Hotel Booking Cancellation Prediction

GitHub: https://github.com/Bhagya-Ram-7/hotel_cancelation_prediction

Bhagya Ram

bhram@ucsd.edu

Reema Alsaeed

ralsaeed@ucsd.edu

Introduction

In the hospitality industry, managing hotel reservations efficiently is a crucial component to maximizing revenue and resource allocation. One of the biggest challenges hotel management faces is booking cancellations. This leads to financial losses as the hotel is left with an inefficient room utilization. Therefore, predicting whether a hotel booking will be cancelled in advance can help hotels implement effective strategies to improve customer retention while minimizing loss and optimizing revenues.

Our project aims to utilize a “Hotel Booking Demand” dataset to uncover patterns, and build models to predict hotel booking cancellations based on a variety of customer and reservation attributes. By analyzing these factors, we seek to identify key patterns that influence a customer’s likelihood of cancelling their booking.

Research Problem

The research question we plan to address in our project is:

Can we predict whether a hotel booking will be cancelled based on customer and reservation attributes?

Predicting cancellations can help hotels make data-driven decisions, such as

offering incentives or implementing strategies, and managing overbooking risks effectively.

Hypothesis

We hypothesize the following:

First, longer lead times, which is the time between booking and arrival, are associated with a higher likelihood of booking cancellations. Second, customers with a history of previous booking cancellations are more likely to cancel again. Third, bookings made through a specific market segment, such as online travel agencies, have a higher cancellation rate. We are uncertain which segments might have a higher likelihood of canceling. We plan to explore that in our exploratory data analysis section. By testing these hypotheses, we aim to gain insights into customer behaviors, and improve hotel strategies and revenues.

Dataset

Identify Dataset

For our project, we will use the “Hotel Book Demand” dataset from Kaggle, which contains over 119,000 observations, each representing a hotel reservation collected from two hotels: City hotel and Resort hotel.

The dataset provides us with detailed information about hotel reservations from 2015 to 2017, including customers’ information and booking details. The following are some of the variables in the dataset:

- Booking lead time: number of days between booking and arrival.

- Number of adults, children, and babies in the booking.
- Number of weekend and weekday nights in the booking
- Whether a customer is a repeated guest or not.
- Number of previous bookings that were cancelled and not cancelled.
- Reserved and assigned room types.
- Number of car parking spaces required by the customer
- Number of special requests made by the customer.

Those are a few of the many features this dataset provides us. With the large sample size of over 119,000 bookings, and the variety of features, this dataset allows us to develop a predictive model that is adept at determining if a booking is likely to be cancelled.

Exploratory Data Analysis (EDA)

Basic Statistics

Our dataset contains 119,390 instances, each with 32 attributes that can help us predict the likelihood of a hotel booking cancellation.

Data Types

The following is a breakdown of columns types:

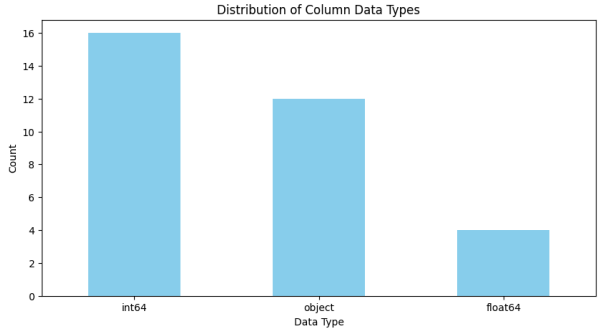

Fig. 1: **Column type breakdown**

By looking at the distribution of the data types, it is clear that most of our columns are integer-based numerical features, either int64 or float64. The plot suggests that most features have discrete numerical values, which could be counts, identifiers, or binary values such as “is_repeated_guest”. The float64 columns are likely to be continuous numerical values. Meanwhile the object type typically is categorical data, which represents a small portion of our dataset. These categorical values will most likely require encoding to numerical values as most machine learning models are designed to handle numerical data.

Check Missing Values

The following shows the number of missing values for each of the 32 columns we have:

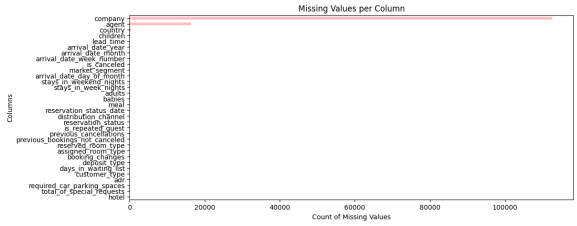

Fig. 2: **Missing Values Count per Column**

We have the following missing values:

- Company: 112593
- Agent: 16340
- Country: 488
- Children: 4

Since over 90% of the column “company” is missing, we decided to drop the column as we believe it is unsuitable for analysis. The “agent” column is the ID of the agent that made the booking, considering the high number of missing values, we have decided to drop the column. For the “country” column, we do not believe this column would be helpful for our model. So, we decided to drop the column. Lastly, for the “children” column, we have 4 missing values, which we filled with 0 as we assume it indicates no children were present in those bookings. This is how we handled missing values for our baseline models, Logistic Regression and CatBoost. However, for our final model, we ended up using LightGBM and the RandomForestClassifier, both of which handle NaNs naturally. By investigating and addressing the missing values, we ensure

that our data is robust and ready for analysis.

Interesting findings

During our exploratory data analysis (EDA), we discovered several key insights from the dataset that can help us build our model and feature selection. The following are some of the insights we uncovered:
The plot in Figure 3 illustrates the monthly cancelation rate of bookings aggregated over the years 2015 - 2017. The blue bars represent the cancellation rate per month, while the red line exhibits the trend. Cancellations generally appear to increase from January to a peak around May-June, followed by a slight decline through the later months. This pattern suggests a seasonal trend in booking cancellations, potentially influenced by higher periods of travel such as summer vacations.

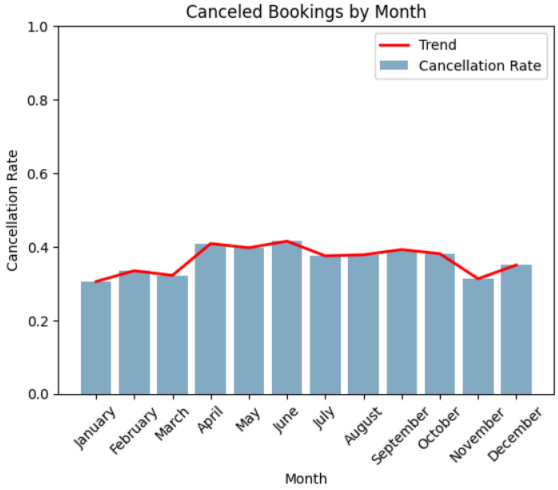

Fig. 3: **Monthly Cancellation Patterns**

Next, we wanted to visualize the relationship between lead time and cancellations.

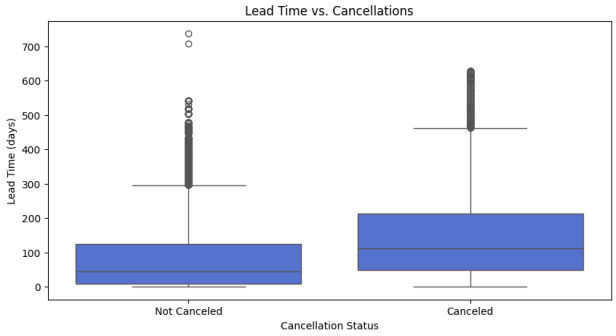

Fig. 4: **Lead Time and Cancellations Boxplot**

The boxplot comparison shows that bookings with longer lead times are more likely to be cancelled. This suggests that uncertainty increases over time, making lead time a crucial feature in predicting cancellation.

Then, we examined the cancellations across the different market segments

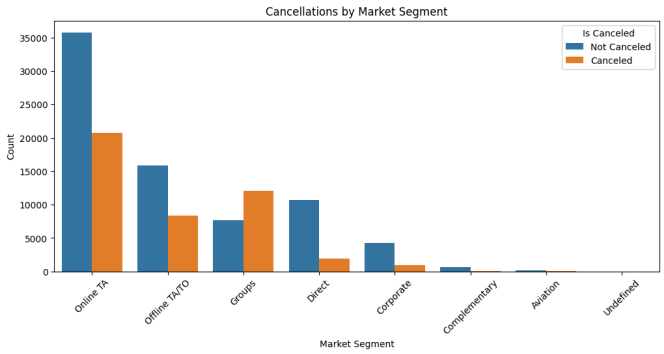

Fig. 5: **Market Segments and Cancellations**

Looking at the plot, we see that online travel agencieS (TA) and group booking show significantly higher cancellation rates compared to other market segments, with offline TA/TO close to them with a moderate cancellation rate. This indicates that source of booking could be a strong predictive factor, as certain market segments exhibit higher likelihood of cancellation than others.

Afterward, we created a histogram of lead time which reveals a right-skewed distribution, with most bookings made close to the arrival date, but we still have a

number of bookings made several months in advance.

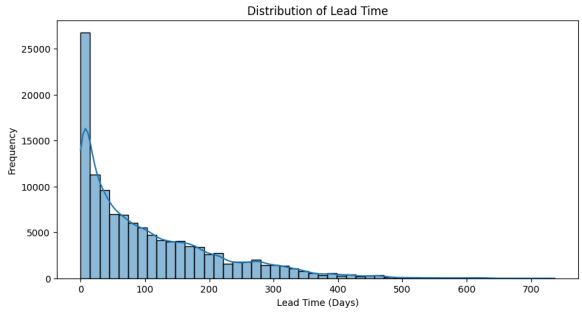

Fig. 6: **Distribution of Lead Time**

We also analyzed the monthly number of visitors for each hotel.

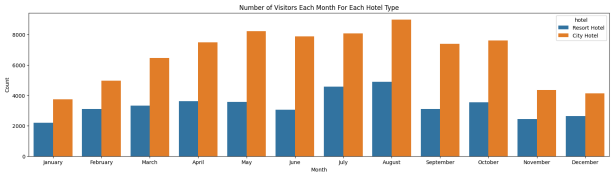

Fig. 7: **Monthly Visitors Count For Each Hotel**

This shows us that throughout the year, the city hotel maintains a higher demand than the resort hotel. On the other hand, the resort hotel experiences peak bookings in the summer, July-August, but it is still lower than the city hotel.

Additionally, we wanted to generate a simple count plot of cancellations.

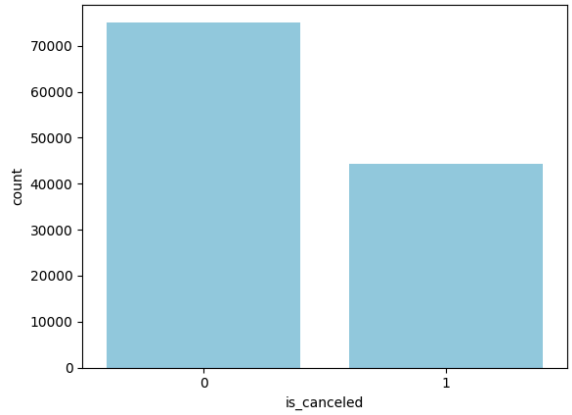

Fig. 8: **Cancellations Count Plot**

This shows us that around a third of the bookings in the dataset were cancelled (is_canceled = 1). This can cause major financial losses to the hotels. This reinforces the importance of developing a predictive model to help hotels manage cancellations effectively. Also, this tells us that there is a class imbalance in our dataset, which is a key insight that can help us determine certain aspects of our model design, such as what evaluation metrics we should use.

With these insights, we can move forward to formulating the predictive task. The patterns observed in the EDA, will directly guide our feature selection and model design.

Predictive task

We aim to classify bookings as likely to be canceled (Class 1) or not (Class 0) based on the available customer and booking attributes. We will evaluate the different models on key metrics such as their precision, recall, F1-score and accuracy across classes.

Baseline Model 1: Logistic Regression

We chose to begin with logistic regression because it excels at binary classification. However, it may struggle with feature interaction terms, which more complex models would be better suited to handle. In our model, the target variable, *is_canceled*, was removed from the dataset. Additionally, *reservation_status* was excluded, as it contained values such as Check-Out, Canceled, and No-Show, which directly indicate the outcome, leading to data leakage. The dataset was split into 80% training and 20% test data. A Logistic Regression model was trained on the data. The table below reflects the model's performance across key evaluation metrics.

Metric	Class 0 (Not Canceled)	Class 1 (Canceled)
Precision	0.79	0.81
Recall	0.92	0.57
F1-Score	0.85	0.67

As seen above, precision and recall vary across classes, indicating that the model performs better at identifying non-canceled reservations than canceled ones. The overall accuracy was found to be 0.79, meaning that correctly classified cancellations and non-cancellations 79% of the time.

Baseline Model 2: CatBoostClassifier

For our second baseline model, we trained a CatBoost classifier because of its ability to handle categorical features efficiently. Unlike our previous model which required explicit one-hot encoding, CatBoost automatically processes these variables. Like before, we split the dataset into 80% training and 20% testing while keeping the CatBoost model at 100 iterations with a depth of 6. The table below compares both models' performance across key evaluation metrics.

Model	LR		CB	
Metric	Class 0	Class 1	Class 0	Class 1
Precision	0.79	0.81	**0.82**	**0.86**
Recall	0.92	0.57	**0.94**	**0.64**
F1-Score	0.85	0.67	**0.87**	**0.74**

As seen above, CatBoost outperforms Logistic Regression across all key metrics. The overall accuracy was found to be 0.83. The main drawback of Logistic Regression is its assumption of a linear relationship between the features and the target variable. CatBoost is a more complex and flexible model. Further, it is a decision tree based model and so it typically achieves higher accuracy due to its ability to capture nonlinear interactions.

Model

We plan on comparing the baseline models with the following advanced ensemble models:

- RandomForestClassifier
- LightGBM

While CatBoost is already a gradient boosting decision tree model, we expect Random Forest's bagging approach and alternative boosting methods like LightGBM to potentially offer improvements. Random Forest builds independent trees in parallel, which might provide better generalization. LightGBM is known to efficiently handle large-scale data and has a high training speed.
We began by training a RandomForestClassifier. We dropped the columns that contained several missing values (*agent, company, country*), as well as the columns that would lead to leakage (*reservation_status*). As we did for the baseline, we filled in the missing values of *children* with 0. We one-hot encoded the categorical columns. This resulted in the following model performance.

Metric	Class 0 (Not Canceled)	Class 1 (Canceled)
Precision	0.87	0.87
Recall	0.93	0.76
F1-Score	0.90	0.81

The RandomForestClassifier was found to have an overall accuracy of 0.87, compared to CatBoost's 0.83. It outperforms CatBoost because it averages the predictions of multiple trees, which reduces the overall variance.
We wanted to employ feature engineering to optimize our model further. As we did for the previous model, we dropped columns that would lead to data leakage or were irrelevant. Instead of dropping *reservation_status_date* as we did last time, we extracted the month value from it. We used a label encoder to automatically

convert categorical values into numerical representations. For example, in the *meal* column, which contains unique values such as BB, HB, SC, FB, and Undefined, the encoder assigned integers as follows: {'BB': 0, 'HB': 1, 'SC': 2, 'FB': 3, 'Undefined': 4}. Next, we created new features to allow the model to understand the patterns better:

- *total_nights*: We can assume that longer stays tend to be more expensive. Thus, the cancellation rates associated with these are lower.
- *cancellation_ratio*: This was calculated by dividing the *previous_cancellations* by *lead_time*, which is the number of days that elapsed between the entering date of the booking into the property management system and the arrival date.
- *special_requests_bool*: This is a one-hot encoded column. 0=>cases with 0 special requests at the time of booking, 1=> cases with at least 1 special request. We assumed guests with a greater number of requests would be less likely to cancel.

We log-transformed the *adr* column, which represents the average daily rate. This column had an incredibly skewed distribution, with values ranging from 0 to 5400. These techniques optimized our model and led to a much higher accuracy of 0.89.

Metric	Class 0 (Not Canceled)	Class 1 (Canceled)
Precision	0.86	0.95
Recall	0.97	0.75
F1-Score	0.92	0.83

The model that afforded us the best accuracy was the LightGBM model. We utilized the exact same data transformations as we did in RF. This resulted in an overall test accuracy of 0.98. The table below summarizes the model's performance.

Metric	Class 0 (Not Canceled)	Class 1 (Canceled)
Precision	0.96	1
Recall	1	0.94
F1-Score	0.98	0.97

Literature

Exploratory Data Analysis of Bookings and Machine Learning to Predict Cancellations By Marcus Wingen

https://www.kaggle.com/code/marcuswingen/eda-of-bookings-and-ml-to-predict-cancelations

Wingen's notebook explored booking data through EDA, analyzing patterns in cancellations. Our EDA is mostly different, but we drew inspiration from Wingen's work. Reviewing their findings alongside ours helped us understand the significance of variables like lead time and market segment and deepened our understanding of key relationships in the dataset.

Hotel Booking Demand Datasets by Antonio, Almeida and Nunes

https://www.sciencedirect.com/science/article/pii/S2352340918315191

This article provides a detailed breakdown of the dataset we used in our project, describing the meaning of each feature, its data type, and how it was collected from hotel management systems. Also, it includes the unique counts for categorical variables and summary statistics, such as mean, standard deviation, median, and percentiles for the numerical variables, presenting them separately for the two hotels in the dataset. This serves as an essential reference for our understanding of the dataset's structure and ensuring

accurate preprocessing. By analyzing this article, we are able to understand the dataset better, determine which features are most relevant for our model, and ensure our data cleaning aligns with the original dataset's intended use.

Hotel Booking Prediction By Nitesh Yadav
https://www.kaggle.com/code/niteshyadav3103/hotel-booking-prediction-99-5-acc/notebook

Yadav's notebook included many different machine learning models, including logistic regression, decision trees, and ensemble methods, to predict hotel booking cancellations using our dataset. We used their notebook as a benchmark for high performing models, but our approach differs in multiple ways. First, we chose to only log-transform *adr* while Yadav log-transformed *agent*, *lead_time*, *company* and certain arrival-related features as well. Second, we automated the label encoding process for categorical features to ensure the model would be able to deal with new categorical features that appeared in the unseen data. On the other hand, Yadav's analysis manually mapped each of the unique values in every categorical column to integers. This can be time-intensive, and may backfire if new categorical columns are introduced to the test data. Yadav also imputed the missing values in *adr* with the column mean, but we did not find any missing values in this column. By analyzing their work, we identified useful modeling techniques and used them as inspirations for our model, such as creating a new feature called *total_nights*, that summed the total *stays_in_weekend_nights* and *stays_in_week_nights*.

Novelty of Our Work

While past studies have analyzed hotel booking and developed cancellation prediction models, our work differs in taking into account whether a customer has previously canceled their booking. We also built a label encoder, so any categorical columns that are newly introduced in the test data are handled effectively. Additionally, unlike previous studies that mainly relied on accuracy, our model evaluation incorporates precision, recall, and F1-score to provide a more comprehensive assessment of cancellation predictions. This ensures that our model is not only accurate but also effective at identifying true cancellations while minimizing false predictions, which makes it more practical for real-world applications.

By integrating these techniques, our model aims to provide greater accuracy in predicting the likelihood of a booking cancellation, ensuring that we provide a practical solution for hotel revenue management.

Comparison with Previous Findings

Our findings align with previous findings in several ways. Similar to Wingen's EDA, we found that lead time and market segment are among the most significant factors in cancellation prediction. However, our analysis also reveals other features that are more significant than lead time and market segment (see Figure 9). Furthemore, while Yadav's model achieved a high accuracy, accuracy alone does not fully capture the model's effectiveness in predicting cancellations, especially given that we have a class imbalance in our dataset. So, in our approach we have used accuracy alongside precision, recall, and F1-score. This provides a deeper understanding of how well the model distinguishes between cancellations and non-cancellations.

Results

Comparison

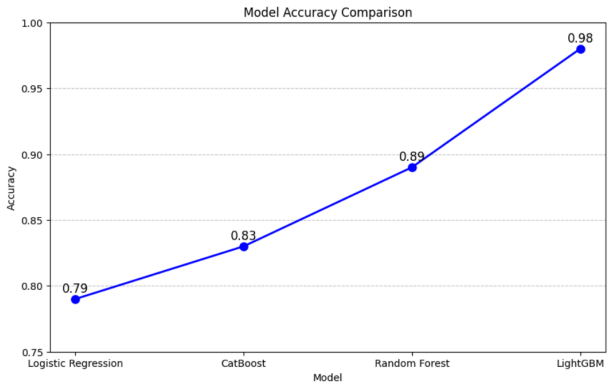

Fig. 9: **Model Accuracy Comparison**

Comparison of Baseline Model 1: Logistic Regression with final model, LightGBM

Model	LR		Light GBM	
Metric	Class 0	Class 1	Class 0	Class 1
Precision	0.79	0.81	**0.96**	**1**
Recall	0.92	0.57	**1**	**0.94**
F1-Score	0.85	0.67	**0.98**	**0.97**
Accuracy	0.79		**0.98**	

Comparison of Baseline Model 2: CatBoostClassifier with LightGBM

Model	CB		Light GBM	
Metric	Class 0	Class 1	Class 0	Class 1
Precision	0.82	0.86	**0.96**	**1**
Recall	0.94	0.64	**1**	**0.94**
F1-Score	0.87	0.74	**0.98**	**0.97**
Accuracy	0.83		**0.98**	

In both tables above, the bolded values on the right highlight the improved performance of the LightGBM model. LightGBM is a gradient boosting technique that is well suited for large datasets such as ours. It utilizes decision trees to understand complex, non-linear relationships between the features and our target variable, *is_canceled*. The feature engineering, log transformation and label encoding we incorporated into our model helped significantly raise its accuracy.

Similar Models Comparison

Our model achieved an accuracy of 0.98, which is slightly lower than Nitesh Yadav's 0.99, which was achieved by training an artificial neural network. While the accuracy difference is minimal, our model is likely to be less resource-intensive and more interpretable, thus being more feasible for hotel management.

Limitations

We conducted a feature importance analysis of our LightGBM model, which revealed that *reservation_status_date*, *arrival_date_month*, *arrival_date_day_of_month* and *lead_time* are among the most important features in predicting the likelihood of cancellation. This means that several missing values in any of these columns would lower the predictive power of our model. Furthermore, our data is incomplete: We have several missing values in company and agent, as highlighted earlier in the EDA section. Our feature analysis shows that agent is the 7th most important feature in determining the cancellation, so having complete values in this column might have boosted our model's accuracy.

Hyperparameters

These are the hyperparameters of our LightGBM model, which had the highest accuracy overall.

- n_estimators=350: Number of trees. Increasing this improves the training accuracy, but at the risk of overfitting. The default number of trees is 100.
- max_depth=15: Maximum depth of each tree. Increasing the depth can help reduce overfitting by helping the model understand more patterns. The default max_depth is 6.
- learning_rate=0.05: Rate at which the model learns, or the step size. The higher the learning rate is, the higher the chance of failing to capture patterns.
- num_leaves=64: Maximum leaves per tree. Similar to max_depth, increasing this helps the model capture data patterns better but might lead to overfitting.
- colsample_bytree=0.8: Number of features used in every tree.
- subsample=0.8: Subset percentage of data used to train each tree.
- min_child_samples=20: Minimum samples per leaf. This helps the model generalize well to unseen data.
- reg_lambda=0.3: Lambda regularization (L2) to penalize against overfitting.
- random_state=42: For reproducibility

We reached these hyperparameters after many trials of extensive hyperparameter tuning.

Effectiveness

Despite the fact we added more features to our model compared to the baseline models, not every feature enhances the model's performance. Some of them did not have a significant impact on our model. For instance, *special_requests_bool* was not used in the baseline but we added it to our model. However it did not significantly improve our model. Nonetheless, the addition of other features helped achieve a high performance for our model.

Major Takeaways

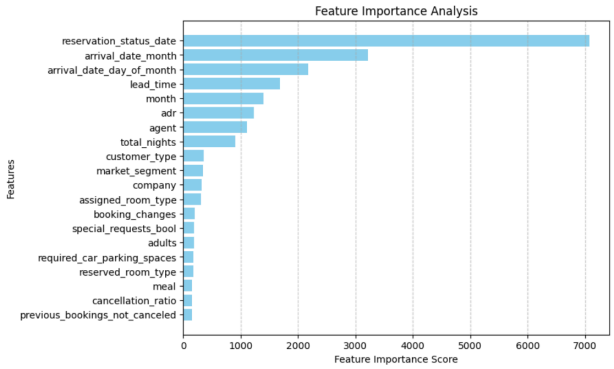

Fig. 9: **Feature Importance Analysis**

Features such as *reservation_status_date* and *arrival_date_month* were found to be major predictors of cancellation. Certain features such as a customer's indicated meal type and *cancellation_ratio* proved to be less indicative of their likelihood of canceling their hotel reservation.

Additionally, we tried different combinations of features, feature engineering and hyperparameters during our model optimization phase. This approach deepened our understanding of the complexity of data analysis and model development, which allowed us to enhance our model performance more effectively.