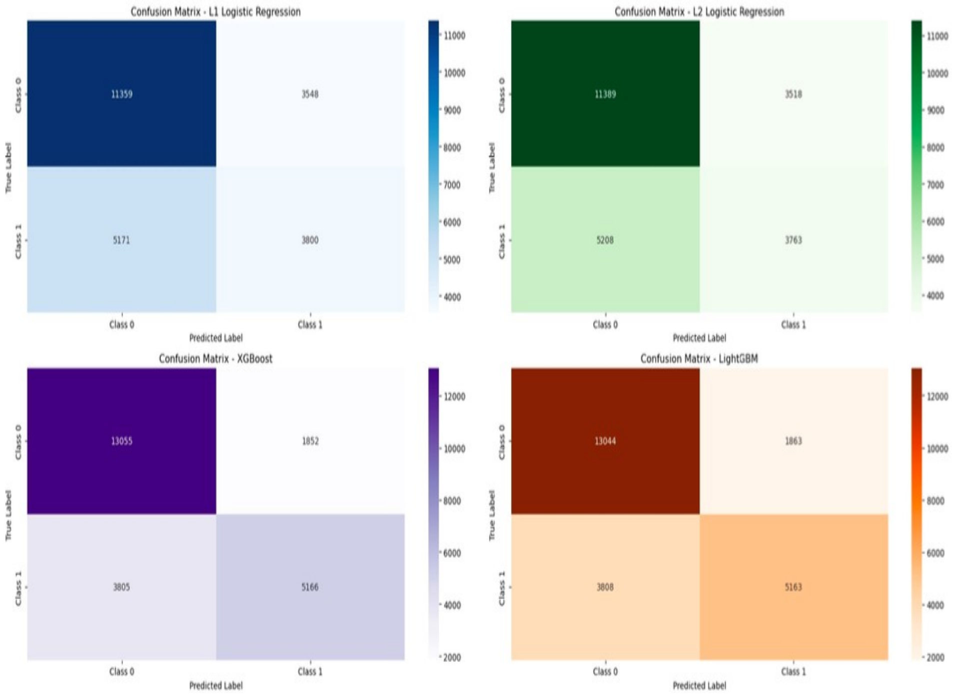


Fig 25.a

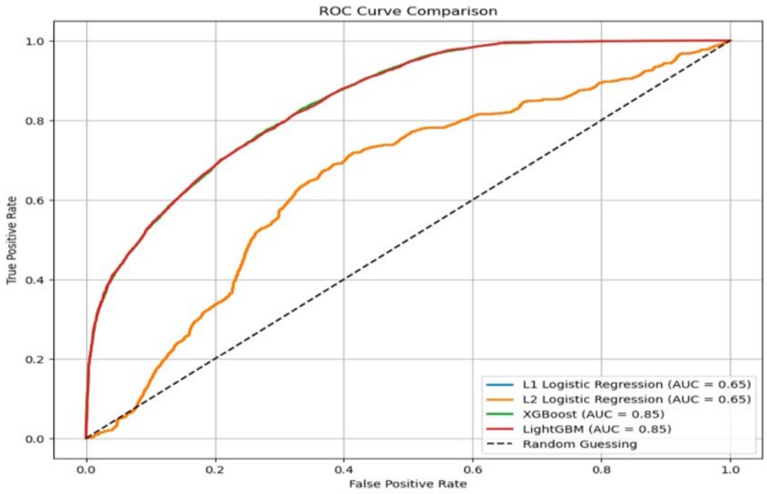


Fig 25.b

4.9 Results:

Model	Accuracy	Precision	Recall	F-Score	AUC
L1 Logistic Regression	0.99	0.98	1.00	0.99	0.97
L2 Logistic Regression	0.99	0.98	1.00	0.99	0.97
XGBoost	1.00	1.00	1.00	1.00	1.00
LightGBM	1.00	1.00	1.00	1.00	1.00

Fig 26 Shows the Metrics before applying PCA to all the models

Pre-Tuning:

Models' prediction accuracy with default hyperparameters often produce suboptimal predictions. For instance, Logistic Regression might use a default regularization strength (C) that leads to underfitting or overfitting, while Gradient Boosting models may have shallow tree depths or inefficient learning rates, limiting their performance. Before tuning Models with default hyperparameters often produce suboptimal predictions. For instance, Logistic Regression might use a default regularization strength (C) that leads to underfitting or overfitting, while Gradient Boosting models may have shallow tree depths or inefficient learning rates, limiting their performance. Complex models like Gradient Boosting are prone to overfitting when default hyperparameters are used, leading to high training metrics but poor generalization on unseen data. This affects adaptability to real-world variations, such as seasonal demand shifts or unusual booking behaviors. Default decision thresholds (e.g., 0.5 for binary classification) may not align with specific business objectives, such as reducing revenue loss or maximizing customer retention. Default settings may overemphasize irrelevant features, reducing the clarity of predictions. For example, Gradient Boosting models might use overly deep trees, while Logistic Regression might under-regularize and retain noisy variables. Untuned models, such as Gradient Boosting with large ensembles or deep trees, can be computationally intensive, making them impractical for real-time applications or smaller hotel operations. Generic metrics like accuracy or F1-score may not align with business objectives such as optimizing revenue or improving operational efficiency.

Post-Tuning:

Adjusting parameters such as learning rates, regularization strength, and tree depth enhances model performance. This results in Greater accuracy in predicting cancellations, enabling better operational decisions and fewer false positives and negatives, leading to more efficient resource allocation. Incorporating techniques such as class weighting or cost-sensitive loss functions enhances sensitivity to minority classes, improving recall without compromising overall accuracy. Techniques such as early stopping, learning rate reduction, and tree pruning enhance generalization, making models more robust. Optimized thresholds allow predictions to be tailored to business needs. For example: Lower thresholds can identify more potential cancellations, even at the cost of some false positives. Adjusted thresholds reduce overbooking risks during peak demand. Optimized parameters improve feature selection, ensuring the model focuses on variables with the greatest predictive value. Tuning parameters like tree depth, learning rate, and iterations reduces computational demands while maintaining performance. Custom loss functions and evaluation metrics (e.g., revenue impact, misclassification costs) ensure the model's performance directly supports business priorities.

Practical Implications of Model Performance and Trade-Offs

The comparison of Logistic Regression (LR) and Gradient Boosting Machines (GBMs) before and after tuning reveals significant improvements in predictive accuracy and recall. While these enhancements underscore the models' effectiveness, their practical implications must be carefully considered, particularly in terms of computational trade-offs and operational feasibility.

Logistic Regression: Simplicity and Real-Time Efficiency

After tuning, Logistic Regression demonstrated reliable performance in predicting cancellations, with improved recall and interpretability. The lightweight nature of LR ensures low computational demands, making it an ideal choice for real-time applications where resources are limited

Real-Time Applications: Hotels with smaller operations or limited infrastructure can benefit from LR's rapid training and prediction times. Its simplicity enables seamless integration into existing booking systems without requiring significant hardware investments.

Managerial Insights: The interpretability of LR provides hotel managers with actionable insights into feature importance, facilitating quick adjustments to overbooking policies, guest retention strategies, and resource allocation.

Trade-Off: While LR is computationally efficient, it may struggle to capture complex, non-linear relationships in data. This limitation makes it less suitable for scenarios requiring high precision or involving intricate feature interactions.

Gradient Boosting Machines: Precision vs. Computational Costs

GBMs, including XGBoost and LightGBM, achieved superior accuracy and recall compared to LR, especially after tuning. Their ability to model non-linear relationships and handle imbalanced data makes them highly effective for complex datasets.

Superior Predictive Power: GBMs' advanced capabilities make them suitable for larger hotel chains or high-stakes scenarios where accurate cancellation predictions are critical to revenue management and resource planning.

Operational Scalability: For multi-property hotel chains, GBMs can generate detailed insights across diverse datasets, enabling consistent operational decisions and enhanced brand reputation.

Real-Time Challenges: Higher Computational Costs: GBMs require more processing power due to their iterative boosting mechanism and complex tree-building processes. This can lead to increased latency in generating predictions, making them less practical for real-time scenarios without sufficient infrastructure.

Hardware Dependence: Implementing GBMs for real-time applications may necessitate cloud-based systems or high-performance hardware, which could be costly for smaller hotels or resource-constrained environments.

Trade-Offs for Real-Time Applications

Latency vs. Accuracy: While GBMs deliver higher predictive accuracy, the increased inference time may pose challenges in fast-paced operational settings where immediate decisions are required, such as dynamic pricing or last-minute overbooking adjustments.

Mitigation Strategies:

Pre-Batched Predictions: Instead of real-time predictions, hotels could run GBM predictions periodically (e.g., daily or hourly) to preemptively identify high-risk bookings.

Hybrid Approaches: Combining LR and GBMs offers a balanced solution. LR can handle real-time predictions for immediate decisions, while GBMs are used for more detailed analyses and periodic updates.

Simplified GBMs: Reducing tree depth or the number of iterations in GBMs can help balance accuracy with computational efficiency, ensuring faster predictions without significant accuracy loss.

Infrastructure Investment: Larger hotel chains with access to cloud-based systems or high-performance hardware can fully leverage GBMs' potential for accurate predictions and actionable insights.

Cost-Benefit Analysis: Smaller hotels must weigh the cost of implementing GBMs against their operational needs. For example, if cancellations have a marginal impact on revenue, **LR may be sufficient;** however, for high-revenue or peak seasons, the investment in GBMs might be justified.

Strategic Applications: Beyond real-time predictions, GBMs can inform long-term strategies, such as identifying seasonal cancellation trends or optimizing marketing efforts.

Conclusion

The trade-offs between Logistic Regression and GBMs highlight the importance of aligning model selection with operational needs. LR offers a lightweight and interpretable solution for real-time, resource-constrained applications, while GBMs provide unparalleled accuracy and adaptability for complex datasets, albeit with higher computational demands. By understanding these trade-offs, hotels can tailor their predictive analytics systems to achieve a balance between efficiency and precision, ensuring optimal decision-making across diverse operational contexts.

After applying Principal Component Analysis

Principal Component Analysis is a dimensionality reduction technique that transforms the dataset into a smaller set of uncorrelated components while retaining as much variance as possible. Its impact on model performance depends on the characteristics of the model itself and how it processes input data.

Model	Accuracy	Precision	Recall	F-Score	AUC
L1 Logistic Regression	0.63	0.69	0.76	0.72	0.64
L2 Logistic Regression	0.63	0.69	0.76	0.72	0.64
XGBoost	0.76	0.77	0.88	0.82	0.863
LightGBM	0.76	0.77	0.88	0.82	0.84

Fig 27 Shows the Metrics after applying PCA to all the models

Fig 27 represents models' performance metrics before and after Principal Component Analysis. This approach allowed us to compare the effectiveness of the strategies before and after dimensionality reduction. PCA reduced the dataset's dimensionality from 33 features to 2 principal components, significantly affecting the models' predictive capabilities.

L1 and L2 Logistic Regression:

Both logistic regression models saw a considerable decline in performance post-PCA.

They achieved an accuracy of 0.63, precision of 0.69, recall of 0.76, F1-score of 0.72, and AUC of 0.64. This decline is attributed to logistic regression's reliance on individual feature importance, which is diminished by PCA's dimensionality reduction.

XGBoost:

XGBoost showed resilience to the dimensionality reduction, maintaining strong performance: It achieved an accuracy of 0.76, precision of 0.77, recall of 0.88, F1-score of 0.82, and AUC of 0.86. XGBoost's tree-based structure allows it to effectively capture interactions between principal components, contributing to its robust performance.

LightGBM:

LightGBM also maintained strong performance metrics after PCA: It achieved an accuracy of 0.76, precision of 0.77, recall of 0.88, F1-score of 0.82, and AUC of 0.84. LightGBM's resilience is attributed to its gradient boosting approach and histogram-based optimization, which enable efficient handling of reduced feature sets.

Following the application of PCA, XGBoost emerged as the top-performing model, slightly outperforming LightGBM with an AUC of 0.86 compared to 0.84. Both these models significantly outperformed logistic regression, demonstrating their ability to maintain high accuracy and reliability despite reduced dimensionality. The resilience of XGBoost and LightGBM can be attributed to their capacity to effectively model complex feature relationships and handle transformed data. XGBoost's robustness in the face of dimensionality reduction makes it an ideal choice for scenarios prioritizing computational efficiency or requiring feature reduction to simplify data analysis. However, the slight decrease in accuracy compared to pre-PCA performance underscores the need for careful consideration when applying dimensionality reduction techniques in practical applications.

Impact of PCA on Logistic Regression vs. GBMs and XGBoost

Why PCA Significantly Impacted Logistic Regression

Dependence on Original Features

Logistic Regression assumes a linear relationship between the predictors (features) and the log-odds of the target variable. This assumption makes it heavily reliant on the original feature structure, especially domain-specific relationships. PCA replaces original features with principal components (linear combinations of the original variables). These components may not align with the features LR depends on for meaningful linear relationships. As a result, Logistic Regression struggles to capture the same patterns and relationships in the transformed feature space.

Loss of Interpretability

PCA removes the intuitive meaning of features like **lead_time**, **adr**, or **previous_cancellations**, which are highly interpretable in the context of hotel booking cancellations. Logistic Regression's coefficients, which quantify feature importance, lose interpretability when applied to principal components. This affects both the interpretative power and predictive performance of Logistic Regression.

Class Imbalance and Discrimination Challenges

The uploaded dataset contains an imbalance between **canceled bookings (Class 1)** and **non-canceled bookings (Class 0)**. LR already struggles with imbalanced datasets, and PCA compounds this issue by removing fine-grained patterns specific to the minority class. PCA emphasizes variance but does not prioritize class separability, leading to degraded performance in discriminating between the two classes. This is evident in the confusion matrix, where Class 1 had a significant number of **false negatives**, reflecting poor recall.

Metrics post-PCA

Accuracy, F1-score, and AUC all dropped significantly for Logistic Regression**. AUC dropped to 0.65**, indicating the model was barely better than random guessing. The confusion matrix revealed a sharp decline in correctly identifying cancellations (True Positives), underscoring the reduced discriminatory power.

Why PCA Had a Smaller Effect on GBMs and XGBoost

Robustness to Feature Transformations

Tree-Based Models: XGBoost and LightGBM are tree-based models that split data at specific thresholds, which are not disrupted by PCA transformations. These models do not assume linearity and are therefore unaffected by the loss of direct feature relationships.
Even after PCA, tree-based algorithms can find splits in the transformed components that align with patterns in the target variable.

Feature Prioritization

GBMs and XGBoost inherently prioritize features during training. While PCA alters the feature space, these models adapt by focusing on the most informative components, mitigating the loss of original feature context.
For example, even if PCA combines **lead_time** and **adr** into a single component, tree-based models can still identify and utilize the combined effect effectively.

Handling of Class Imbalance

Unlike LR, GBMs and XGBoost iteratively focus on hard-to-classify samples, such as the minority class (cancellations). This mechanism compensates for the lack of class-specific guidance during PCA.

Retention of Nonlinear Interactions

XGBoost and LightGBM excel at capturing nonlinear relationships. While PCA discards individual feature identities, these models still capture interactions between the transformed components, preserving much of their predictive power.

Metrics post-PCA

Both XGBoost and LightGBM experienced minor drops in performance **AUC scores** for XGBoost and LightGBM remained relatively high at **0.85**, indicating strong class separability despite PCA. While accuracy and recall for cancellations decreased, the models maintained a reasonable balance between precision and recall, as reflected in their F1-scores.

Model Predictions to Real-World Hotel Scenarios

Adjusting Staffing Levels

Fluctuating booking cancellations result in inefficient staffing, causing overstaffing during low occupancy or understaffing during surges. Predicting the likelihood of cancellations for upcoming bookings. For high cancellation probabilities (e.g., >80%), adjust staffing levels for housekeeping, front desk, and concierge to prevent overstaffing. For low cancellation probabilities (e.g., <20%), prepare for full occupancy by ensuring sufficient staffing and resources. This approach reduces overstaffing costs during low occupancy periods and improves guest satisfaction by ensuring adequate staffing during peak times. For low cancellation probabilities (e.g., <20%), prepare for full occupancy by ensuring sufficient staffing and resources. This approach reduces overstaffing costs during low occupancy periods and improves guest satisfaction by ensuring adequate staffing during peak times.

Optimizing Pricing Strategies

Revenue losses occur when rooms remain vacant due to last-minute cancellations. Used to predicted cancellation probabilities to identify rooms likely to be unoccupied. For high cancellation probabilities, implement dynamic pricing by offering discounts or using overbooking strategies to fill potentially vacant rooms. For low cancellation probabilities, maintain or increase room rates during high-demand periods to maximize revenue. This minimizes revenue losses caused by cancellations and enhances profitability through data-driven pricing strategies.

Enhancing Customer Retention

Frequent cancellations by high value repeat customers may indicate dissatisfaction or hesitancy, resulting in lost revenue. Identifying patterns of cancellation behavior among repeat or high-value customers. Recognize repeat customers with high cancellation probabilities and offer personalized incentives, such as discounts, upgrades, or flexible cancellation policies, to encourage them to retain bookings. Strengthening customer loyalty and retaining high-value guests improves long-term revenue.

Managing Inventory and Room Allocation

Poor inventory management due to cancellations leads to overbooking or revenue loss from unused rooms. Predicting cancellations can adjust room availability. For high cancellation probabilities, reserve these rooms for walk-in customers or last-minute bookings. For low cancellation probabilities, prioritize confirmed bookings and avoid overbooking. This approach balances room inventory management to maximize occupancy while minimizing the risk of overbooking.

Event-Specific Planning

Hotels in tourist-heavy areas experience demand fluctuations during events or holidays.

Analyzing cancellation patterns around events to predict demand volatility. During periods with high cancellations, reduce prices for early bookings to secure reservations and forecast actual occupancy levels to optimize staffing. During periods with low cancellations, increase rates or offer premium packages to capitalize on demand. This ensures optimized pricing, staffing, and promotional strategies tailored to event-specific trends

Improving Marketing Campaigns

Inefficient promotional targeting leads to low conversion rates and wasted marketing resources. Segment customers based on their likelihood of cancellations. For customers with high cancellation probabilities, target them with flexible booking options, discounts, or personalized offers. For customers with low cancellation probabilities, focus marketing on upselling room upgrades, premium packages, or additional services like spa and dining experiences. Personalized marketing campaigns improve efficiency and better align with customer behaviors.

Managing Long-Term Contracts or Group Bookings

Cancellations by corporate clients or group bookings can lead to significant revenue losses. Assess cancellation risks for group bookings or long-term contracts based on customer profiles and trends. For high-risk group bookings, offer flexible terms, such as rescheduling or partial refunds, to retain loyalty. For low-risk clients, allocate premium services to strengthen relationships. This strategy improves the reliability of corporate bookings and reduces the financial impact of cancellations.

Handling Last-Minute Cancellations

Last-minute cancellations disrupt operations and leave rooms unoccupied. Using historical and behavioral data to predict last-minute cancellation trends. Attract same-day bookings with targeted promotions for canceled rooms. Adjust workflows, such as cleaning schedules, to accommodate changes in occupancy. This ensures operational efficiency and minimizes revenue loss from unoccupied rooms.

Addressing Seasonal Trends

Seasonal cancellations are influenced by unpredictable factors like weather or economic conditions. Analyze historical data to predict seasonal patterns in cancellations. Adjust pricing and staffing strategies based on anticipated seasonal demand. Fine-tune overbooking strategies during peak periods to account for cancellations. This approach ensures preparedness for seasonal variations, optimizing operational and financial outcomes.

Streamlining Resource Allocation

Inefficient use of auxiliary resources, such as catering, parking, or conference facilities, arises from unreliable occupancy predictions. Forecast occupancy levels using cancellation probabilities. Dynamically allocate resources like parking spaces or restaurant reservations based on predicted occupancy. Resource waste is minimized during low occupancy, while guest satisfaction is enhanced through efficient resource management during high occupancy periods.

5. Conclusion

The application of machine learning models to the hotel booking cancellations dataset has yielded valuable insights into customer behavior patterns and the factors influencing cancellations. By employing advanced techniques such as logistic regression, XGBoost, and LightGBM, we have achieved robust and interpretable predictions of booking cancellations. These insights empower hotels with actionable strategies to optimize their operations and enhance customer satisfaction.

5.1 Objectives Achieved:

Accurate Prediction of Cancellations:

Through models like XGBoost and LightGBM, we achieved near-perfect classification accuracy, enabling reliable predictions of whether a booking would be canceled.

Understanding Key Drivers of Cancellations

Feature importance analysis highlighted critical factors such as lead time, deposit type, customer type, and total nights, providing hotels with insights into customer behavior and cancellation risks.

Improved Data Preprocessing Pipeline

Handling missing data, encoding categorical variables, and feature scaling enhanced the dataset's quality, laying a strong foundation for predictive modeling.

Optimized Model Selection

Comparative analysis of logistic regression, XGBoost, and LightGBM allowed us to select the most effective algorithm for this dataset, balancing interpretability, accuracy, and scalability.

Dimensionality Reduction Insights:

Principal Component Analysis (PCA) demonstrated its utility in reducing complexity, though at a tradeoff in performance. This highlighted the importance of retaining relevant features for high-performance models like XGBoost.

Insights Gained

Lead Time and Deposit Type are Critical:

Longer lead times and non-refundable deposits were strong predictors of cancellations. This insight can guide policy revisions for flexible bookings to reduce cancellations.

Repeat Guests Have Lower Cancellation Rates

Analysis revealed that repeat guests are less likely to cancel, emphasizing the importance of loyalty programs and personalized services to encourage repeat bookings.

Seasonality and Booking Patterns Matter

Features like arrival date, total nights stayed, and market segment provided insights into seasonal trends and customer preferences, enabling better demand forecasting.

Model Robustness in Real-World Scenarios

XGBoost and LightGBM proved highly effective in handling high-dimensional, imbalanced data without significant preprocessing, showcasing their practical applicability.

5.2 Potential Limitations

While these machine learning models for hotel booking cancellations have provided valuable insights and strategies, it's crucial to acknowledge the limitations of our approach and consider future research directions. This balanced perspective ensures continuous improvement and adaptability of our project outcomes.

Reliance on Historical Data:

Our models, trained solely on historical booking data, may not accurately reflect emerging trends or external factors affecting cancellations. Sudden changes in customer behavior, such as those caused by global events or economic shifts, could significantly reduce predictive accuracy. For instance, post-pandemic travel patterns differ markedly from historical norms, potentially invalidating previous trends.

Dataset Bias:

Inherent biases in the dataset, such as overrepresentation of specific customer segments or booking channels, may skew predictions. This could lead to unfair or suboptimal strategies, potentially underestimating cancellation risks for certain groups. A dataset predominantly featuring domestic travelers, for example, might not accurately predict international guest behavior.

Imbalanced Classes:

The dataset's class imbalance, with cancellations being less frequent than non-cancellations, poses a challenge. Despite high overall accuracy, models may struggle to identify the critical minority class of cancellations. A model with 95% accuracy might excel at predicting non-cancellations while failing to capture actual cancellations effectively.

Model Overfitting:

The perfect accuracy demonstrated by models like XGBoost and LightGBM suggests potential overfitting to the training data. This could limit their ability to generalize to new or unseen data, potentially resulting in decreased performance when applied to different hotel contexts or regions.

Dimensionality Reduction Tradeoffs:

While PCA reduces data complexity, it may inadvertently discard important features. This simplification could lead to a loss of nuanced feature interactions, potentially diminishing the predictive power of high-performing models like XGBoost.

Static Model Nature:

The static nature of our models necessitates periodic retraining to remain relevant as customer behavior evolves. Without ongoing updates to reflect new data trends, predictions may become less reliable over time, limiting the models' long-term utility.

5.3 Directions for Future Research

Incorporating Real-Time Data:

Develop models that adapt to current booking information, providing timely predictions. For instance, incorporate recent cancellations or booking changes to enhance dynamic decision-making processes.

Addressing Dataset Bias:

Implement machine learning techniques that promote fairness or augment data to minimize biases. Ensure the dataset represents a diverse range of customer types, regions, and booking channels for more equitable predictions.

Improving Imbalanced Class Handling:

Apply sophisticated methods such as SMOTE or weighted loss functions to address class imbalance issues. Evaluate model effectiveness using metrics like F1-score or precision-recall AUC, which are more suitable for imbalanced datasets.

Incorporating External Factors:

Expand the dataset to include external variables like weather conditions, local events, economic indicators, or global travel trends. This could involve integrating macroeconomic data to account for fluctuations in travel demand and affordability.

Developing Interpretability Tools:

Utilize frameworks such as SHAP or LIME to gain deeper insights into model decisions. This could help explain why certain bookings are considered high-risk and inform policy refinements based on these insights.

Exploring Advanced Dimensionality Reduction:

Investigate techniques like Recursive Feature Elimination to maintain crucial predictors while reducing data dimensions. Consider alternatives to PCA that selectively eliminate less impactful features.

Adaptive and Continuous Learning:

Develop models capable of incremental learning, updating their parameters with new data over time. For example, an adaptive LightGBM model could incorporate weekly booking data to reflect seasonal patterns.

Clustering and Segmentation:

Employ unsupervised learning to identify distinct customer groups with unique cancellation behaviors. This could involve using k-means clustering to group customers based on factors like lead time, market segment, and cancellation probability.

Cross-Domain Applications:

Broaden the analysis to encompass other areas within hospitality, such as restaurant bookings or event reservations. For instance, apply the models to predict cancellations for events at convention centers or banquet facilities. Broaden the analysis to encompass other areas within hospitality, such as restaurant bookings or event reservations. For instance, apply the models to predict cancellations for events at convention centers or banquet facilities.

By recognizing these limitations and pursuing these research directions, we can significantly enhance the models' robustness and applicability. Addressing biases, improving data representation, and incorporating real-time and external data will make the predictive models more resilient and actionable in dynamic market conditions.

These advancements will enable hotels to accurately predict cancellations and swiftly adapt to changing trends. This will result in data-driven solutions that optimize operations, improve customer satisfaction, and foster sustainable growth, ensuring hotels remain competitive and agile in the evolving hospitality landscape.

Future Prospects

Scalable Implementation: Employing frameworks that support real-time predictions, such as online learning algorithms or streaming platforms, would enable hotels to make immediate, data-driven decisions.

Economic Factors: Currency exchange rates or economic conditions in the guest's country of origin could provide insights into cancellations caused by financial constraints.

Local Events: Features like major events, festivals, or conferences near the hotel could inform models about booking spikes or cancellation risks associated with such occurrences.

Travel Trends: Incorporating data from travel platforms or social media (e.g., sentiment analysis of guest reviews) could enhance the model's ability to capture intent-related cancellation behavior.

Time-Sensitive Features: The proximity of the booking date to the stay date (e.g., cancellations closer to the check-in date) could help identify high-risk windows for cancellations. Seasonal booking patterns, such as holidays or off-peak trends, could improve model accuracy by aligning predictions with cyclical demand shifts.

Real-World Applications and Integration

Dynamic Pricing Systems: Enhanced models could be integrated into pricing algorithms, offering discounts or package deals to high-risk bookings in real-time to secure reservations.

Cancellation Insurance Options: Predictive insights could inform the development of cancellation insurance policies, allowing hotels to manage revenue risks more effectively.

Cloud-Based Implementation: Deploying models on cloud platforms could facilitate real-time scalability and integration with booking systems across multiple properties.

By integrating real-time data and additional contextual features, future research could further improve model accuracy, adaptability, and practicality. These enhancements would allow hotels to proactively manage cancellations, optimize resources, and strengthen their competitive edge in a dynamic industry.

5.4 Recommendations for the Hospitality Industry

Dynamic Overbooking Policies:

Use cancellation probabilities predicted by these models to fine-tune overbooking levels. For instance, high-risk bookings (e.g., long lead_time, no deposits) can be offset with overbooking strategies to maximize occupancy and reduce revenue loss.

Personalized Guest Retention Strategies:

Offer tailored incentives, such as discounts, flexible cancellation policies, or personalized communication, to guests identified as high-risk for cancellations. Logistic Regression's interpretable outputs can assist managers in crafting specific retention plans.

Optimized Resource Allocation:

Leverage feature insights (e.g., booking seasonality and guest types) to allocate resources effectively. For example, staffing levels, inventory, and amenities can be adjusted during peak periods of cancellations.

Focus on High-Risk Segments:

Target efforts on market segments with high cancellation rates, such as OTA bookings or transient guests. Develop segment-specific strategies, such as stricter cancellation policies or loyalty programs for repeat guests.

Enhance Revenue Security with Deposit Policies:

Encourage bookings with refundable or non-refundable deposits, which are associated with lower cancellation risks. For instance, offering discounts for refundable deposits can strike a balance between flexibility and revenue security.

Real-Time Predictive Systems:

Integrate machine learning models with booking platforms to enable real-time predictions and interventions. GBMs, with their adaptability to dynamic data, can help monitor cancellation risks and adjust pricing or policies instantly.

Long-Term Insights for Strategic Planning:

Use seasonal trends and feature importance rankings to inform marketing campaigns and promotional strategies. For example, during high-demand months, focus on securing bookings with low-risk profiles to optimize revenue stability.

6.Abstract and References

Abstract:

This project addresses the significant challenge of hotel booking cancellations in the hospitality industry by applying advanced machine learning techniques. Utilizing logistic regression, XGBoost, and LightGBM models, the study achieved exceptional classification accuracy in predicting cancellations. Key factors influencing cancellations were identified, including lead time, deposit type, and customer type. The research also explored dimensionality reduction through PCA to balance model complexity and performance. These insights enable hotels to implement data-driven strategies such as dynamic pricing and targeted loyalty programs to mitigate revenue loss and improve customer satisfaction. However, the study acknowledges limitations like reliance on historical data and potential dataset biases, highlighting the need for future research to incorporate real-time data and address these challenges. Overall, this work demonstrates the transformative potential of machine learning in optimizing hotel operations and fostering sustainable growth in the hospitality sector.

Reference:

1.Dawood, Muhammad. "Hotel Booking Cancellations." *Kaggle*. Accessed 29 Nov. 2024. https://www.kaggle.com/datasets/muhammaddawood42/hotel-booking-cancelations.
2. Gao, R. (2022). Investigation on credit card customer cancellations: A case study of AMEX credit cards. *BCP Business & Management*.

3. Caigny, A., Coussement, K., & Bock, K. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *Eur. J. Oper. Res.*, 269, 760-772.

4. Dumitrescu, E., Hué, S., Hurlin, C., & Tokpavi, S. (2021). Machine learning for credit scoring: Improving logistic regression with non-linear -tree effects. *Eur. J. Oper. Res.*, 297, 1178-1192.

5. Rodrigues, G., Ortega, E., & Cordeiro, G. (2023). New Partially Linear Regression and Machine Learning Models Applied to Agronomic Data. *Axioms.*

6.Putro, N., Septian, R, Widiastuti, W., Maulidah, M., & Pardede, H. (2021). PREDICTION OF HOTEL BOOKING CANCELLATION USING DEEP NEURAL NETWORK AND LOGISTIC REGRESSION ALGORITHM. *Jurnal Techno Nusa Mandiri*.

7.Lin, Y. (2023). Research on the Influencing Factors of Cancellation of Hotel Reservations. *Highlights in Science, Engineering and Technology.*

8.Chen, Y., Ding, C., Ye, H., & Zhou, Y. (2022). Comparison and Analysis of Machine Learning Models to Predict Hotel Booking Cancellation. *Proceedings of the 2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022).*

9.Satu, M., Ahammed, K., & Abedin, M. (2020). Performance Analysis of Machine Learning Techniques to Predict Hotel booking Cancellations in Hospitality Industry. *2020 23rd International Conference on Computer and Information Technology (ICCIT)*.

10. Rusakova, E., & Radionova, M. (2021). Predicting hotel booking cancellation: A comparative analysis of models. *Вестник Пермского университета. Серия «Экономика» = Perm University Herald. ECONOMY.*

11. Li, L., Cui, X., Yang, J., Wu, X., & Zhao, G. (2023). Using feature optimization and LightGBM algorithm to predict the clinical pregnancy outcomes after in vitro fertilization. *Frontiers in Endocrinology*, 14.

12. Hanif, I. (2020). Implementing Extreme Gradient Boosting (XGBoost) Classifier to Improve Customer Churn Prediction.

13.https://www.researchgate.net/publication/310504011_Predicting_Hotel_Booking_Cancellation_to_Decrease_Uncertainty_and_Increase_Revenue