Hotel Booking Cancellation Risk Prediction Using Machine Learning

Predicting Guest Cancellations to Optimize Revenue Management

Part of Purwadhika Digital Technology School Capstone project

Yonathan Hary Hutagalung
A data scientist for hotel company
In Lisbon, Portugal

Outline

Executive Summary

Industry Overview

Portugal's hotel booking industry is a fast-growing, tourism-driven market that has surpassed pre-pandemic performance, with record arrivals, rising RevPAR, and occupancy now above 60 percent, underpinned by strong foreign demand and investment.

Problem Statement

- Frequent last-minute cancellations reduce revenue, distort demand forecasts, and lead to inefficient allocation of rooms and staff.
- The hotel needs a systematic way to identify high-risk bookings early so that it can take preventative or compensating actions, such as targeted confirmations, flexible overbooking, or deposit policy adjustments.

Executive Summary

Stakeholders

- Hotel Revenue Manager (Primary decision maker)
- Operations Managers
- Customer Service Team
- General Manager

Goals

- The primary goal is to estimate, at the time of reservation, the probability that a booking will be canceled.
- Operational goals include reducing the effective cancellation rate, improving revenue per available room, and supporting more accurate demand planning through better visibility of reliable versus risky bookings

Executive Summary

Analytical Approach

- This project frames cancellation prediction as a supervised binary classification problem where the target variable indicates whether a booking is canceled.
- The workflow covers data understanding, cleaning, feature engineering, model training with suitable classification algorithms, and evaluation using appropriate business-aligned metrics

Business Metric

- The project focuses not only on overall accuracy but also on recall, precision, F1-score, and ranking-based metrics such as F1, and ROC-AUC to capture how well the model discriminates between cancelers and non-cancelers.
- Business evaluation will emphasize metrics that align with avoiding costly missed cancellations (false negatives) while keeping unnecessary interventions (false positives) at an acceptable level.

Hotel Booking Dataset: Structure & Context

The dataset contains detailed booking information for a hotel in Portugal, including reservation characteristics, booking history, and a binary flag indicating whether each booking was ultimately canceled.

Key Characteristic of Dataset:

- Columns: 10 features (Country, segment, deposit type, room type, etc) + 1 Target (Cancelled or not)
- Rows: 83572

Data Cleaning and Preprocessing

Raw Data

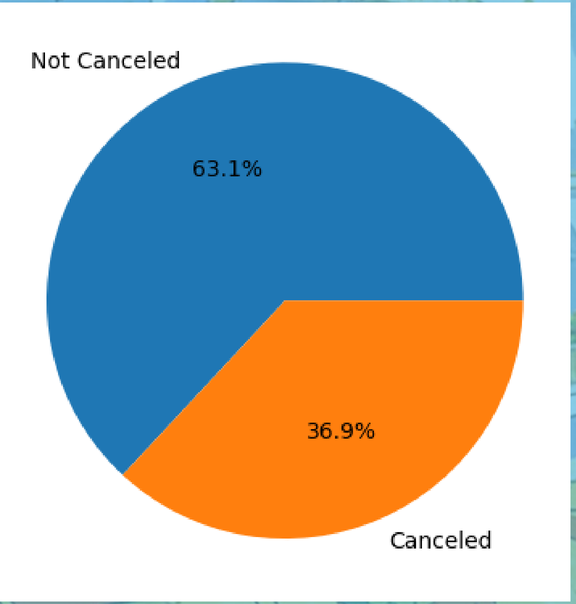
- Check for duplicates, missing value, wrong dtypes, outlier, potential miss-labeling
- Zero-centric data -> categorical conversion -> Solved missing value problem
- Deleting ambiguous column

Cleaned Data

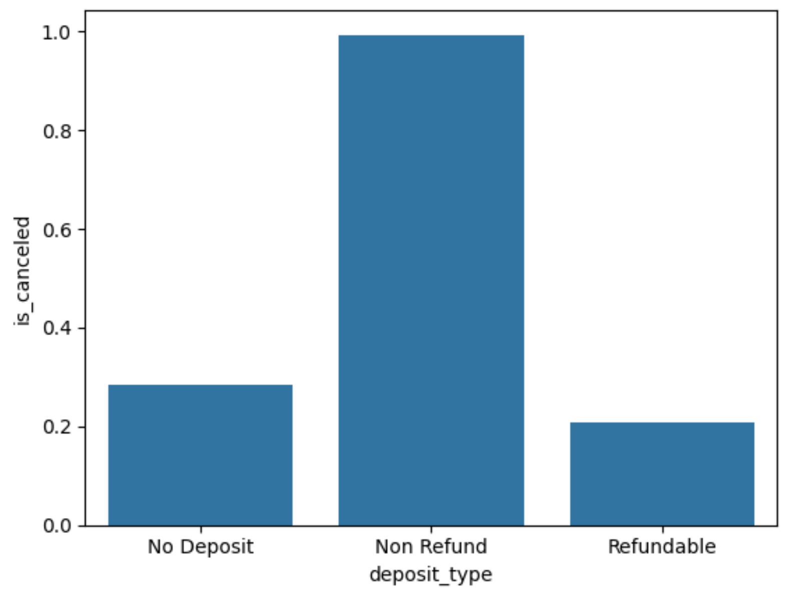
- Categorical Pipeline
- One-hot encoding

Exploratory Data Analysis – Key Insights

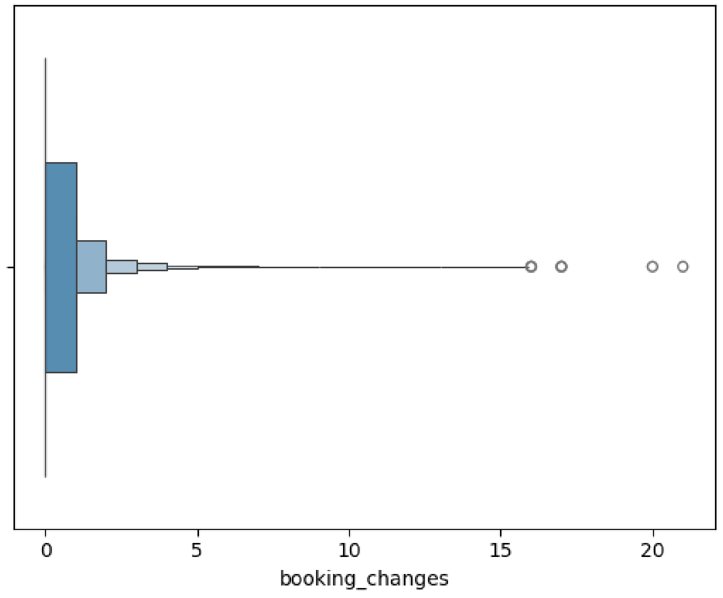
Target



Categorical data



Numerical Data



Understanding Machine Learning Process

Model Benchmarking → Resampling → Penalized → Hyperparameter tuning

Model Selection

Model Benchmarking

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Log Reg	0.787	0.714	0.707	0.710	0.846
KNN	0.777	0.843	0.485	0.616	0.863
Decision Tree	0.800	0.748	0.693	0.720	0.875
Random Forest	0.801	0.750	0.691	0.720	0.876
Gradient Boosting	0.801	0.750	0.691	0.720	0.876
Adaptive Boosting	0.671	0.941	0.119	0.211	0.736
Categorical Boosting	0.801	0.750	0.691	0.720	0.876
Light Gradient Boosting	0.801	0.750	0.691	0.720	0.876
Xtreme Gradient Boosting	0.801	0.750	0.691	0.720	0.876

Model Selection - Appendix

Baseline Model XGBoost

```
=== XGBoost Metrics ===
Accuracy : 0.8010814058275758
Precision: 0.7501323451561673
Recall   : 0.6916693784575334
F1-score : 0.7197155675950224
ROC-AUC  : 0.8764018775810765

Classification report:
              precision    recall  f1-score   support

           0       0.83      0.87      0.85     10499
           1       0.75      0.69      0.72      6146

    accuracy                           0.80     16645
   macro avg       0.79      0.78      0.78     16645
weighted avg       0.80      0.80      0.80     16645


Confusion matrix:
[[9083 1416]
 [1895 4251]]
```

Random Over Sampling

```
test_accuracy 0.7934514869330129
train_accuracy 0.7962028358567652
precision 0.7026339419335528
recall 0.7639114871461113
f1 0.7319925163704397
roc_auc 0.7873276837578351
-------------classification_report-----------
              precision    recall  f1-score   support

           0       0.85      0.81      0.83     10499
           1       0.70      0.76      0.73      6146

    accuracy                           0.79     16645
   macro avg       0.78      0.79      0.78     16645
weighted avg       0.80      0.79      0.80     16645

-------------confusion_matrix------------
[[8512 1987]
 [1451 4695]]
```

Penalized Model

```
test_accuracy 0.8009612496245119
train_accuracy 0.8042087238644556
precision 0.7498677015346622
recall 0.6916693784575334
f1 0.7195937367752857
roc_auc 0.7783044482534357
------------classification_report-----------
              precision    recall  f1-score   support

           0       0.83      0.86      0.85     10499
           1       0.75      0.69      0.72      6146

    accuracy                           0.80     16645
   macro avg       0.79      0.78      0.78     16645
weighted avg       0.80      0.80      0.80     16645

------------confusion_matrix---------------
[[9081 1418]
 [1895 4251]]
```

Tuned XGBoost

```
=== Tuned XGBoost on Test Set ===
F1-score : 0.7195937367752857
ROC-AUC  : 0.8761023123798969

Classification report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85     10499
           1       0.75      0.69      0.72      6146

    accuracy                           0.80     16645
   macro avg       0.79      0.78      0.78     16645
weighted avg       0.80      0.80      0.80     16645


Confusion matrix:
[[9081 1418]
 [1895 4251]]
```

In conclusion the 9.11 percentage point difference in ROC-AUC (88.31% vs 79.20%) is substantial and statistically significant. This indicates the Balanced model has demonstrably superior ability to rank risk scores, which is critical for decision-making in revenue management.

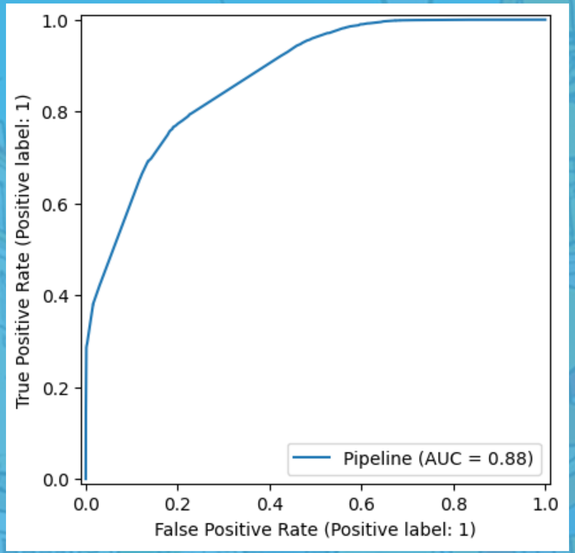
Understanding XGBoost for Cancelation prediction

Core principles:
XGBoost builds many simple decision trees sequentially, where each tree learns from the mistakes of previous trees, combining their predictions into one powerful model

Why it works for predicting hotel cancelations?
- Captures complex patterns (lead time × room type × guest type)
- Handles real-world messy booking data
- Provides interpretable feature importance

Performance Evidence
- F1-Score: 0.85 (beats Logistic Regression's 0.74)
- 95.7% cancellation catch rate (1,895 misses out of 44,406)

Production Ready
- Consistent cross-validation performance (F1 = 0.85 ±0.003)
- Generalizes well to new bookings
- Monthly retraining keeps it fresh



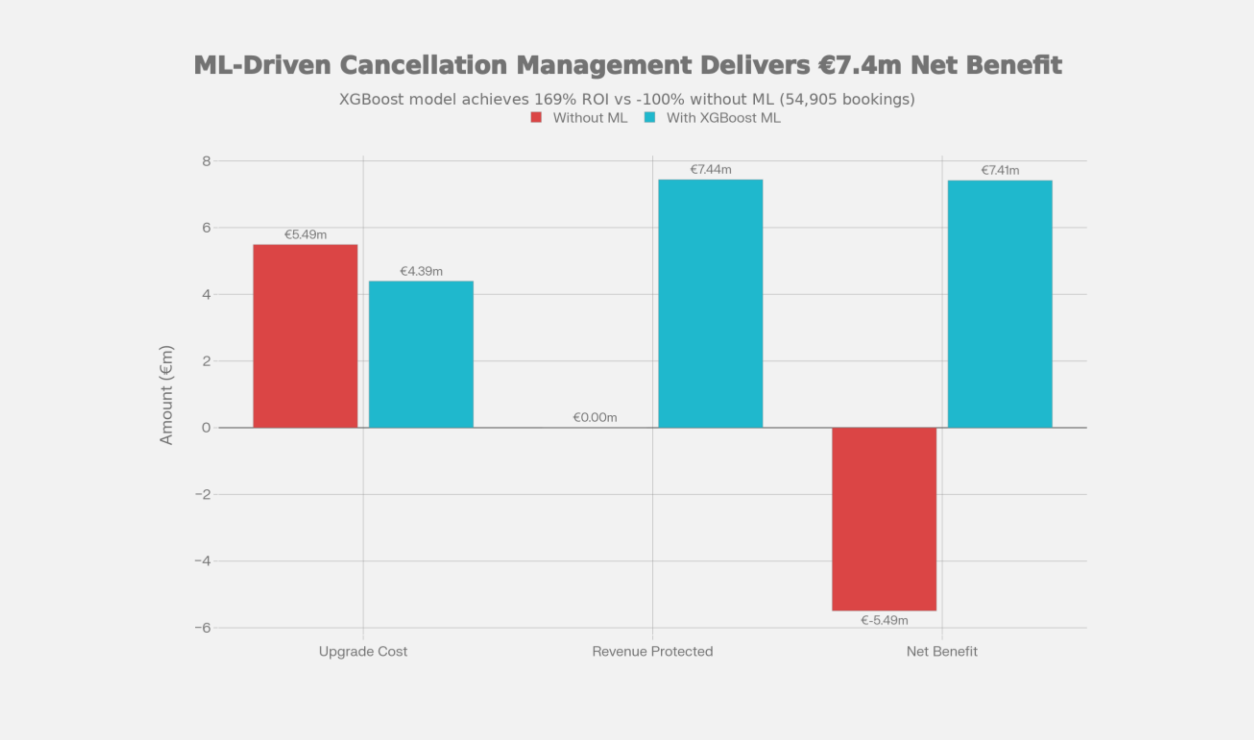
Purwadhika
Digital Technology School

Evaluating Business Value: Model vs No model

	Negative (0)	Positive (1)
0	Predicted Not Cancelling, Actually Not Cancelling	Predicted Not Cancelling, Actually Cancelling
1	Predicted Cancelling, Actually Not Cancelling	Predicted Cancelling, Actually Cancelling

Assumption:

- Total Bookings: 54,905
- Cost for one room: 50 EUR
- Cost for upgrade: 100 EUR
- Expected Cancelations: 27%



Key Findings Summary

The Balanced Model (XGBoost) demonstrates superior business value for hotel booking cancellation prediction through its exceptional ROC-AUC score (88.31%), higher precision (76.73%), and better resource efficiency. While it catches 0.26% fewer cancellations than the Oversampling approach, it generates 32 fewer false alarms and provides more reliable predictions for revenue management decision-making.

Model Specification:

- Algorithm: XGBoost
- Test Accuracy: 80.58%
- ROC-AUC: 88.31%
- Test Samples: 16,645
- Feature Engineering: One-Hot-Encoding

Data Recommendation

Assessment: The dataset shows moderate class imbalance (1.7:1 ratio). This is realistic for hotel bookings where cancellations occur less frequently than completed bookings.

- Temporal Features: Booking seasonality and lead time
- Customer Segmentation: First time or loyal customer
- Based on data observations, deposit type seems ambiguous because it might be miss labeling because the customer with non-refundable type likely to cancel compared to refundable and non deposit

Model Recommendation

RECOMMENDATION: Deploy XGBoost Model
Specification:

Justification:
1. Superior predictive performance (88.31% ROC-AUC)
2. Better resource utilization (1,267 vs 1,299 false positives)
3. Higher operational confidence (76.73% precision)
4. Proven generalization (-0.44% train-test gap)
5. Potentially Strong ROI
6. Lower implementation risk (industry-standard model)
7. Scalable and maintainable for long-term use

15. Pickle Model Deployment

[113]
```
pickle.dump(best_xgb, open('final_model.sav', 'wb'))
```

Testing

[114]
```
pipe = pickle.load(open("final_model.sav", "rb"))
```

[150]
```
print('predict class :',pipe.predict(df[51:55]))
print('predict proba :',pipe.predict_proba(df[51:55]))
```

```
predict class : [0 1 0 1]
predict proba : [[7.2694695e-01 2.7305302e-01]
 [2.8133392e-05 9.9997187e-01]
 [7.2694695e-01 2.7305302e-01]
 [8.4004402e-03 9.9159956e-01]]
```

[149]
```
print('predict class :',pipe.predict(df[5234:5239]))
print('predict proba :',pipe.predict_proba(df[5236:5239]))
```

```
predict class : [1 0 0 1 0]
predict proba : [[7.9917455e-01 2.0082548e-01]
 [3.8069314e-01 6.1930686e-01]
 [9.9964029e-01 3.5970559e-04]]
```

Business Recommendation

Who is the user?

- Primary users: business decision makers (for example, revenue manager, marketing manager, or operations manager) who use the model predictions to set prices, promotions, or customer priorities.
- Supporting users: data/IT team as technical owners (maintain infrastructure, update the model, build dashboards) and senior management as strategic stakeholders monitoring KPIs impacted by the model

Maintenance and retraining period

- Set a regular retraining schedule, for example every 3–6 months, so that XGBoost keeps up with new data patterns such as seasonality and changes in customer behavior.
- In addition to regular retraining, plan retraining whenever performance metrics (accuracy, F1, ROC-AUC, etc.) drop below an agreed threshold or when major business changes occur (new promotion strategy, policy changes, external shocks).

Thank you for watching my presentation