Predicting Hotel Booking Cancellations using ML

Project: Supervised Learning- Classification - INN Hotels

Sanjana Addanki
11/14/2025

AGENDA

- Executive Summary
- Business Problem & Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

- INN Hotels facing major losses from cancellations
  - 32.8% cancellations
- Cancellations:
  - Higher for high lead times and higher room prices
  - Vary by market segment (online has highest no. of cancellations)
  - Repeated guests are not very likely to cancel (16/930)
- Machine Learning model reached around 80% accuracy and recall of 0.74 (a significant improvement) with optimized thresholds
- After pruning, decision tree showed 0.85 recall

Business Problem

Cancellations cause:

- Loss of revenue
- More spending on marketing and distribution
- Poor resource planning
- Overall lower occupancy

Which is why INN Hotels wants to predict cancellations in advance, to avoid running into these issues and escalating the situation further.

Solution Approach

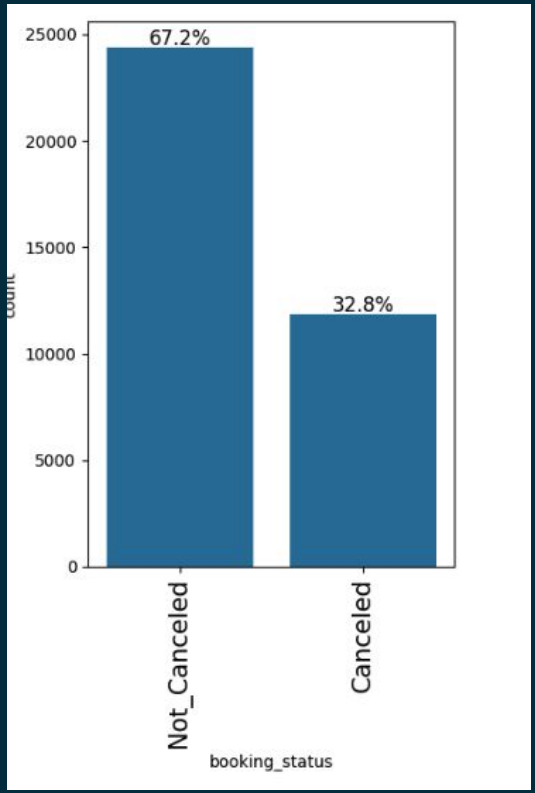
To help solve the problem, we perform

- Exploratory Data Analysis
- Preprocessing (including outliers, encoding, and splitting)
- Logistic Regression
- Threshold Tuning (with AUC-ROC and precision-recall)
- Decision Tree modeling along with pruning
- Comparison between final models

Data Overview

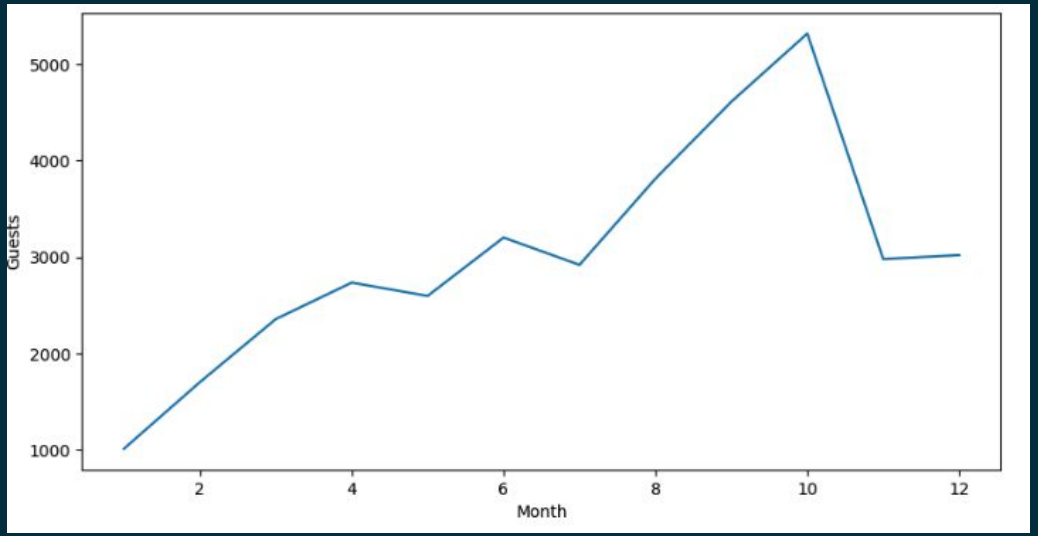
- 36,275 rows x 19 columns

- Balance of class:
  - Canceled: 32.8%
  - Not Canceled: 67.2%



EDA

Monthly Booking Demand & Cancellation Rates



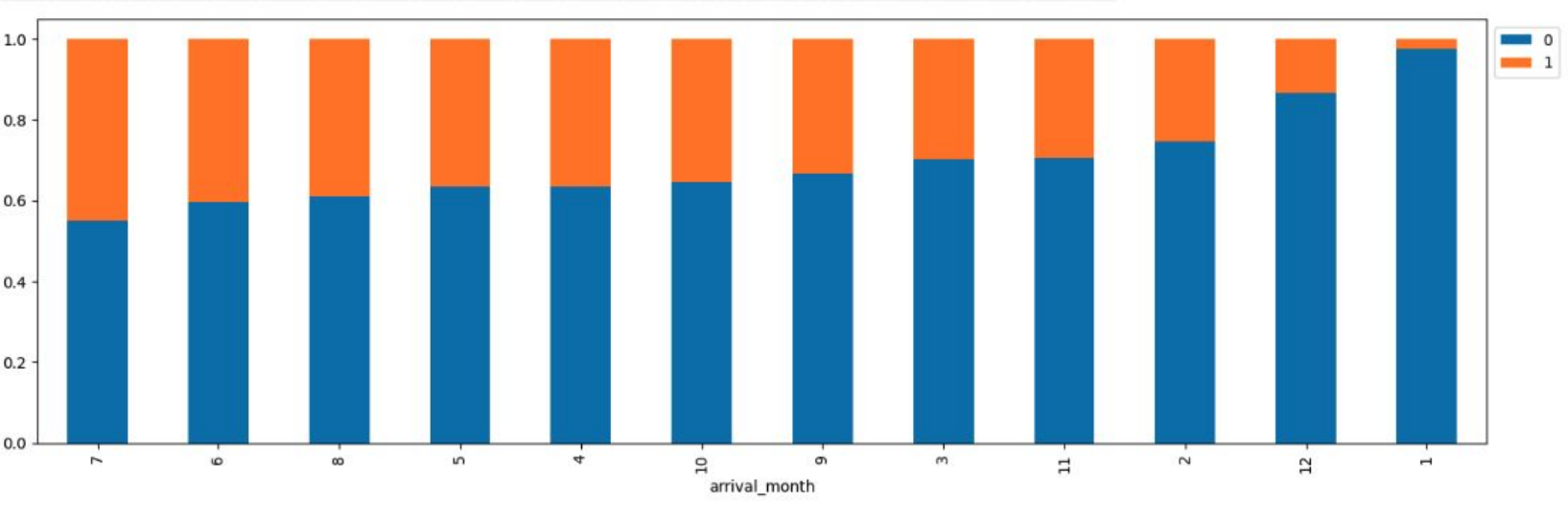
Peak Months:

- August
- September
- October (highest)

The lowest demand appears to be in January, with guests at about 1000, compared to the peak in October of over 5000 guests.

EDA

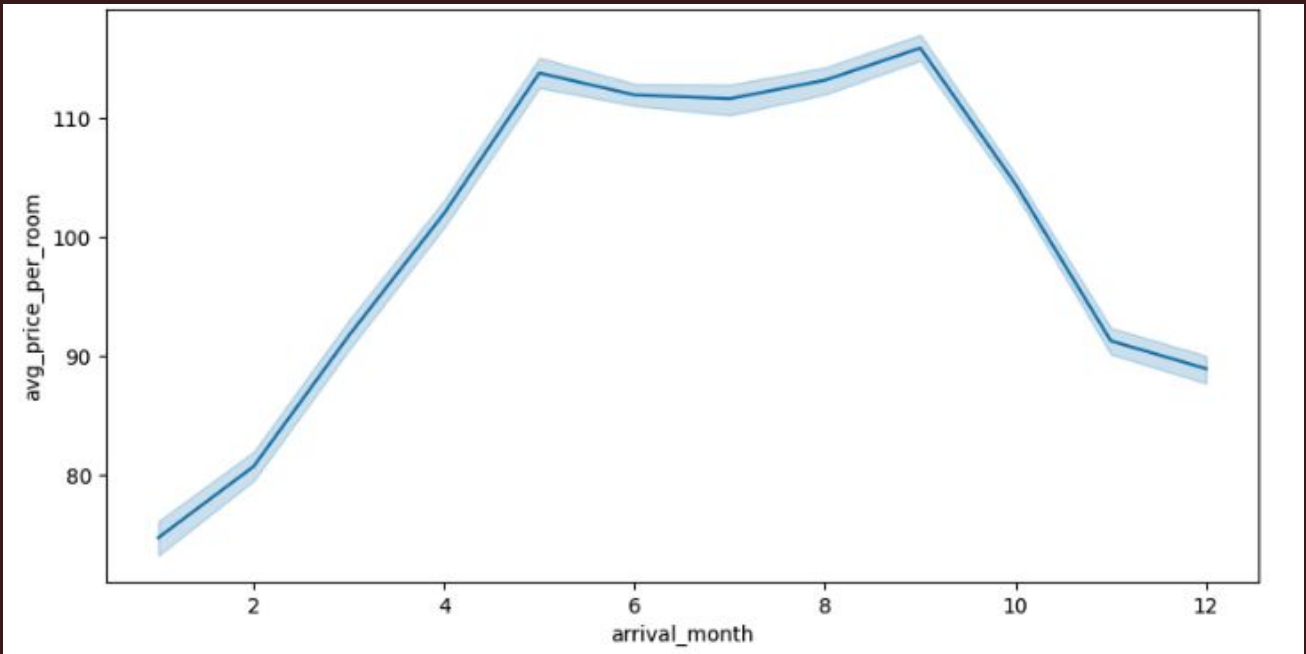
Monthly Cancellation Rates



- Highest number of cancellations in July, August, September, and October
  - These also happen to be the highest booking months as seen earlier

EDA

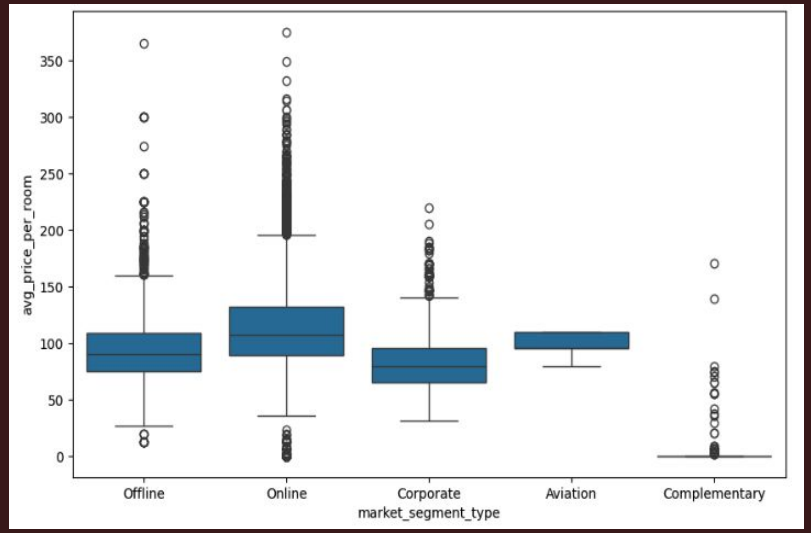
Price Trends by Month

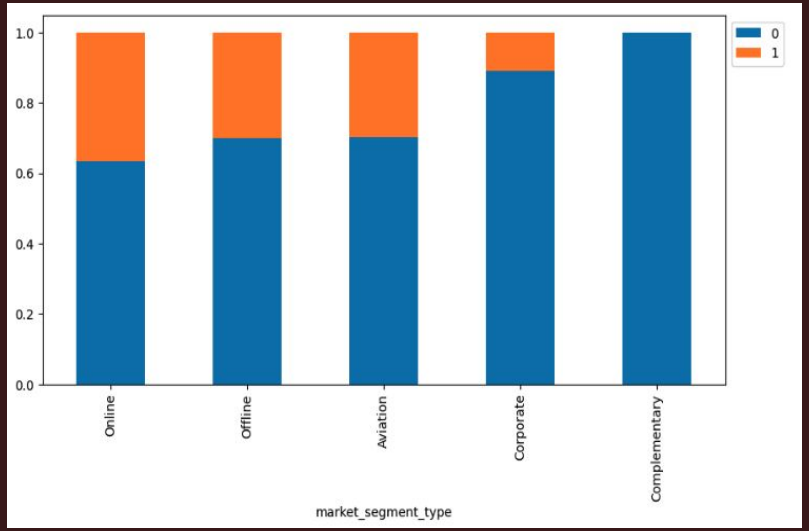


- We observe that there are dynamic spikes in pricing during the peak months (July/August - October)
  - Explains the correlation between cancellations and price

EDA

Market Segment Analysis

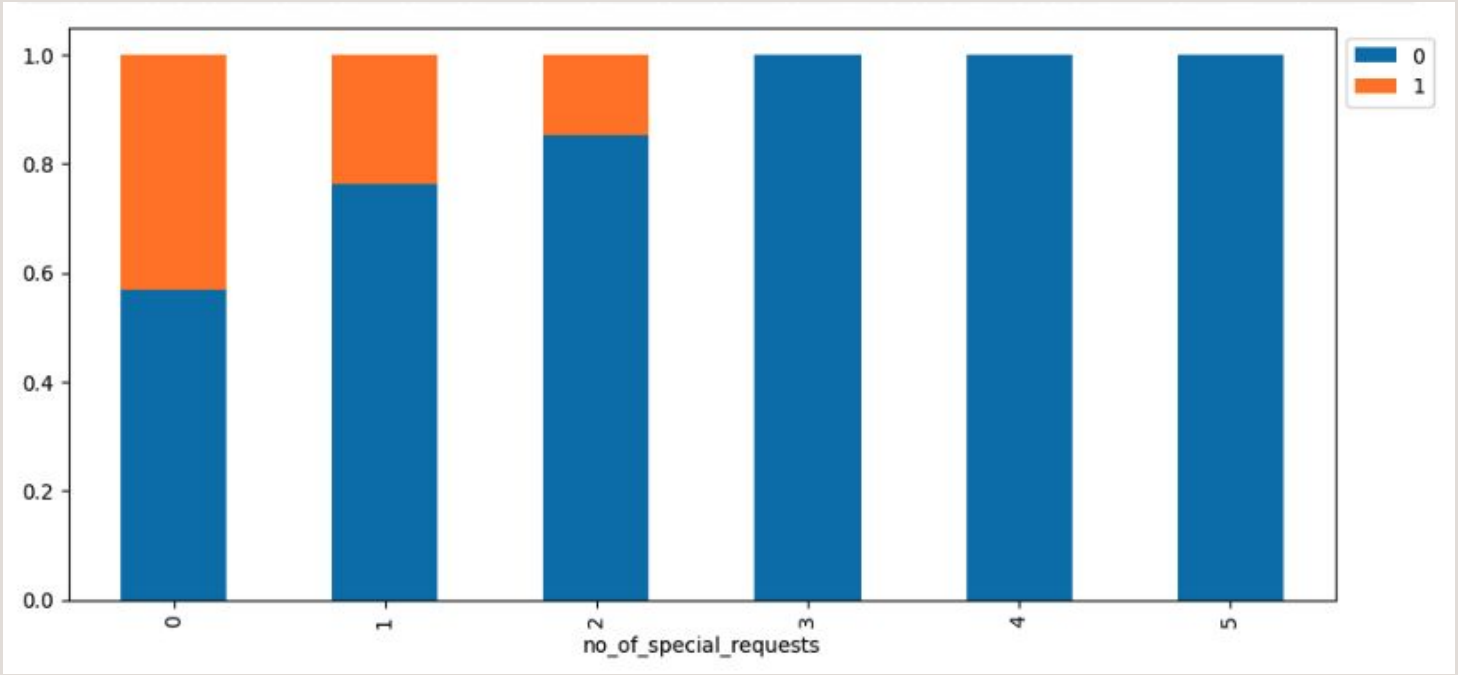




- We can observe that the online segment has significantly more cancellations in both graphs.
  - The more price sensitive and stable segment would be the offline segment.

EDA

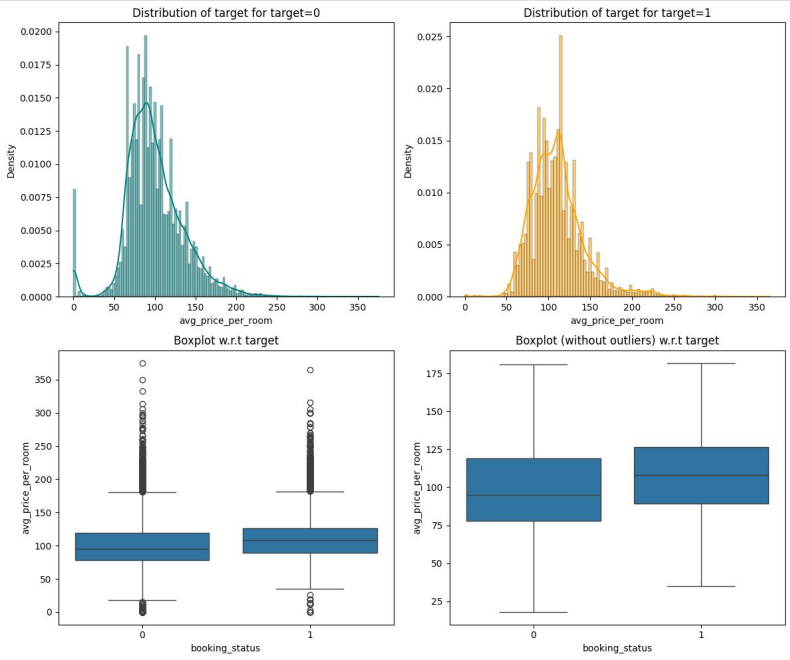
Special Requests & Cancellations

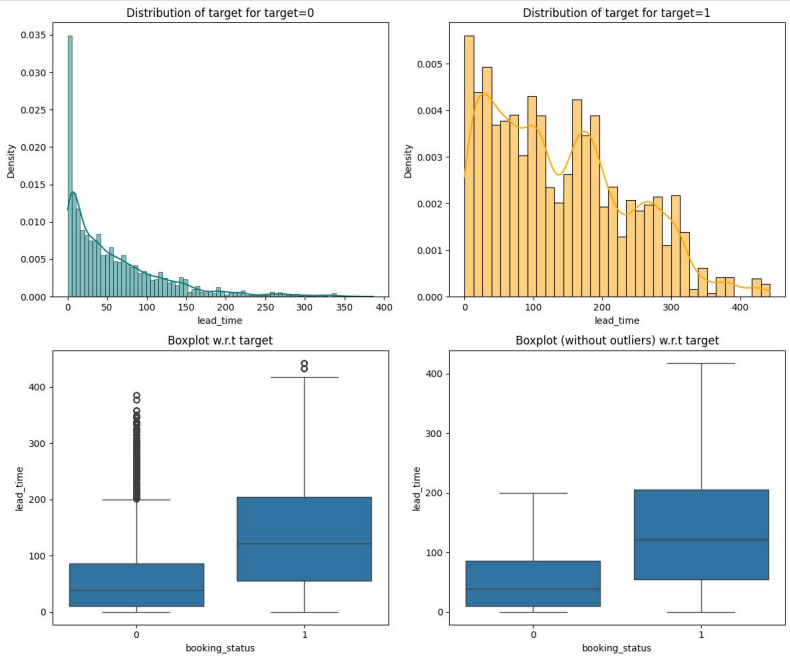


- As the number of special requests increase, there seem to be fewer cancellations.
- This could mean that customers with special requests are more committed to their booking.

EDA

Lead Time & Price Effects

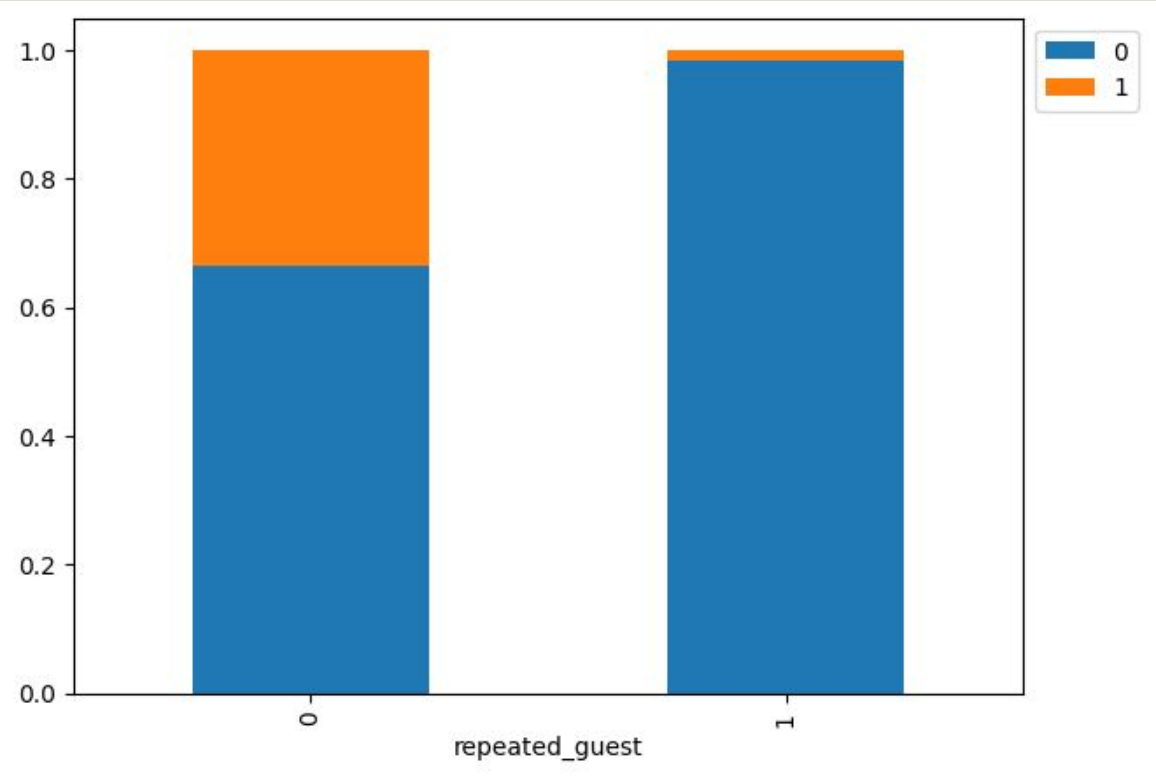




- We can confirm from both distribution graphs that higher prices and higher lead time mean more cancellations

EDA

Repeated Guests



We can observe from this graph that repeated guests seem to be very reliable. This essentially supports the loyalty program strategy.

Data Preprocessing

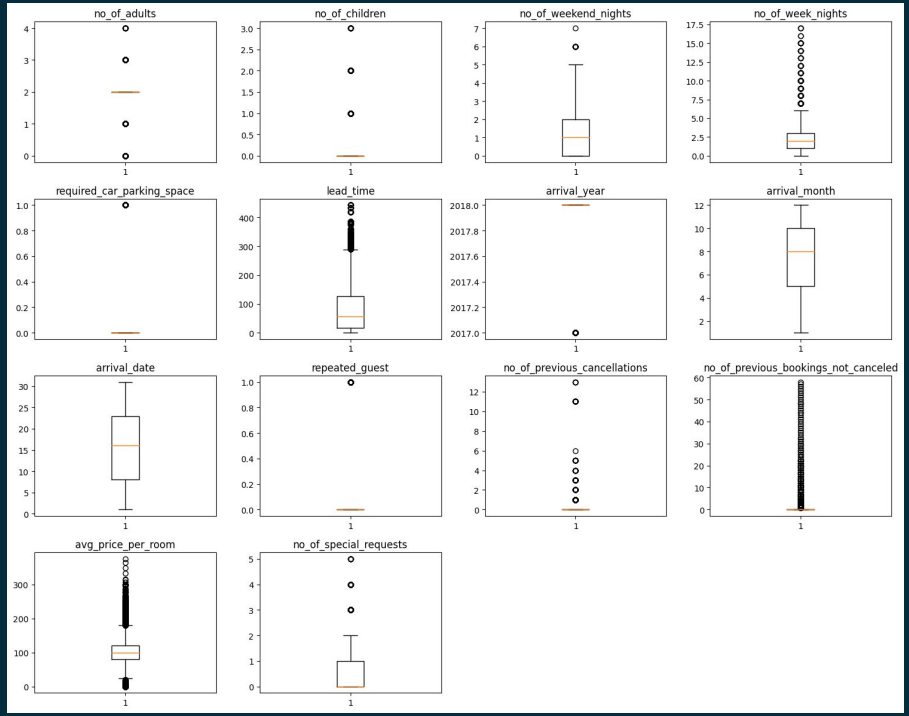
Missing values:

- None in the dataset

Outliers:

- Outliers were detected using 16 feature boxplots

- Were not removed because price and lead_time vary naturally



Feature Engineering

New Variables Created:

- no_of_family_members = no_of_adults + no_of_children
- total_days = no_of_week_nights + no_of_weekend_nights

Encoded:

- Converted these categorical variables into dummy variables (one-hot encoded columns):
  - type_of_meal_plan
  - room_type_reserved
  - Market_segment_type
- Also added constant term for logistic regression

Model Performance Summary

- Built with statsmodels
- Train/test split: 70/30

Initial Performance:

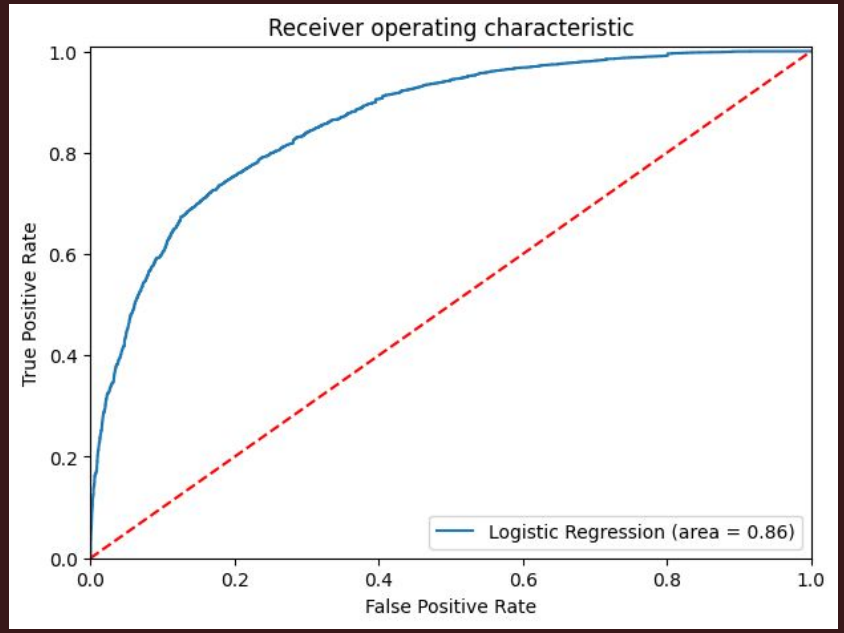
Metric	Value
Accuracy	0.805
Recall	0.632
Precision	0.739
F1	0.682

Important Coefficients

Top Strongest Predictors:

Feature/Variable	Effect on Cancellation	Values Associated
no_of_previous_cancellations (significance seen in EDA)	Increases cancellations	+25.71181% increase in odds per unit
lead_time (significance seen in EDA)	Increases cancellations	+1.58331% increase in odds per unit
no_of_special_requests	Decreases cancellations	-77.00374% decrease in odds per unit
repeated_guest	Decreases cancellations GREATLY	-93.52180% decrease in odds per unit
market_segment_type_Offline	Decreases cancellations	-83.22724% decrease in odds per unit

Threshold Tuning



- To see if we could improve the recall score even further, we changed the model using AUC-ROC curve.
- Thresholds discovered:

0.37 (AUC-ROC) → recall = 0.739 (test)

0.42 (PR-curve) → best balance of precision and recall

Logistic Regression Final Metrics

Training Performance Comparison:

	Logistic Regression-default Threshold	Logistic Regression-0.37 Threshold	Logistic Regression-0.42 Threshold
Accuracy	0.80545	0.79265	0.80132
Recall	0.63267	0.73622	0.69939
Precision	0.73907	0.66808	0.69797
F1	0.68174	0.70049	0.69868

There seems to be clear model improvement with the thresholds than when compared to the default, especially in the recall section.

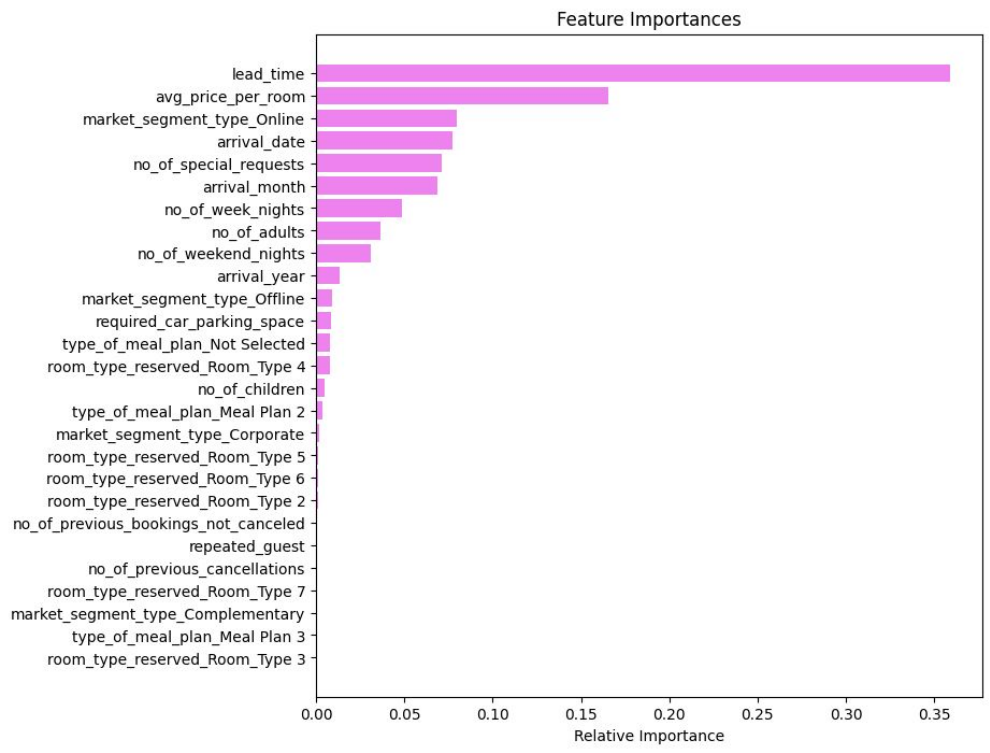
- ★ Accuracy is not the correct metric for this problem. The dataset is not balanced
  - Correctly predicting “not canceled” dominates the accuracy score
  - Improving “cancelled” predictions impacts accuracy less

Decision Tree Model

- To fix the imbalance in the dataset, the decision tree model was built with class_weight = “balanced”

- Pre-pruning and Post-pruning methods were applied
  - We trained 3 different versions
    - Default overfitted tree
    - Pre-pruned tree
    - Post-pruned tree (using cost-complexity pruning)
  - Post-pruning had the best-performing model → recall = 0.85

Feature Importance (Tree)



Top Features/Variables:

- Avg_price_of_room
- Lead_time
- Arrival_month
- No_of_special_requests

- The tree repeatedly split on these top variables because they produced the largest differences in cancellation outcomes

Decision Tree Performance Comparison

Training:

- ★ Sklearn Tree: Accuracy 0.994, Recall 0.987
- ★ Pre-pruning: Accuracy 0.830, Recall: 0.795
- ★ Post-pruning: Accuracy 0.888, Recall: 0.884

Testing:

- ★ Sklearn Tree: Accuracy 0.873, Recall 0.803
- ★ Pre-pruning: Accuracy 0.830, Recall: 0.778
- ★ Post-pruning: Accuracy 0.872, Recall: 0.851

- We can see overfitting in default trees and the improvement in post-pruning

Final Model Selection

- Post-pruned Decision Tree gives best recall of 0.85 along with balanced metrics

- Logistic Regression with threshold 0.37
  - ➢ Improves recall, but not as high

- INN Hotel wants to minimize false negatives which is maximizing recall
  - ➢ This would align best with the post-pruned decision tree, because it has the higher recall.

Business Recommendations

Recommendations:

- ★ Target highest-risk segments → Online market segment
- ★ Monitor high lead-time bookings
  - Could offer incentives to reduce likelihood of cancellation
- ★ Highly encourage special requests
- ★ Dynamic pricing strategy
  - High price bookings correlate with cancellations
- ★ Loyalty programs need to be strengthened
  - Repeated guests almost never cancel

Appendix: Logistic Regression Details

VIF Checks:

	feature	VIF
0	const	39497686.20788
1	no_of_adults	1.35113
2	no_of_children	2.09358
3	no_of_weekend_nights	1.06948
4	no_of_week_nights	1.09571
5	required_car_parking_space	1.03997
6	lead_time	1.39517
7	arrival_year	1.43190
8	arrival_month	1.27633
9	arrival_date	1.00679
10	repeated_guest	1.78358
11	no_of_previous_cancellations	1.39569
12	no_of_previous_bookings_not_canceled	1.65200
13	avg_price_per_room	2.06860
14	no_of_special_requests	1.24798
15	type_of_meal_plan_Meal Plan 2	1.27328
16	type_of_meal_plan_Meal Plan 3	1.02526
17	type_of_meal_plan_Not Selected	1.27306
18	room_type_reserved_Room_Type 2	1.10595
19	room_type_reserved_Room_Type 3	1.00330
20	room_type_reserved_Room_Type 4	1.36361
21	room_type_reserved_Room_Type 5	1.02800
22	room_type_reserved_Room_Type 6	2.05614
23	room_type_reserved_Room_Type 7	1.11816
24	market_segment_type_Complementary	4.50276
25	market_segment_type_Corporate	16.92829
26	market_segment_type_Offline	64.11564
27	market_segment_type_Online	71.18026

Those squared off in red are the only problematic VIF values, but this is due to the dummy variable

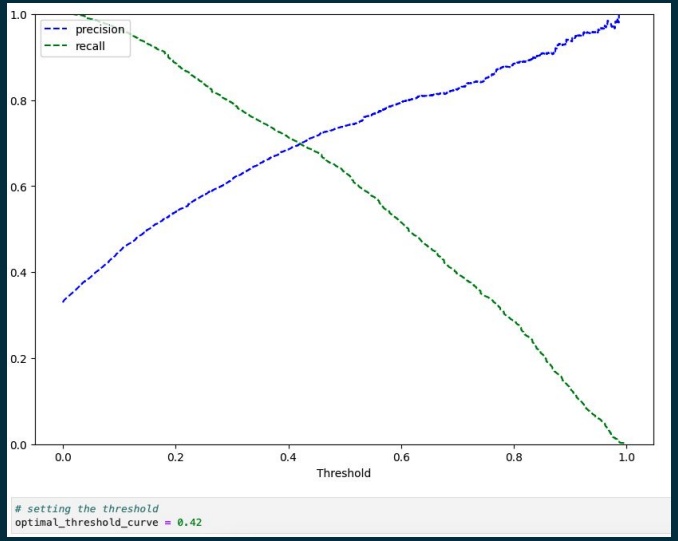
Appendix: Threshold Discovery

```
3  fpr, tpr, thresholds = roc_curve(y_train, lg1.predict(X_train1))
4
5  optimal_idx = np.argmax(tpr - fpr)
6  optimal_threshold_auc_roc = thresholds[optimal_idx]
7  print(optimal_threshold_auc_roc)
```

✓ 0.0s

```
0.3700522558708125
```

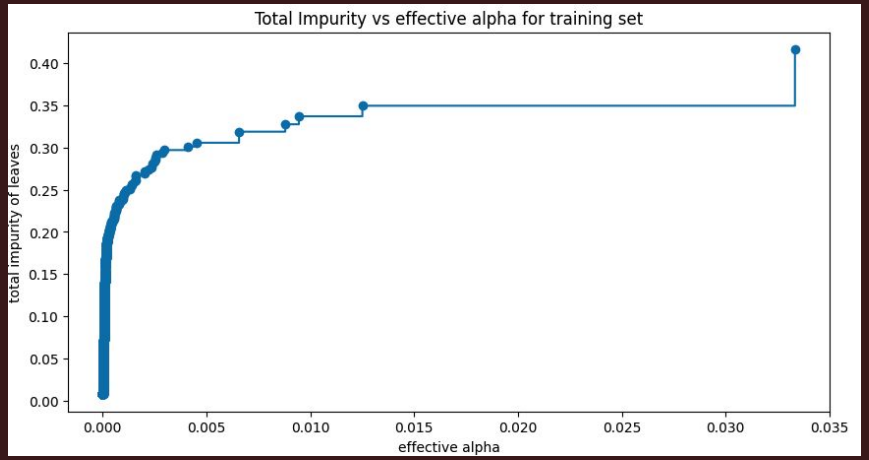
This is how the ROC AUC threshold was found




```
# setting the threshold
optimal_threshold_curve = 0.42
```

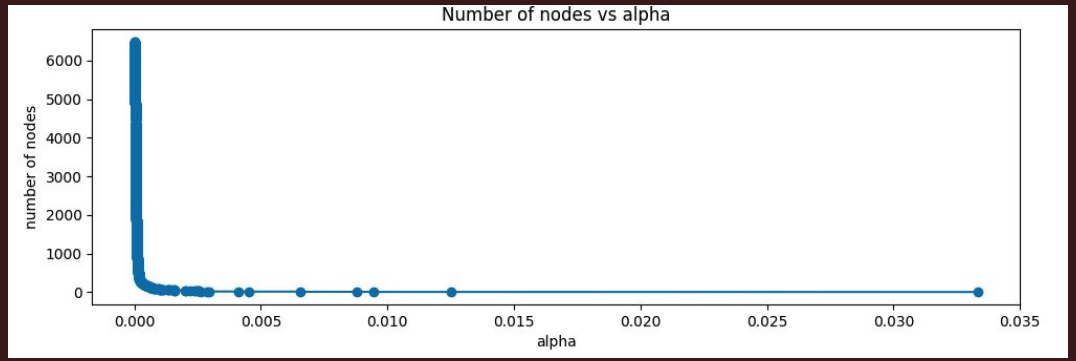
Checked to see if Precision-Recall curve would give a better threshold. Found threshold from the graph at 0.42

Appendix: Decision Tree Pruning



Each point represents a pruned version of the decision tree.

As alpha increases → the tree is pruned → impurity increases a bit because the tree becomes simpler and stops overfitting



Each point equals the size of tree after applying that alpha.

The original, unpruned tree is far too deep and complex.

Thank you