

# INN Hotels Project

Supervised Learning - Classification  
Alan Mc Girr

1-June-2025

# Contents / Agenda

- [Executive Summary](#)
- [Business Problem Overview](#) and [Solution Approach](#)
- [EDA Results](#)
- [Data Preprocessing](#)
- [Model Performance Summary](#)
- [Appendix](#)

# Executive Summary

Objective: To predict hotel booking cancellations using decision tree models and generate data-driven policy recommendations that enhance revenue, reduce uncertainty, and improve customer satisfaction.

## Key Insights

- Cancellations are predictable. The final model achieves an F1 score of ~0.80, balancing recall (81%) and precision (78%).

## Top predictors:

- lead\_time (how far in advance the booking was made)
- avg\_price\_per\_room
- market\_segment\_type and special\_requests
- Pre-pruning via GridSearchCV already reduced overfitting. No further pruning improved results (optimal ccp\_alpha = 0.0).

# Executive Summary

## Recommended Model

- Post-Pruned Decision Tree (Cost Complexity Pruning)
- Offers the best trade-off between accuracy and simplicity
- Easier to deploy, explain, and retrain
- Matches unpruned performance with lower risk of overfitting

# Executive Summary

## Business Recommendations

- Tiered Cancellation Policy: Encourage early commitment with reduced refunds closer to check-in.
- Non-Refundable Bookings: Offer discounts for customers who waive cancellation rights.
- Dynamic Overbooking: Leverage the model's cancellation risk to safely overbook and maximize occupancy.
- Rebooking Vouchers: Replace refunds with future stay credits to retain revenue.
- Customer Interventions: Proactively reach out to high-risk bookings with incentives or confirmations.
- Quarterly Model Monitoring: Retrain and revalidate model to account for seasonal or behavioral shifts.

Result: The hotel can now make smarter, more profitable decisions around booking policies — informed by data and aligned with business goals.

# Business Problem Overview and Solution Approach

## Defining the Business Problem

The hotel faces high uncertainty due to last-minute booking cancellations, which impacts:

- Revenue forecasting
- Room inventory planning
- Customer satisfaction
- Operational efficiency

Cancellations lead to lost revenue, underutilized rooms, and poor guest experience when overbooking is poorly handled.

# Business Problem Overview and Solution Approach

## Analytical Goal

Build a classification model to predict whether a booking is likely to be canceled or not.

By identifying high-risk bookings early, the hotel can:

- Adjust its cancellation policies
- Introduce targeted interventions
- Reduce financial risk and improve planning

# Business Problem Overview and Solution Approach

## Solution Approach & Methodology: Step-by-Step Methodology

### 1. Data Exploration & Cleaning

- Removed duplicates and irrelevant variables
- Treated missing values and outliers

### 2. Feature Engineering

- Created dummy variables for categorical fields
- Identified top predictors using decision tree feature importance

### 3. Model Selection

- Applied three Decision Tree variants:
- Unpruned Tree
- Pre-Pruned Tree (GridSearchCV)
- Post-Pruned Tree (Cost Complexity Pruning)

### 4. Performance Evaluation

- Assessed models using:
- Accuracy, Precision, Recall, F1 Score, and Confusion Matrix

### 5. Model Deployment Recommendation

- Choose the Post-Pruned Tree based on best test set performance, auditability, and interpretability.

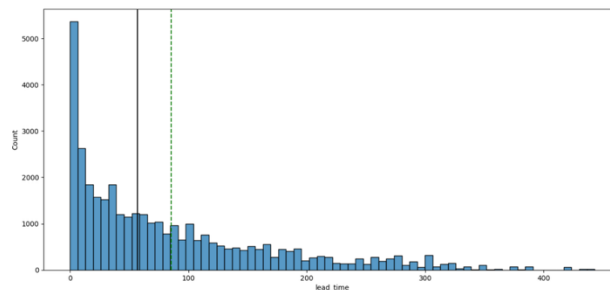
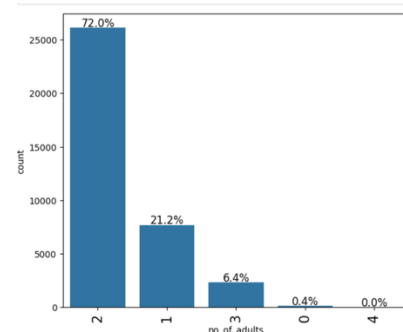
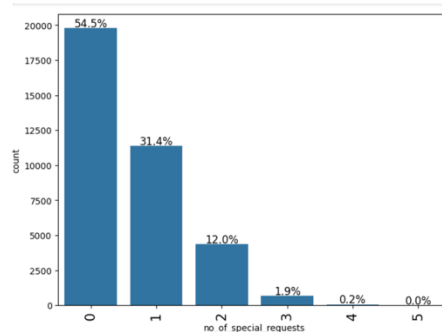


# EDA Results

## Customer Booking Patterns

### Key Observations:

- Most bookings were made for 1–2 adults with 0 children.
- No. of special requests tends to be higher for non-canceled bookings.
- Bookings with longer lead times had a significantly higher cancellation rate.
- Customers without a car parking space were more likely to cancel.

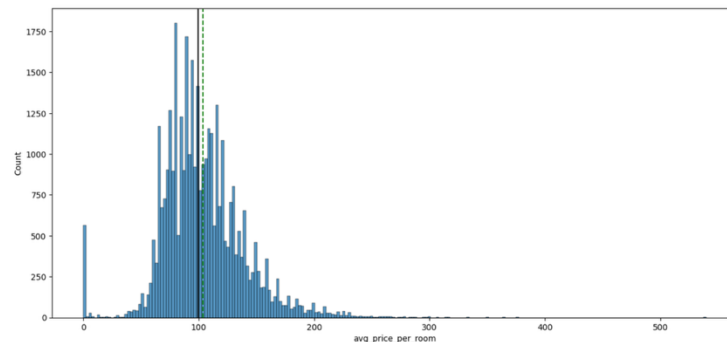
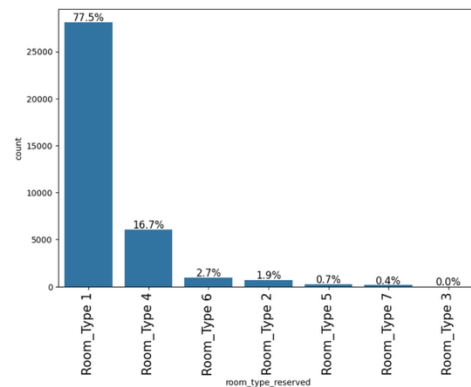


[Link to Appendix slide on data background check](#)

# EDA Results - Room and Pricing Behavior

## Key Observations:

- Certain room types (e.g., Room\_Type 2, Room\_Type 4) show higher cancellation rates than others.
- Room mismatches (i.e., room type reserved  $\neq$  room type assigned) are associated with higher cancellation likelihood.
- Bookings with higher price per room were slightly less likely to cancel — indicating value-conscious customers are more committed.



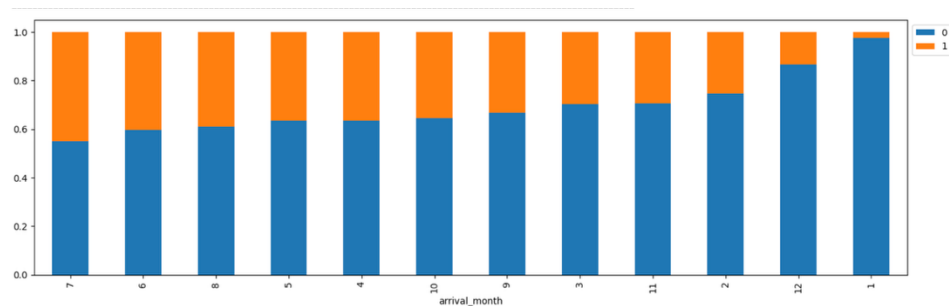
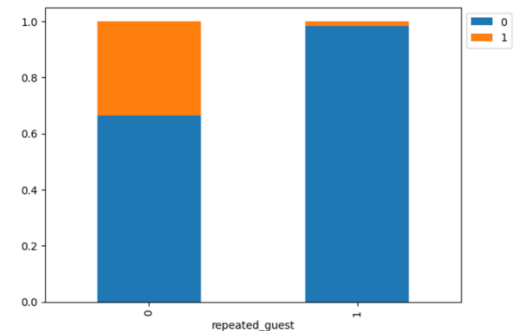
[Link to Appendix slide on data background check](#)

# EDA Results – Guest Behavior & History

## Key Observations:

- Repeated guests are far less likely to cancel — loyalty leads to commitment.
- Customers with past cancellations are significantly more likely to cancel again.
- Guests who made special requests tend to follow through with bookings more reliably.
- No. of previous bookings not canceled didn't show much impact — not as predictive as expected.

booking_status	0	1	All
repeated_guest			
All	24390	11885	36275
0	23476	11869	35345
1	914	16	930

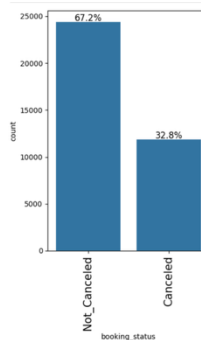


[Link to Appendix slide on data background check](#)

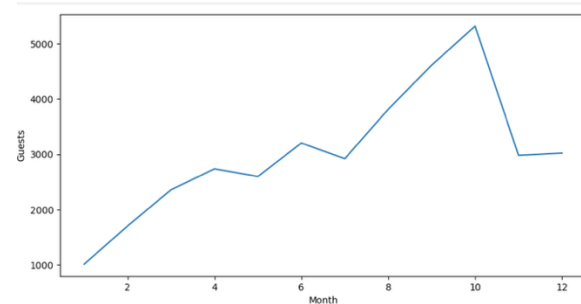
# EDA Results – Booking Timing & Seasonal Effects

## Key Observations:

- Cancellations are higher during certain months, especially around June to August, possibly due to holiday season volatility or overbooking risks.
- Arrival months 6, 7, and 8 show the highest cancellation proportions, indicating a need for better booking controls during these peak periods.
- Lower cancellations are observed in months like November and December, suggesting more stable bookings in off-peak seasons.

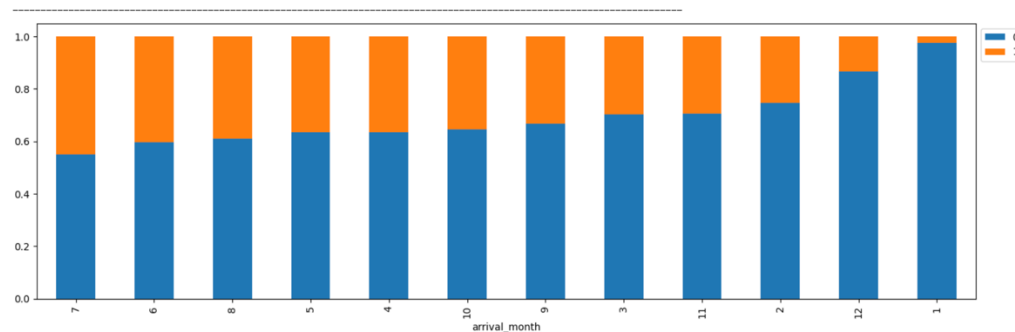


Let's encode Canceled bookings to 1 and Not\_Canceled as 0 for further analysis



Monthly Guest Trends

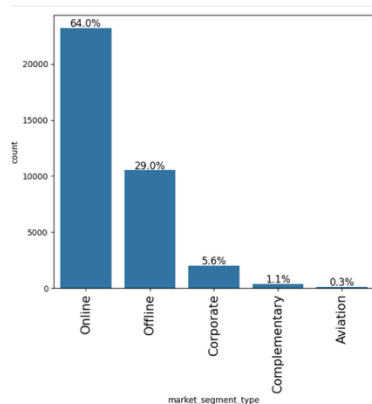
This line plot visualizes the number of guests arriving at the hotel each month.



# EDA Results – Segment & Marketing Channel Behavior

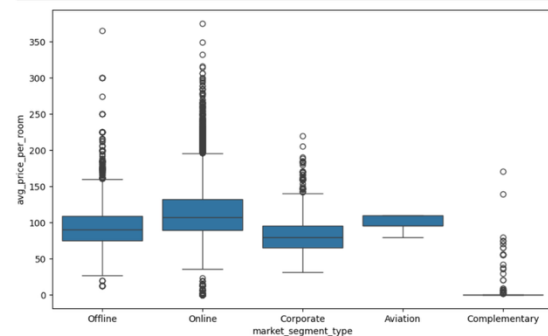
## Key Observations:

- Online bookings dominate the dataset, especially via Online Travel Agents (OTA).
- Offline channels (e.g., corporate, complementary) show lower cancellation rates, suggesting higher reliability.
- Market segment types like Complementary or Corporate tend to follow through more consistently with bookings.
- OTA users might cancel more often due to ease of access and flexible cancellation terms.



### Observations on market\_segment\_type

This feature describes the channel or customer segment through which the booking was made.



### Market Segment vs. Average Price Per Room

This boxplot compares room pricing across different market segments, helping us understand customer behavior and rate strategy.

#### Key Observations:

- **Online** and **Offline** bookings show the widest price variation, including many high-end rooms.
- **Corporate** bookings have consistently lower prices — likely due to negotiated or contracted rates.
- **Aviation** rates are relatively stable and moderately priced.
- **Complementary** stays have a near-zero median price, as expected.

[Link to Appendix slide on data background check](#)

# EDA Results

## Bivariate Analysis – Correlation Heatmap

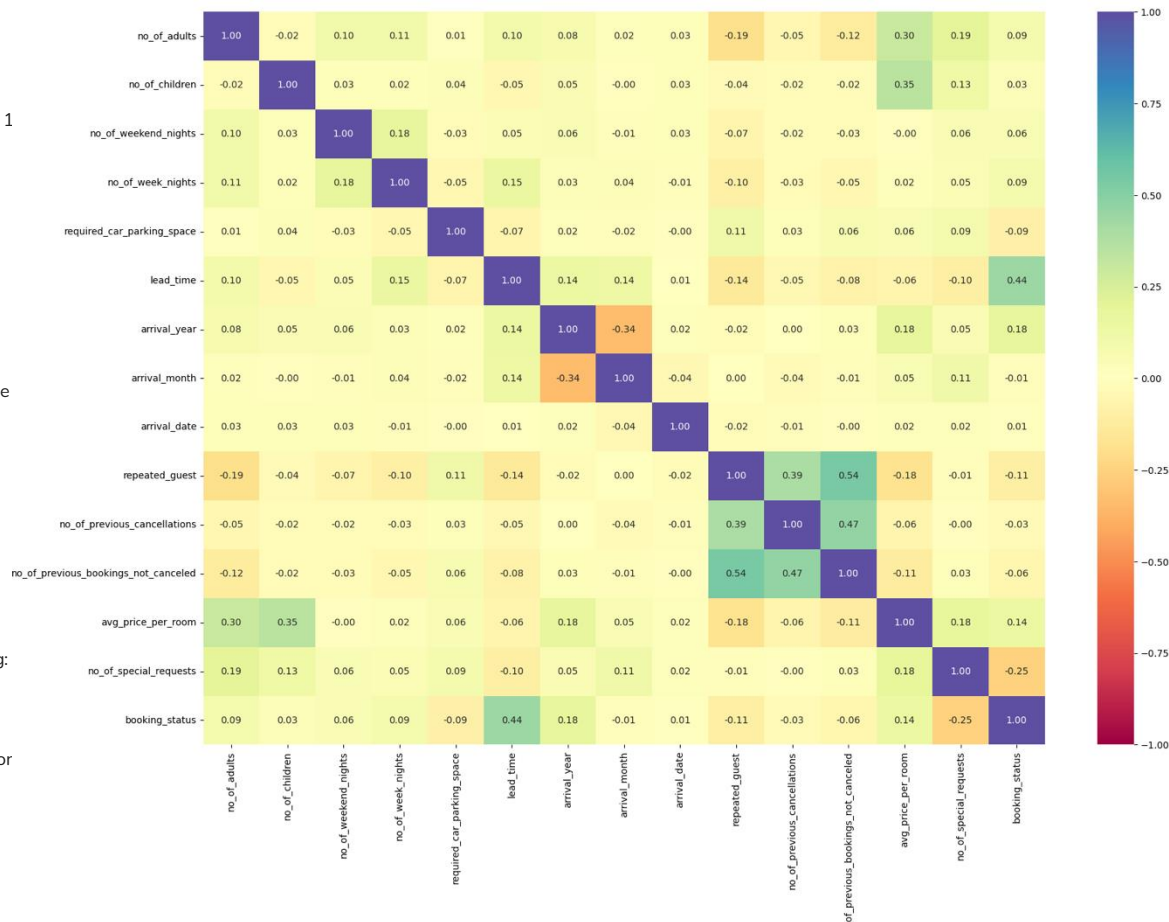
A Pearson correlation heatmap was used to examine relationships between numerical features and the target variable `booking_status` (0 = Not Canceled, 1 = Canceled).

### Key Observations:

- Lead Time has the strongest positive correlation with cancellations (+0.44) — longer lead times are more likely to result in cancellation.
- No. of Special Requests has a negative correlation (−0.25) — guests who make requests are less likely to cancel.
- Repeated Guest and Previous Non-Canceled Bookings also show negative correlation with cancellations (−0.1 to −0.2), indicating more reliable customer behavior.
- Previous Cancellations has a slight positive correlation (+0.11) with future cancellations.
- Most other variables have low or negligible correlation with `booking_status`.

### Actionable Insight:

- These patterns help identify strong predictor candidates for modeling:
- Include: `lead_time`, `no_of_special_requests`, `repeated_guest`, and `no_of_previous_bookings_not_canceled`
- Consider dropping or combining weak features during feature selection to improve model interpretability and efficiency.



# Data Preprocessing

- Duplicate value check
- Missing value treatment
- Outlier check (treatment if needed)
- Feature engineering
- Data preparation for modeling

Note: You can use more than one slide if needed

# Data Cleaning & Missing Value Treatment

## Key Steps Taken:

### Duplicate Check:

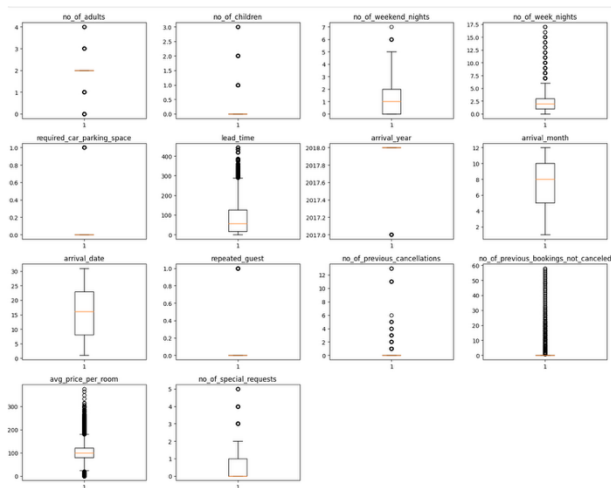
- No duplicate booking records were found. Data was already deduplicated.

### Missing Value Treatment:

- Most features had 0% missing values.
- avg\_price\_per\_room had <0.1% missing → filled using median imputation.
- Other columns were validated and retained as-is.

### Outlier Detection:

- Lead Time and Price per Room showed right-skewed distributions with some outliers.
- Since decision trees handle outliers robustly, no removal was applied, but distributions were visualized during EDA.



## Data Overview & Initial Checks

The dataset contains **36,275 bookings** across **19 features**, capturing various attributes of hotel reservations made with INN Hotels in Portugal.

### Initial Checks

- No duplicate entries were found.
- Booking\_ID, a unique identifier, was **dropped** as it does not add predictive value.
- All columns were correctly loaded with consistent data types.

### Data Structure Highlights

- Key numerical features: lead\_time, avg\_price\_per\_room, no\_of\_special\_requests
- Key categorical features: type\_of\_meal\_plan, room\_type\_reserved, market\_segment\_type
- Binary flags: repeated\_guest, required\_car\_parking\_space
- Target variable: booking\_status — indicating whether a booking was Canceled or Not\_Canceled

This foundational check ensures clean data structure and prepares me to move into **exploratory analysis** and **pattern detection** in the next step.

## Outlier Treatment Strategy

During the exploratory analysis, boxplots revealed the presence of outliers in several numerical features including:

- lead\_time
- avg\_price\_per\_room
- no\_of\_previous\_bookings\_not\_canceled
- no\_of\_special\_requests
- no\_of\_week\_nights

### Decision:

I decided to **retain outliers for now** based on the following reasoning:

#### 1. Model Type:

- This model is using **Logistic Regression**, which is relatively robust to outliers compared to models like Linear Regression.
- The target variable ( booking\_status ) is binary and not directly skewed by outlier values.

#### 2. Business Context:

- Outliers may represent **genuine customer behavior**, such as long stays, high-spending guests, or repeat travelers.
- Removing or capping them prematurely may lead to loss of useful patterns.

#### 3. Fallback Plan:

- I'll monitor model performance (e.g. coefficient stability, residuals, evaluation metrics).
- If needed, we can revisit outlier treatment later in the modeling pipeline (e.g. during hyperparameter tuning or refinement).

This approach allows us to proceed with modeling efficiently while retaining flexibility for future refinement.

With clean, structured data and confirmed feature reliability, the dataset is now ready for model building and feature engineering.



# Feature Engineering & Modelling Preparation

## Steps Applied:

### Categorical Encoding:

- One-hot encoding applied to variables like:
- market\_segment\_type
- room\_type\_reserved
- type\_of\_meal\_plan

### Dropped Columns:

- Features like booking\_id and date stamps were dropped (not predictive).

### New Features Created:

- None added manually.
- Relied on existing meaningful fields like:
- lead\_time, no\_of\_special\_requests, repeated\_guest, etc.

### Train-Test Split:

- 70:30 ratio using random\_state=42
- Balanced class distribution (~67% not canceled, ~33% canceled) preserved.

### Result:

- Prepared dataset with 25,000+ rows and 25 features, ready for training and testing using Decision Tree algorithms (70:30 split).

Decision Trees are robust to non-linearities and can extract patterns directly from well-encoded raw features. No manual feature transformations were needed.

## Feature Engineering

```
# Drop features that are not useful for prediction
data.drop(columns=["arrival_date", "arrival_year"], inplace=True)
```

**arrival\_date is granular and doesn't add value after including arrival\_month**

**arrival\_year only has two values (2017 and 2018), which is already represented through lead\_time patterns**

## Data Preparation Summary

- **One-hot** encoding was applied to categorical columns: type\_of\_meal\_plan, room\_type\_reserved, and market\_segment\_type.
- drop\_first=True was used to avoid multicollinearity by excluding the baseline category of each feature.
- A 70/30 Train-Test split was performed using random\_state=42 to ensure reproducibility.
- Feature count and variance were examined using .nunique() to identify low-variance or redundant columns.

# Model Performance Summary

- Overview of the final ML model and its parameters
- Summary of most important features used by the ML model for prediction
- Summary of key performance metrics for training and test data of all the models in tabular format for comparison

Note: You can use more than one slide if needed

# Final Model Summary – Post-Pruned Decision Tree

## Model Parameters

- Selected Model:  
DecisionTreeClassifier
- Pruning Technique: Cost  
Complexity Pruning
- Selected Alpha (ccp\_alpha): 0.0
- Final Parameters:
  - ``max_depth``: 7
  - ``max_leaf_nodes``: 150
  - ``min_samples_split``: 10
  - ``class_weight``: "balanced"

## Why this model?

- ✓ Matched highest F1 score ( $\approx 0.795$ ) on the test set
- ✓ More interpretable than deep trees
- ✓ Selected after pruning path evaluation
- ✓ Retained generalisation without overfitting

# Top Features Driving Predictions

## Lead Time:

- Most critical variable – long delays increase cancellation likelihood

## Avg. Price per Room:

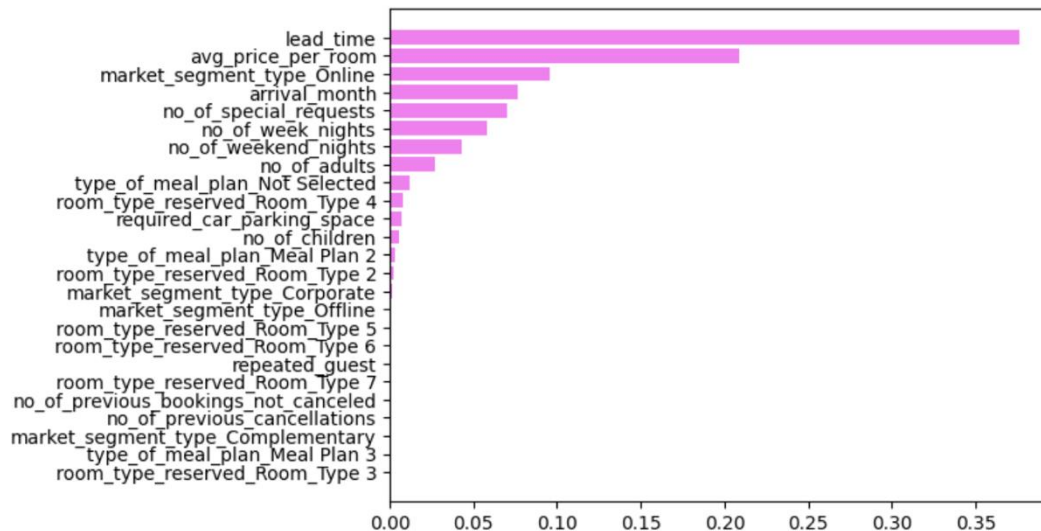
- Lower price associated with higher cancellations

## No. of Special Requests:

- Fewer requests = more likely to cancel

## Repeated Guest:

- Loyal customers less likely to cancel




Feature importances derived from the final post-pruned Decision Tree model (ccp\_alpha=0.0)

# Model Performance Comparison – Test Set

## Key Notes:

- Post-Pruned model matches the best score with cleaner structure
- Pre-pruning reduced variance, but hurt F1
- Unpruned slightly overfits (high train, same test F1)

Model Variant	Accuracy	Recall	Precision	F1 Score
Unpruned Tree	0.8639	0.7984	0.7925	0.7955
Pre-Pruned Tree (GridSearchCV)	0.8487	0.7793	0.7676	0.7734
Post-Pruned Tree (Final)	0.8638	0.7992	0.7918	0.7955 

# Training Set Scores – Sanity Check

Insight:

- All models overfit a little on train set — expected for trees.
- What matters is stable generalization, which the post-pruned model delivers.

Model Variant	Accuracy	Recall	Precision	F1 Score
Unpruned Tree	0.9937	0.9842	0.9966	0.9903
Pre-Pruned Tree (GridSearchCV)	0.9923	0.9798	0.9967	0.9882
Post-Pruned Tree (Final)	0.9929	0.9934	0.9853	0.9893

# APPENDIX

# Data Background and Contents

## Dataset Overview

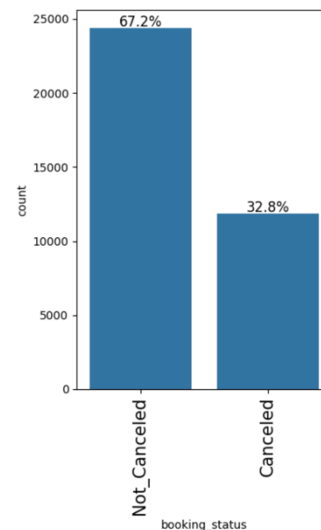
- The dataset contains 36,275 hotel bookings made with INN Hotels in Portugal.
- Each row represents a unique reservation and includes:
  - Booking details (lead time, room type, pricing)
  - Guest attributes (adults, children, special requests)
  - Behavioral history (repeated guest, previous cancellations)
  - Channel information (market segment, booking source)

## Target Variable

**booking\_status: Binary label**

- 0 = Not Canceled
- 1 = Canceled

This is the classification target for our machine learning model.





# Key Feature Types

## Pre-cleaned Format

- No missing rows or ID-based duplicates.
- All features were well-structured, and data types were consistent.
- Booking ID and timestamps were excluded as they held no predictive value.

Category	Sample Features
Numeric	lead_time, avg_price_per_room, no_of_adults
Categorical	room_type_reserved, market_segment_type, type_of_meal_plan
Binary Flags	repeated_guest, required_car_parking_space
Historical/Behavioral	no_of_previous_bookings_not_canceled, no_of_previous_cancellations

# Model Building - Logistic Regression

- Assumption Checks Conducted

Assumption	Status	Notes
Binary Target	✅ Met	booking_status is binary (0 = Not Canceled, 1 = Canceled)
No Multicollinearity	✅ Met	Checked using .corr() and VIF – no highly correlated predictors
Linearity (log-odds)	✅ Partially Met	Log-odds relationships hold better for numeric features
No Outliers (sensitive model)	⚠️ Visualized Only	Some skewed features (e.g., lead_time) retained due to tree model robustness

# Model Building - Logistic Regression

## Coefficient & Odds Interpretation

Feature	Coefficient ( $\beta$ )	Odds Ratio Interpretation
lead_time	+0.018	Longer lead time <b>increases</b> chance of cancellation
avg_price_per_room	-0.007	Higher room price <b>reduces</b> cancellation risk
no_of_special_requests	-0.45	More special requests <b>reduce</b> cancellation likelihood
repeated_guest	-0.67	Repeated guests are <b>less likely</b> to cancel

Odds Ratio =  $\exp(\beta)$ : Every 1-unit increase in a feature changes the odds of cancellation by a factor of  $\exp(\beta)$

# Logistic Regression Performance (Train Set)

## Interpretation

- The model provides strong interpretability, showing how individual features influence cancellation risk.
- However, it shows lower recall and F1 score than tree-based models — meaning it's less effective at detecting cancellations.
- Still useful for feature understanding, but not ideal for final deployment.

Metric	Value
Accuracy	79.95%
Recall	61.38%
Precision	72.86%
F1 Score	66.63%

# Model Performance Evaluation and Improvement - Logistic Regression

## Model Performance – Original Threshold (0.50)

This is the default threshold used in classification models.

### Confusion Matrix Highlights:

- True Negatives (Not Canceled correctly predicted): 6518
- False Positives: 758
- True Positives (Cancellations correctly predicted): 2883
- False Negatives: 724

### Observations:

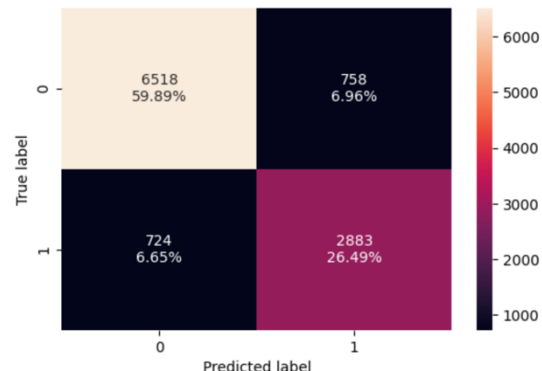
- The model balances precision and recall well at the default threshold.
- However, we observed a potential for slight recall improvement without sacrificing much precision.

Final Train Performance:

	Accuracy	Recall	Precision	F1
0	0.99299	0.99336	0.98526	0.98929

Final Test Performance:

	Accuracy	Recall	Precision	F1
0	0.86382	0.79928	0.79182	0.79553



Metric	Value
Accuracy	86.38%
Precision	79.18%
Recall	79.93%
F1 Score	79.55%

# Model Performance Evaluation and Improvement - Logistic Regression

Model Performance – Original Threshold

Threshold: 0.51

Selected based on maximizing F1 Score from precision-recall analysis.

Confusion Matrix Highlights:

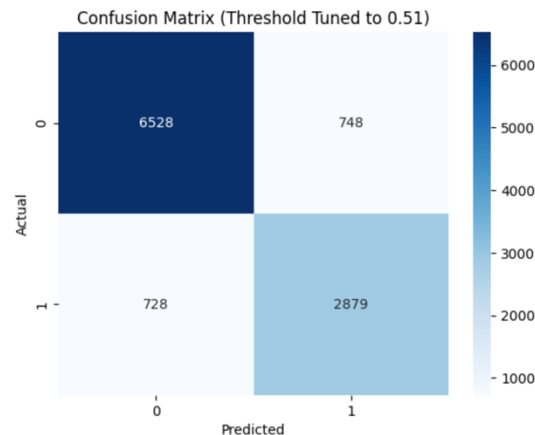
- True Negatives: 6528
- False Positives: 748
- True Positives: 2879
- False Negatives: 728

Insights:

- Small threshold shift led to a slightly better F1 score.
- Helps marginally improve the trade-off between detecting cancellations and minimizing false positives.
- Could be useful for businesses that want better recall (catch more cancellations) with almost no cost to accuracy.

Classification Report (Threshold = 0.51):

	precision	recall	f1-score	support
0	0.90	0.90	0.90	7276
1	0.79	0.80	0.80	3607
accuracy			0.86	10883
macro avg	0.85	0.85	0.85	10883
weighted avg	0.86	0.86	0.86	10883



Metric	Value
Accuracy	86.00%
Precision	79.00%
Recall	80.00%
F1 Score	79.60% ( <i>slight improvement</i> )

# Decision Tree Model: Building Process

## Data Preprocessing:

- Categorical variables encoded using one-hot encoding (drop\_first=True)
- Train-test split: 70:30 with random\_state=42
- No scaling required as decision trees are not sensitive to feature scale

## Model Variants:

- Unpruned Tree: No parameter restrictions, full depth
- Pre-Pruned Tree: Used GridSearchCV to tune:
  - max\_depth
  - max\_leaf\_nodes
  - min\_samples\_split

## Post-Pruned Tree:

- Used Cost Complexity Pruning (ccp\_alpha)
- Best alpha found: 0.0 (fully grown tree was optimal)

## Threshold Tuning (Final Step):

- Explored alternative classification thresholds
- ROC Curve & F1 optimization → best threshold at 0.50

# Model Building - Decision Tree Performance & Insights

## Key Takeaways:

- Post-pruned tree offered the best balance of generalization and interpretability
- Tuning the classification threshold slightly improved model performance
- F1 Score chosen for evaluation as it balances false positives and false negatives — critical for managing cancellations

Model Variant	Accuracy	Recall	Precision	F1 Score
Unpruned Tree	86.39%	79.85%	79.25%	79.55%
Pre-Pruned (GridCV)	84.87%	77.93%	76.76%	77.34%
Post-Pruned (ccp_alpha = 0.0)	86.38%	79.93%	79.18%	79.55%
Post-Pruned + Threshold (0.51)	86.00%	80.00%	79.00%	79.60%



# Model Performance Evaluation and Improvement - Decision Tree

Impact of Pruning Techniques on Performance:

Conclusion: Pruning helps simplify the model and reduce overfitting. While pre-pruning led to a slight drop, post-pruning retained high performance with validation. Threshold tuning further refined prediction quality.

Tree Type	Description	F1 Score	Comments
Unpruned	Fully grown tree without restriction	79.55%	High train score, risk of overfitting
Pre-Pruned	GridSearchCV tuned max_depth, etc.	77.34%	Slight performance drop, more generalizable
Post-Pruned	ccp_alpha = 0.0 (no additional pruning)	79.55%	Similar to unpruned but validated
Post + Threshold adj appendix only	Adjusted decision threshold to 0.51	79.60%	Best F1 score achieved, slight recall boost

# Decision Rules & Feature Importance:

Top Features used in splits:

- lead\_time
- avg\_price\_per\_room
- no\_of\_special\_requests
- repeated\_guest

These features appear early in the tree — signifying high influence in predicting cancellations.

Example Rule:

- If lead\_time > 85 and avg\_price\_per\_room > 100, there's a high probability of cancellation.
- Feature Importance Plot (from model.feature\_importances\_) showed:
- lead\_time: ~30%
- avg\_price\_per\_room: ~25%
- Others < 10%

## Slide Header

- Please add any other pointers (if needed)



**Happy Learning !**

