PREDICTING HOTEL BOOKINGS CANCELLATION WITH A MACHINE LEARNING CLASSIFICATION MODEL

DATA MINING AND MACHINE LEARNING PROJECT

Tommaso Falaschi

Master's Degree in Artificial Intelligence
and Data Engineering

INTRODUCTION

Frequent Cancellations force hotels to rely on:

- **Overbooking**, which may lead to service denial, poor guest experience, and reputational damage.
- **Rigid cancellation policies**, which can deter customers and reduce booking volumes.

Consequences:

- Lower occupancy accuracy and revenue predictability
- Loss of customer trust and future bookings
- Inefficient room allocation and pricing decisions

Predicting cancellations in advance allows hotel managers to:

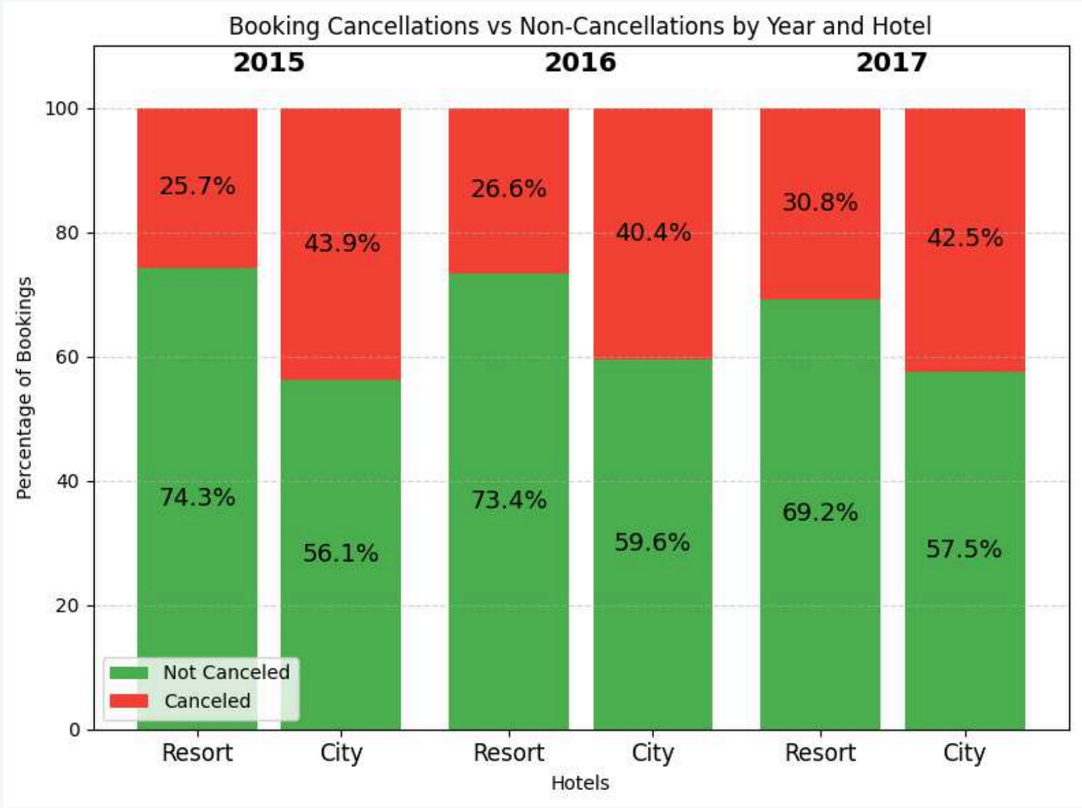
- Proactively mitigate losses with offers or upgrades
- Adjust pricing and overbooking strategies more precisely

DATASET OVERVIEW

- This project uses a real-world dataset of hotel bookings from two distinct hotel types:
- **Resort Hotel** (H1): 40,060 records
- **City Hotel** (H2): 79,330 records
- Covers bookings scheduled to arrive between July 2015 and August 2017
- Includes both confirmed and canceled reservations

Feature Composition (31 Variables):

- **Numerical**: Guests, stay duration, lead time, daily rate (ADR)
- **Categorical**: Customer type, deposit policy, booking channel, room type, country
- **Temporal**: Year, month, week, day of week of arrival
- **Target**: IsCanceled: Binary cancellation flag



DATASET PREPROCESSING

Handling Missing & Undefined Values

- Replaced NaN in Children with 0 (assumed no children)
- Replaced NULL in Agent and Company with 0 (no intermediary)
- Meal values like "Undefined" replaced with SC (Self Catering)
- Dropped rare undefined rows in MarketSegment and DistributionChannel
- Removed bookings with zero guests (Adults + Children + Babies = 0)

Preventing Data Leakage

- Removed Country (e.g., default value “Portugal” often updated only after check-in)
- Removed AssignedRoomType, ReservationStatus, ReservationStatusDate (revealed post check-in)

Feature Engineering

- Created ADRThirdQuartileDeviation.

$$\frac{ADR}{Q3_{ADR}}$$

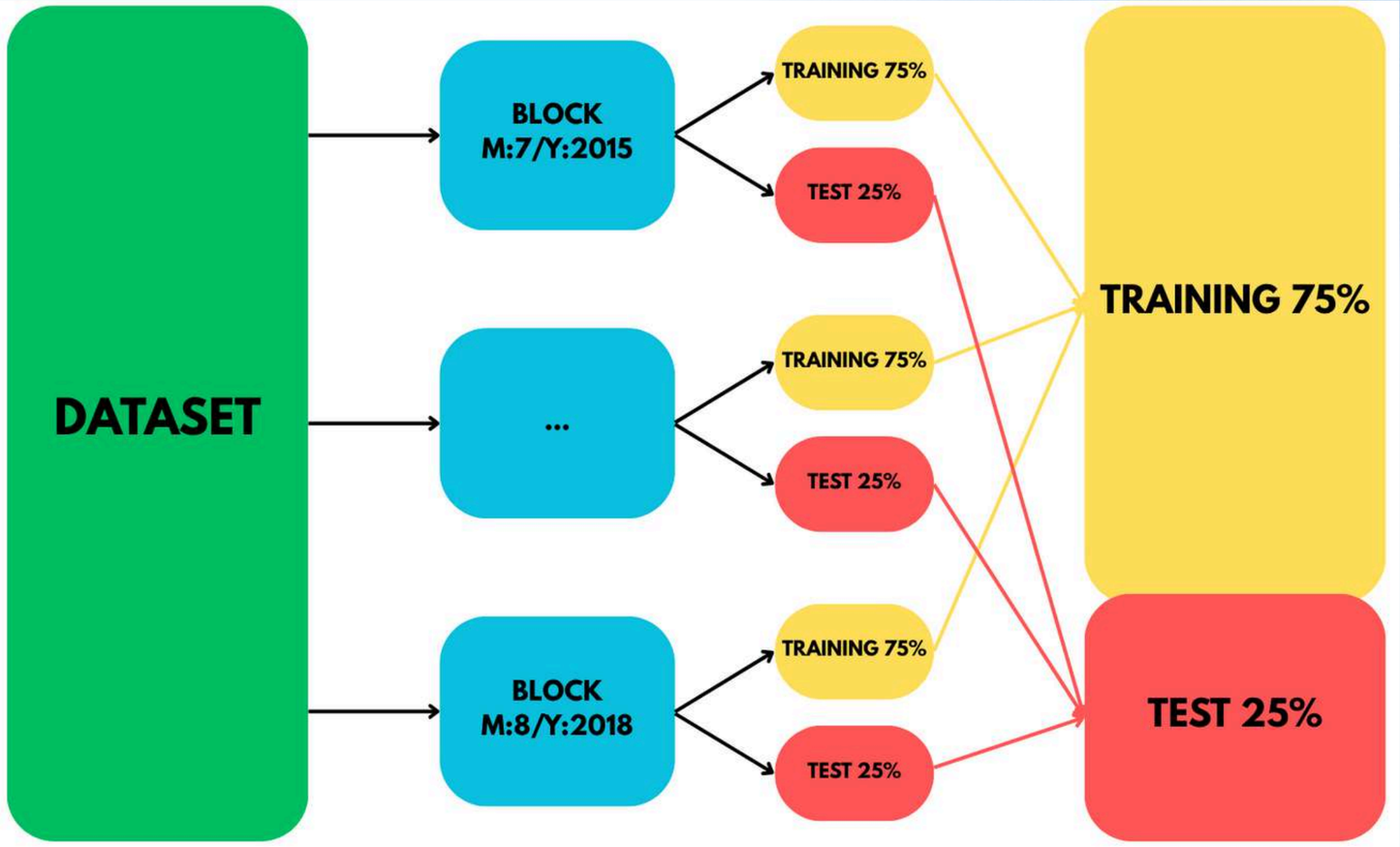
Encoding & Scaling

- Applied Rare Encoding (for high-cardinality categorical features like Agent, Company)
- Applied One-Hot Encoding to all non-binary categorical features
- Used Standard Scaler for all numerical variables to ensure uniform scale

DATASET SPLITTING

Convenience Splitting

- Bookings ordered by arrival date, then grouped into month/year blocks
- Each block is split: 75% training, 25% test (stratified on cancellation)
- Allows the capture of what calls “non-stationary temporal data”



Stratified K-Fold

- Applied only on the training set
- Used 5 folds, preserving class distribution of the target variable.

The test set from convenience split is used only once for final hold-out evaluation. **All temporal features removed before model training to avoid leakage**

MODEL TRAINING: PIPELINES BUILDING

PIPELINES:

1) **Feature engineering**: ADRThirdQuartileDeviation
2) **Sampling**: SMOTENC, None
3) **Preprocessing**: One-Hot Encoding + StandardScaler
4) **Classifier**: Logistic Regression, Random Forest, XGBoost, LightGBM, CatBoost, AdaBoost, Decision Tree, KNN

Although the pipeline with **SMOTENC reduced precision,** it **improved** low **recall** of default pipeline , which is crucial for correctly identifying cancellations. This makes the pipeline with oversampling more suitable for the goals of the project.

Table 3.3: Difference (With SMOTENC - Baseline)

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	-0.025	-0.119	+0.089	+0.002	-0.001
Random Forest	-0.007	-0.043	+0.038	+0.000	-0.001
XGBoost	-0.010	-0.074	+0.066	+0.006	-0.003
LightGBM	-0.010	-0.088	+0.079	+0.009	-0.003
CatBoost	-0.008	-0.074	+0.070	+0.009	-0.002
AdaBoost	-0.023	-0.122	+0.094	+0.006	-0.002
Decision Tree	-0.007	-0.022	+0.022	-0.001	+0.000
KNN	-0.011	-0.051	+0.049	+0.002	+0.000

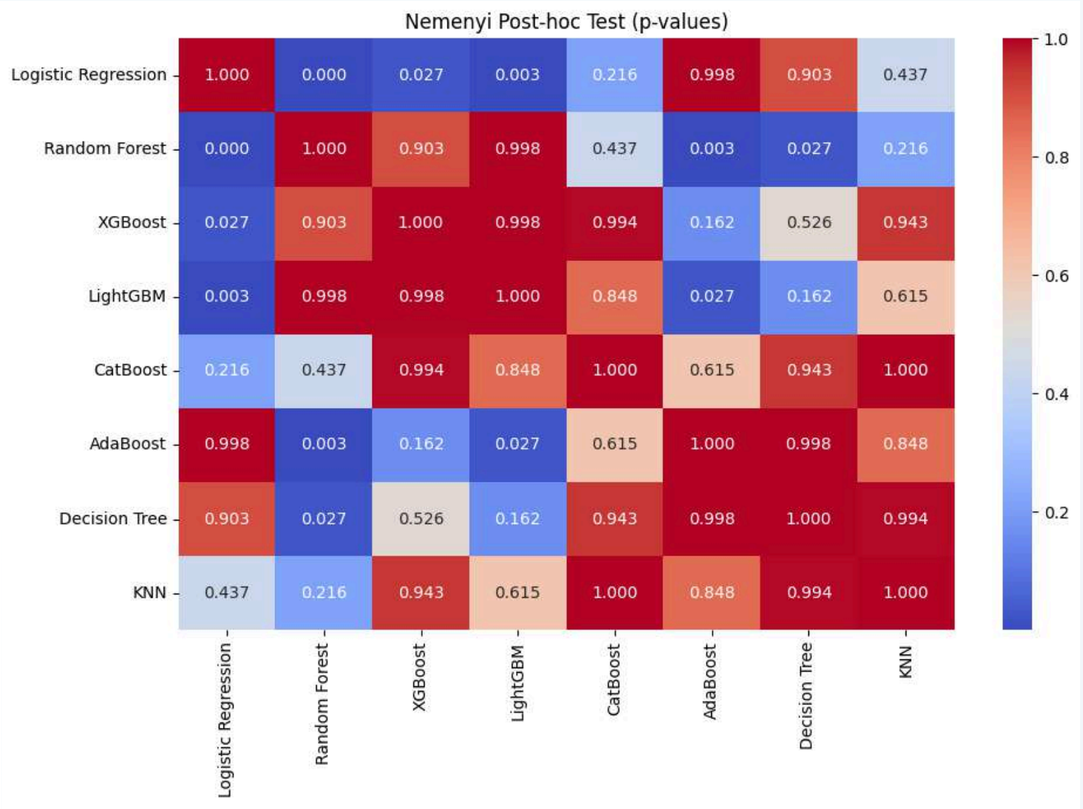
HYPERPARAMETER TUNING AND STISTICAL COMPARISON

We performed extensive hyperparameter tuning using GridSearchCV, optimizing each model based on F1-score. Each optimized model was compared with its default counterpart to verify actual improvements.

Statistical tests (Friedman and Nemenyi) confirmed significant performance differences across models. Based on these results, the top-performing models selected for final hold-out evaluation were: Random Forest, LightGBM, XGBoost, CatBoost, and K-Nearest Neighbors.

Table 3.4: Comparison of Default vs. Optimized Models

Model	F1 (Optimized)	F1 (Default)	Improvement	Chosen Version
KNN	0.764	0.740	+0.024	Optimized
LightGBM	0.775	0.759	+0.016	Optimized
Decision Tree	0.747	0.735	+0.012	Optimized
XGBoost	0.772	0.765	+0.007	Optimized
Random Forest	0.786	0.782	+0.004	Optimized
AdaBoost	0.720	0.717	+0.003	Optimized
Logistic Regression	0.712	0.712	−0.000	Default
CatBoost	0.763	0.768	−0.005	Default

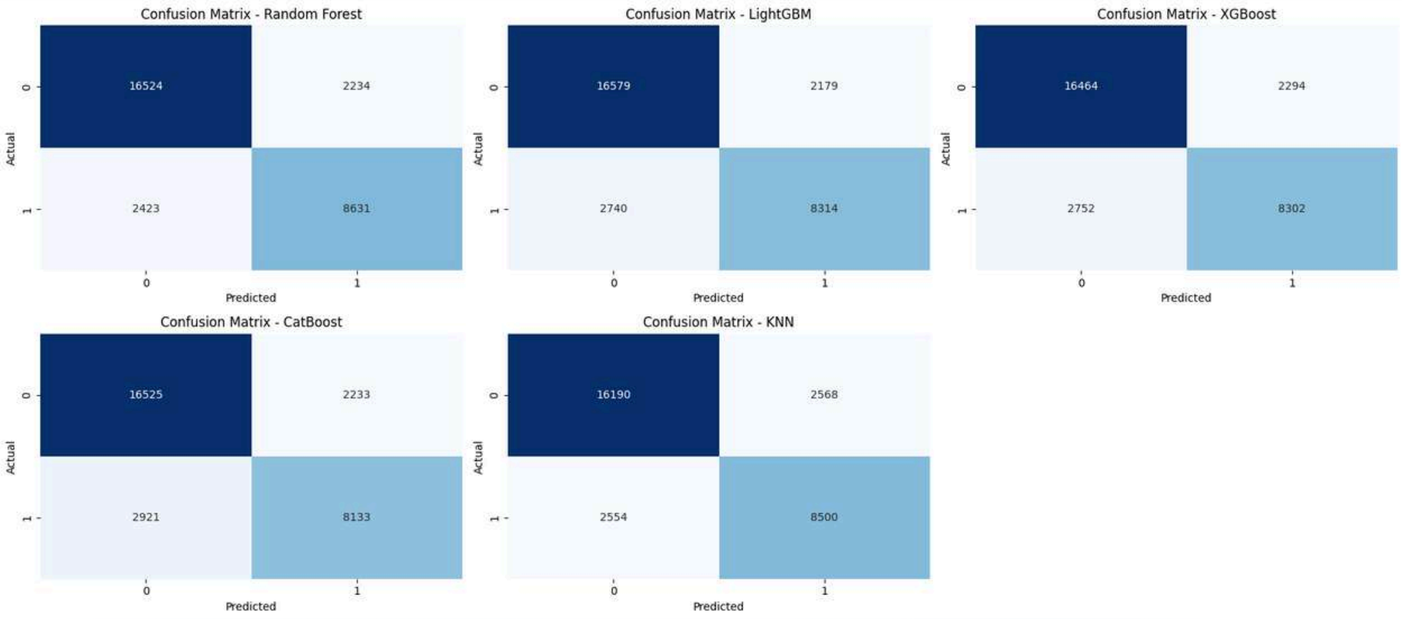


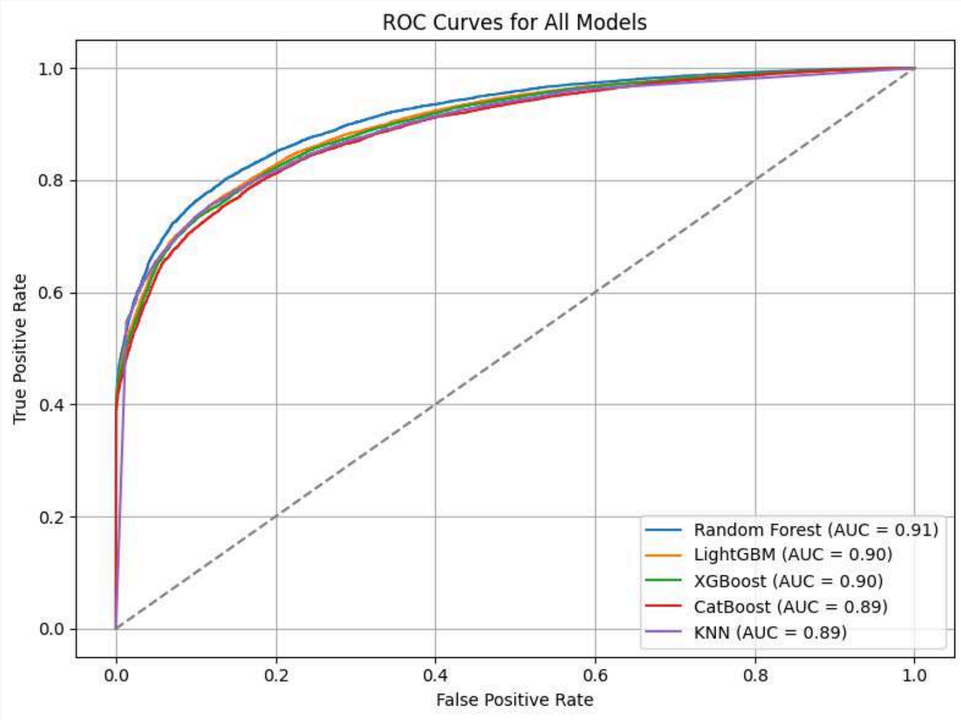
FINAL HOLD-OUT EVALUATION

The top 5 models were evaluated on the **hold-out** test set. **Random Forest** achieved the best overall performance, with all the highest metrics . LightGBM and XGBoost followed closely, while KNN showed strong recall, making it valuable in minimizing false negatives such as missed cancellations.

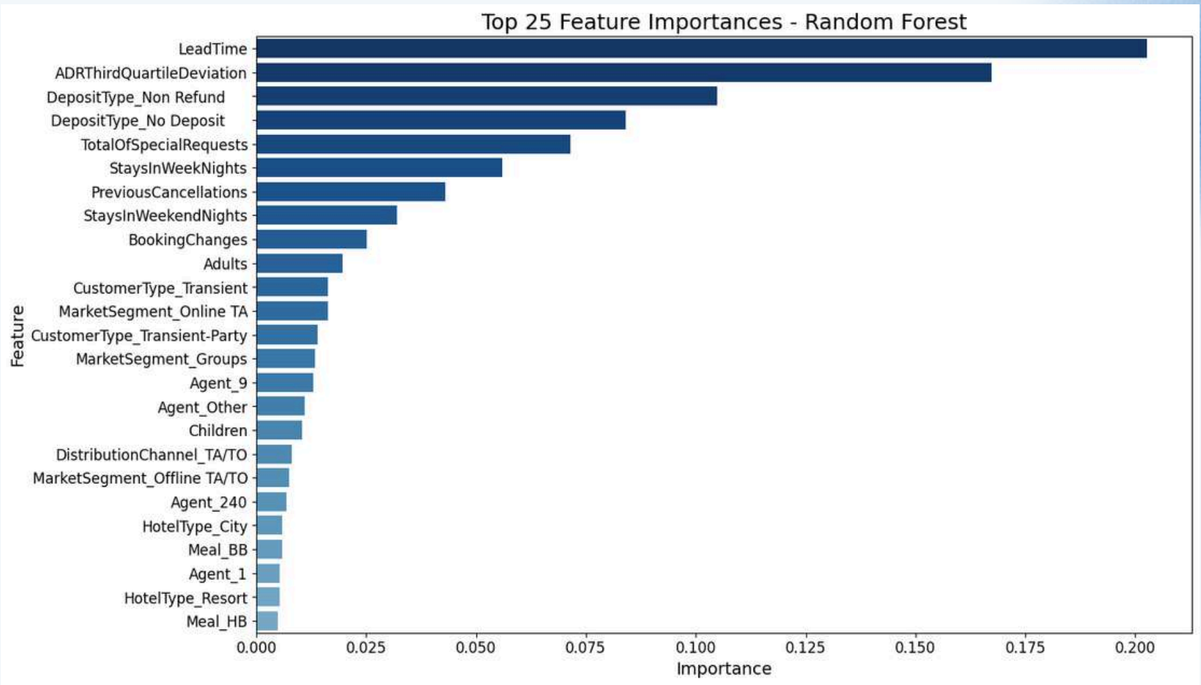
Table 3.5: Hold-Out Evaluation Metrics for Final Models

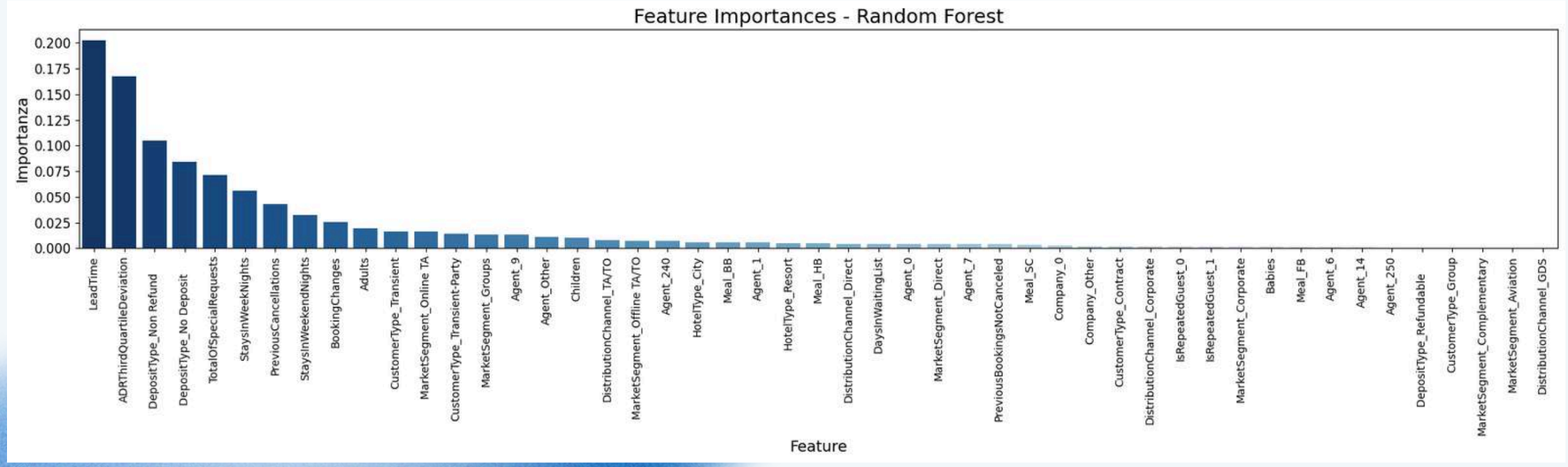
Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Random Forest	0.844	0.794	0.781	0.788	0.914
LightGBM	0.835	0.792	0.752	0.772	0.902
XGBoost	0.831	0.784	0.751	0.767	0.900
CatBoost	0.827	0.785	0.736	0.759	0.892
KNN	0.828	0.768	0.769	0.768	0.894



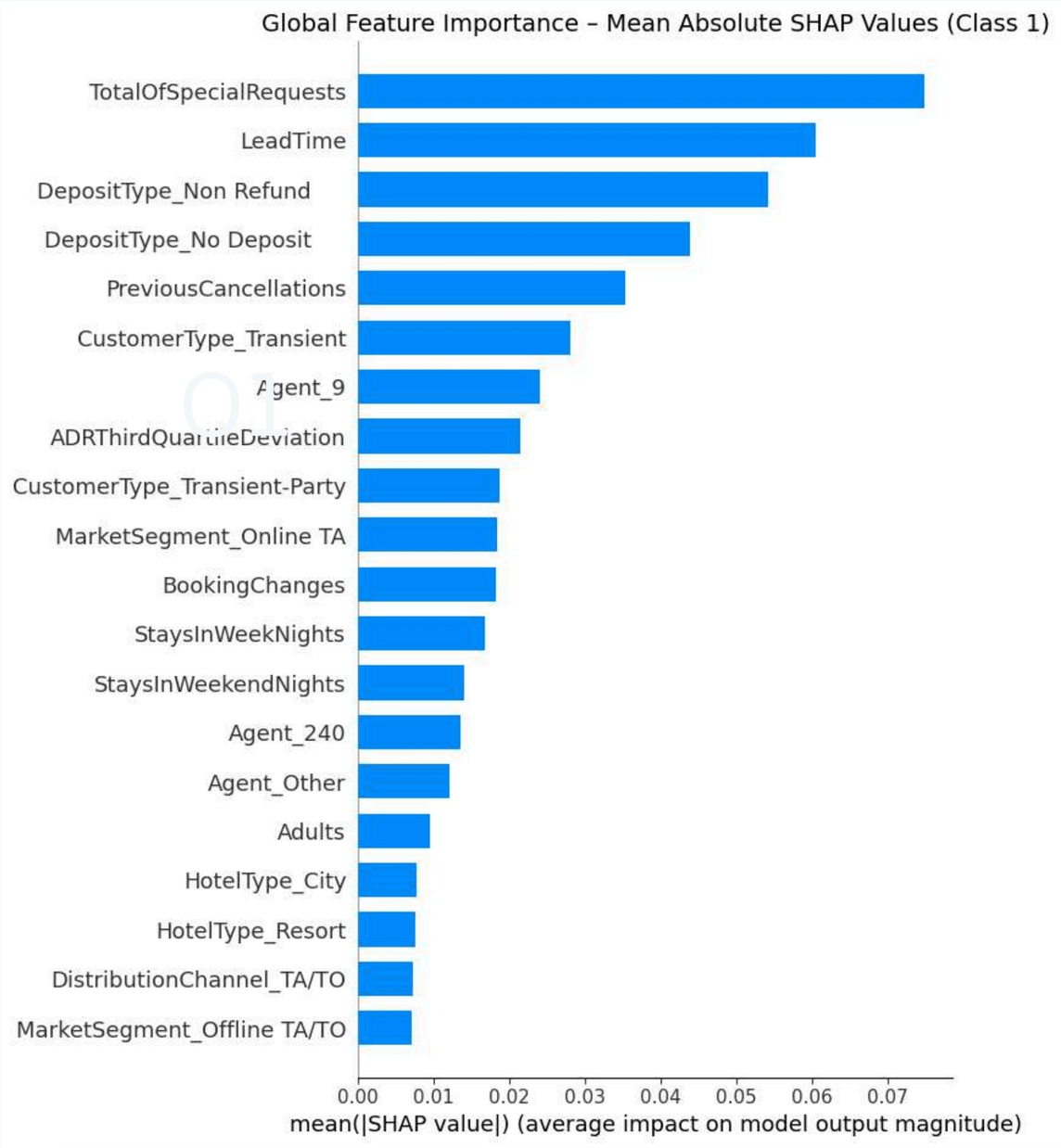


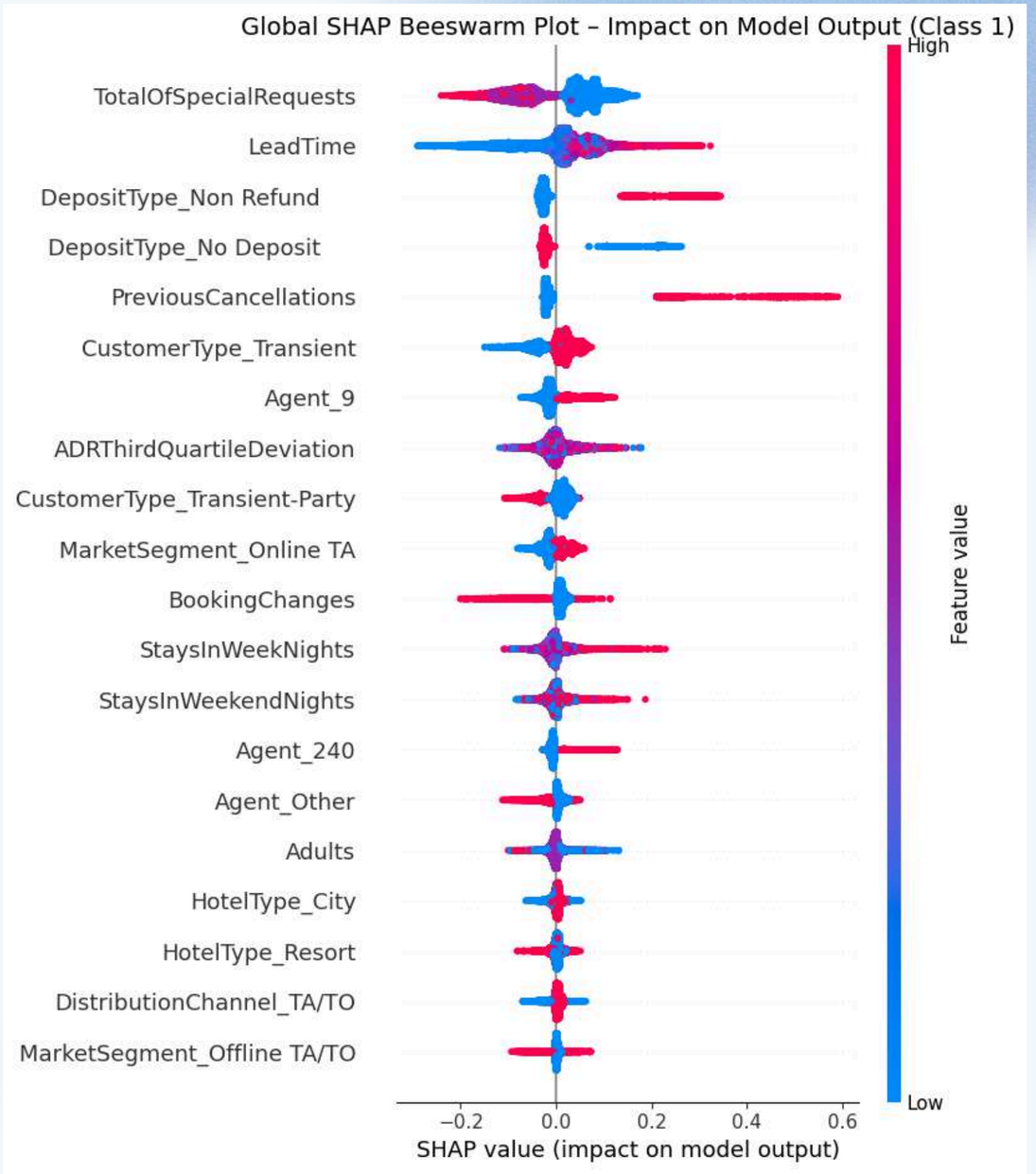
FEATURE IMPORTANCE



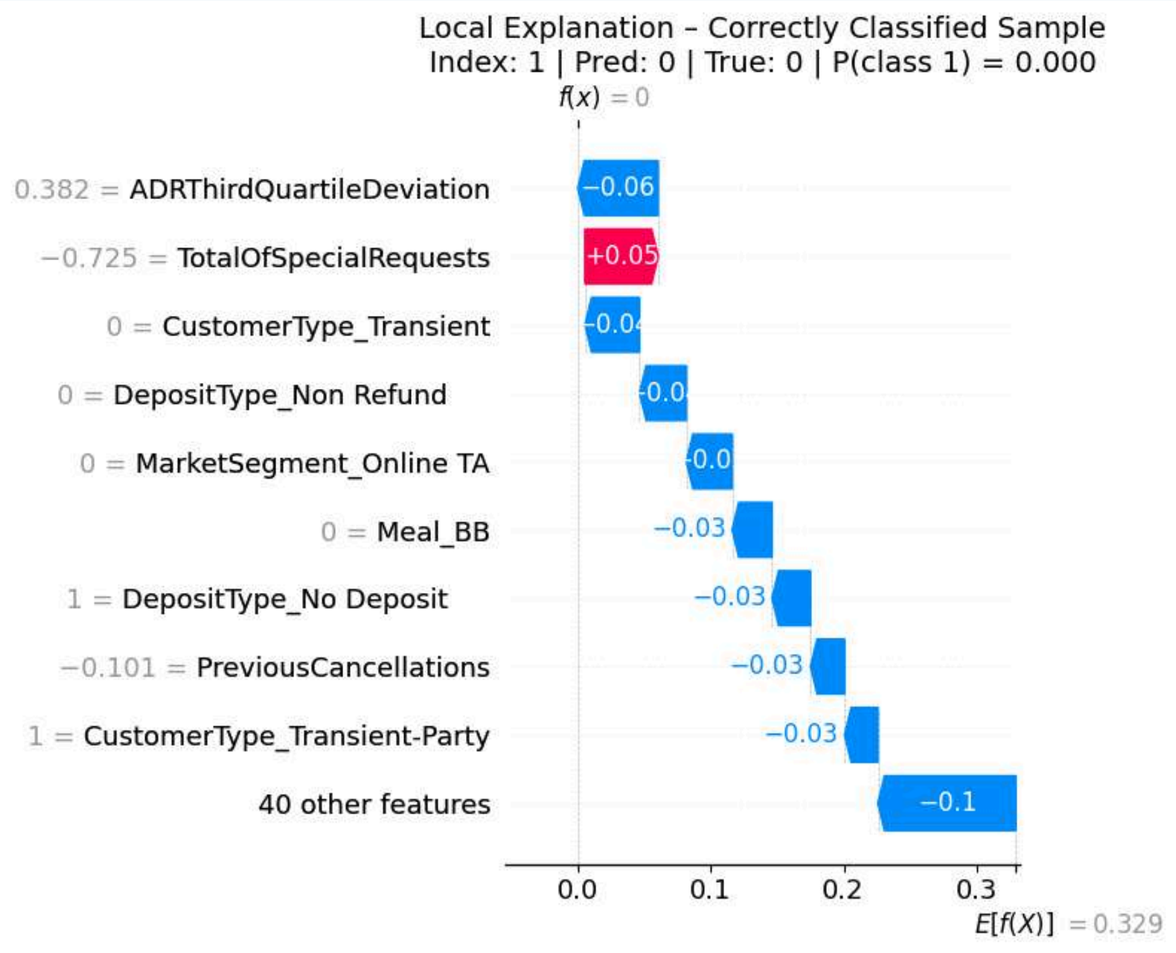


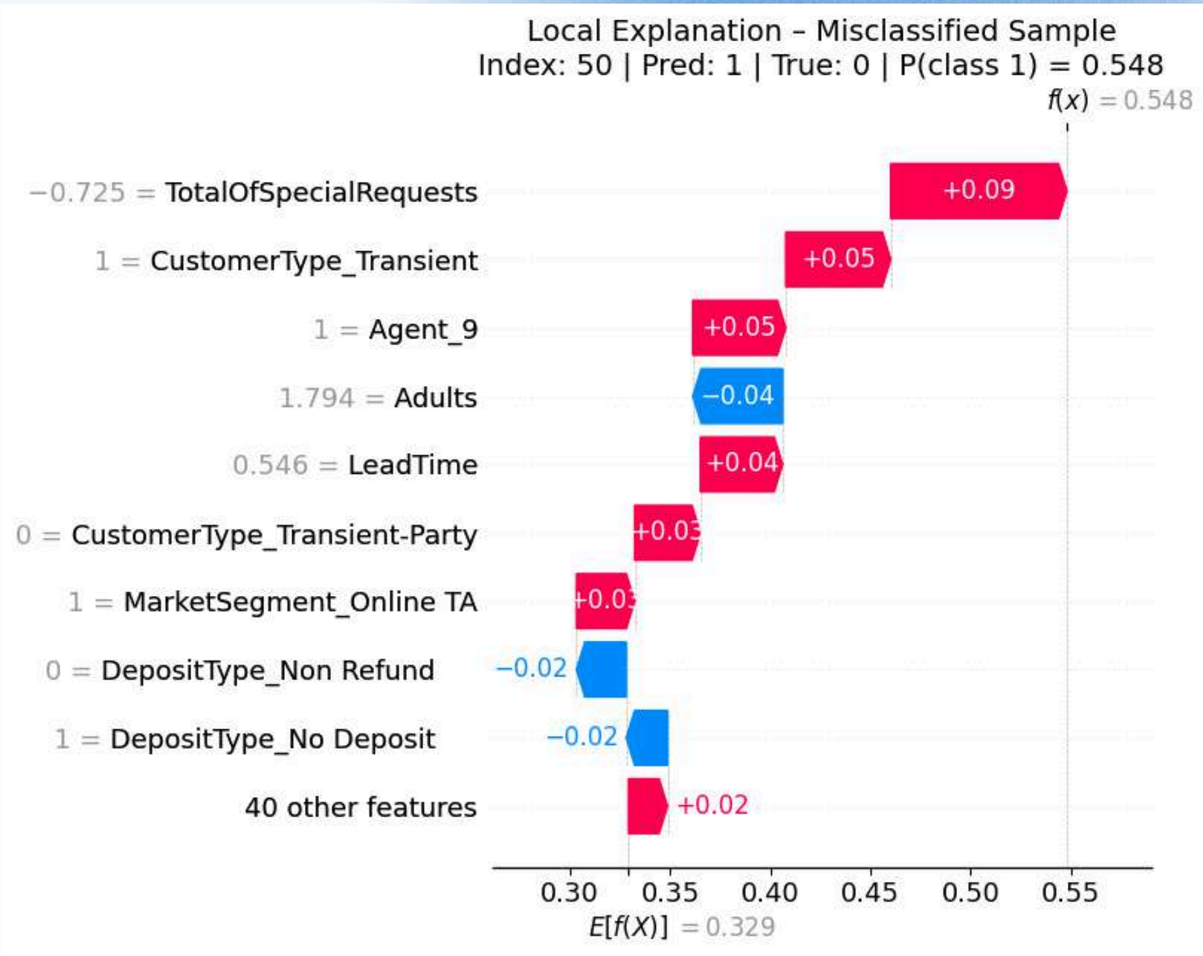
GLOBAL EXPLAINABILITY



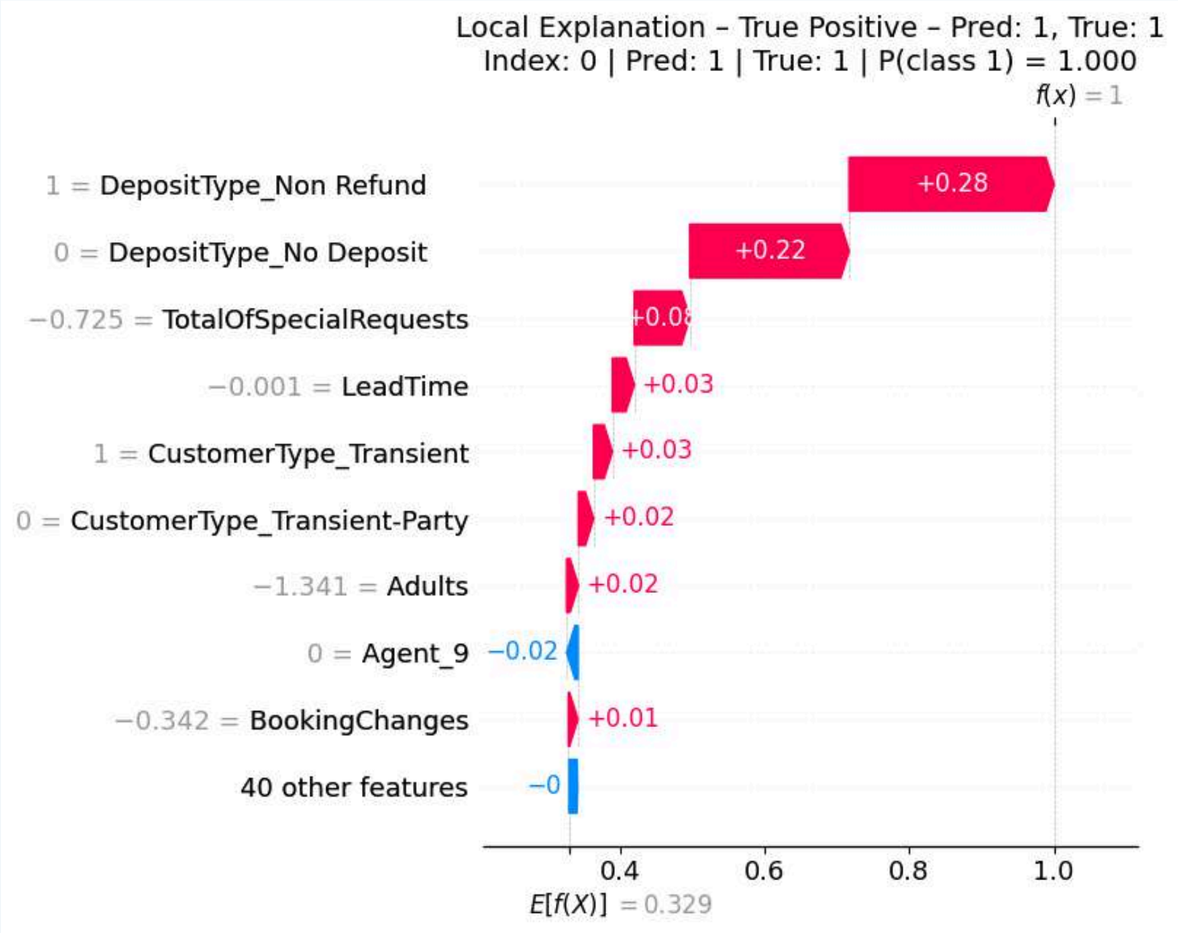


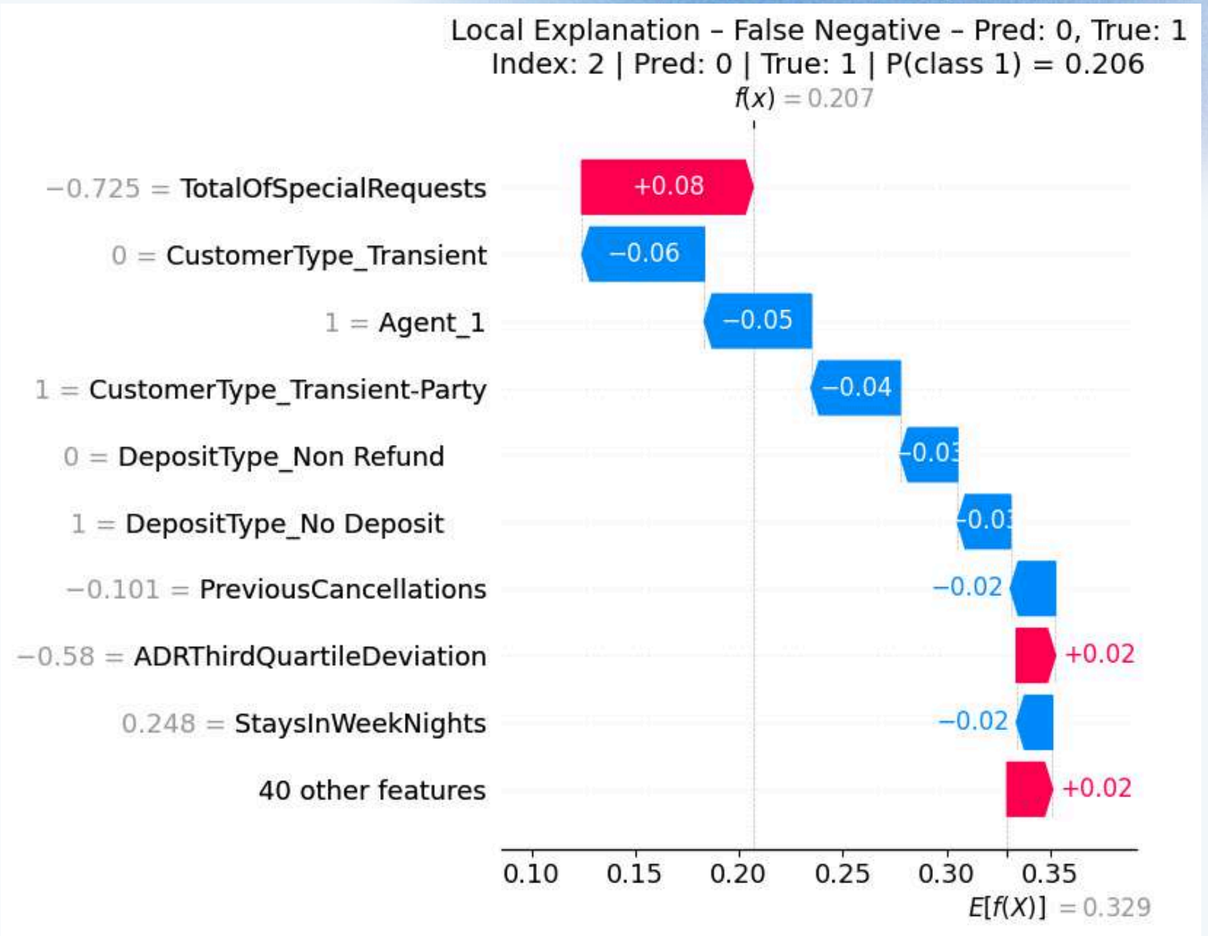
LOCAL EXPLAINABILITY: CLASS 0



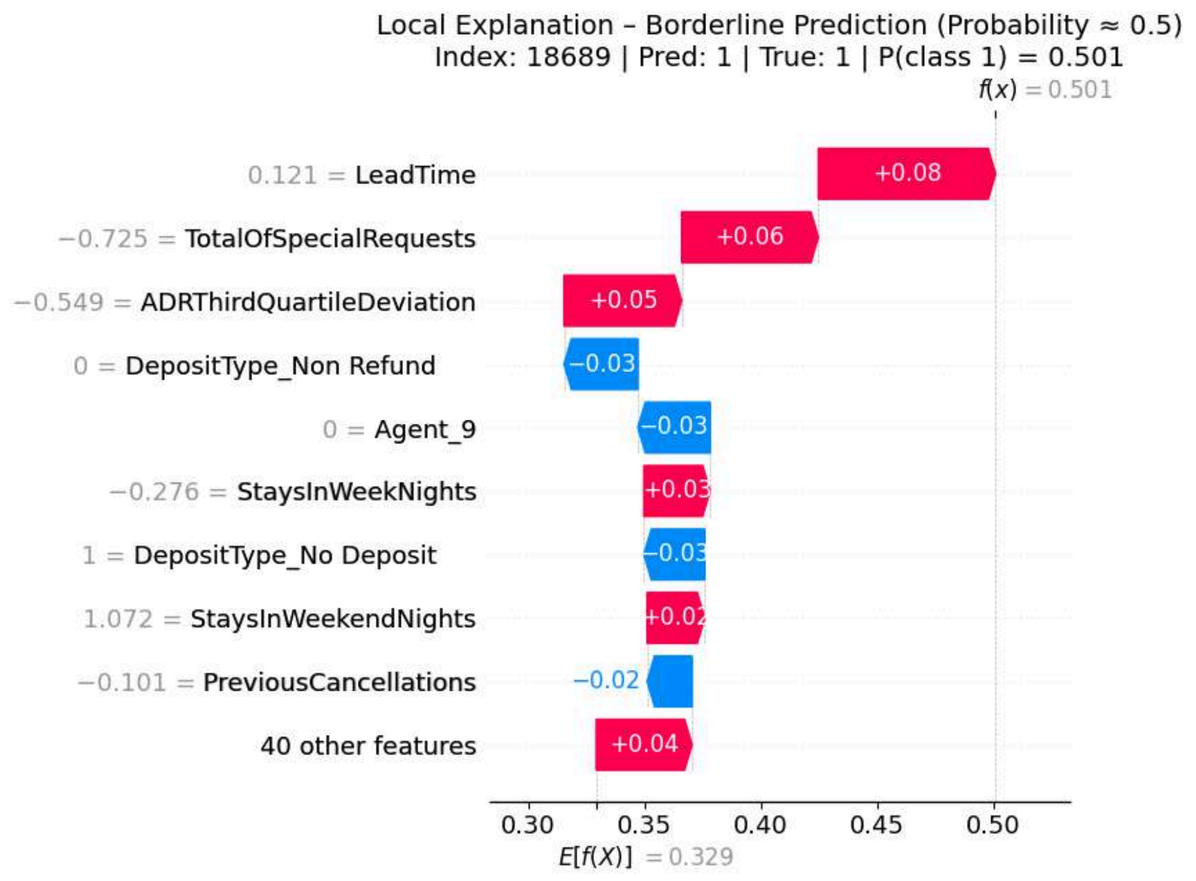


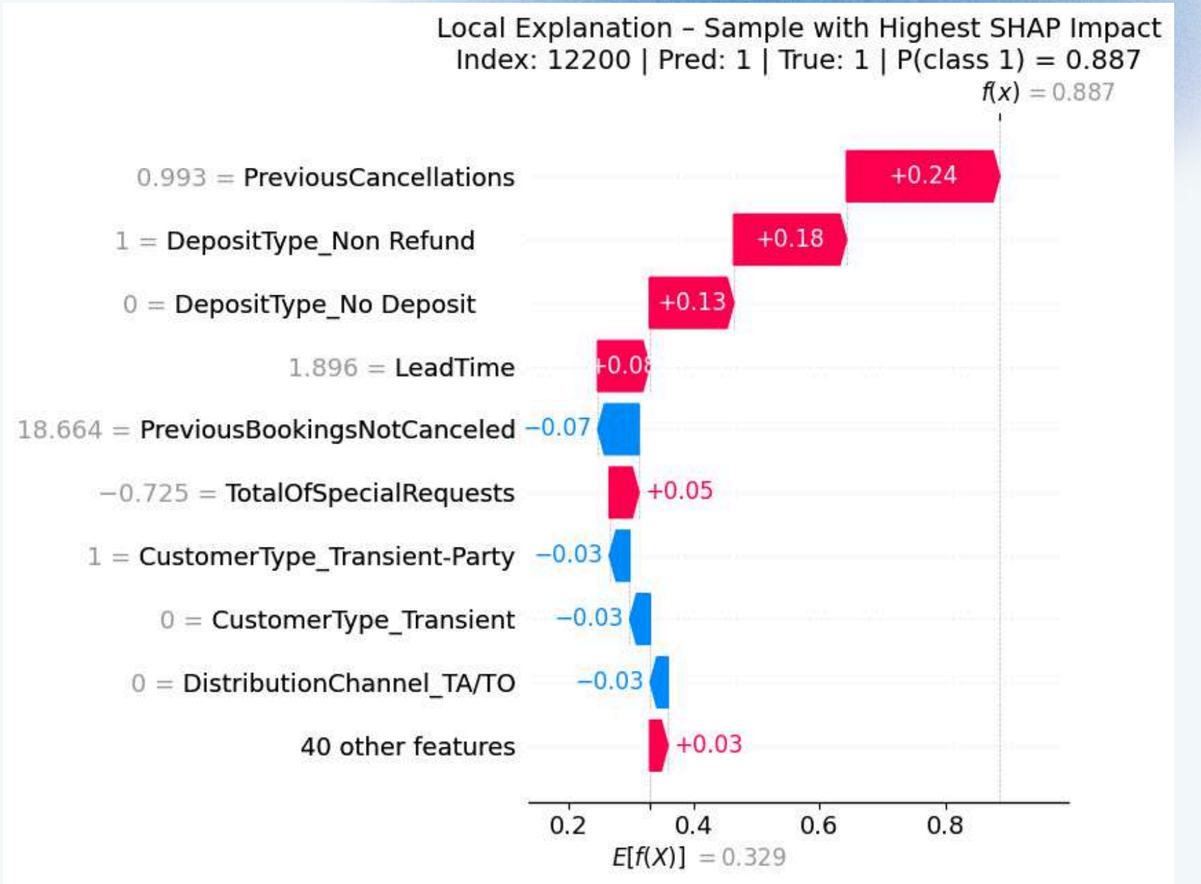
LOCAL EXPLAINABILITY: CLASS 1





LOCAL EXPLAINABILITY: INTERESTING CASES





PREVIOUS STUDIES COMPARISON

In comparing our work with previous studies, we found that both adopted tree-based models—XGBoost in the first and Random Forest in the second. While performance in terms of accuracy was comparable to ours, our approach stands out for several reasons. Themost important difference is that this project included a feature importance analysis and explainability using SHAP values, providing insights not only into performance but also into the model's decision-making process—something missing in earlier research.

Original Paper Evaluation metrics

Hotel.	*Dataset*	*Acc.*	*AUC*	*Prec.*	*Sensit.*	*Specif.*
H1	Train	0.846	0.910	0.839	0.626	0.950
	Test	0.842	0.877	0.811	0.603	0.941
H2	Train	0.857	0.934	0.876	0.793	0.909
	Test	0.849	0.922	0.869	0.779	0.905

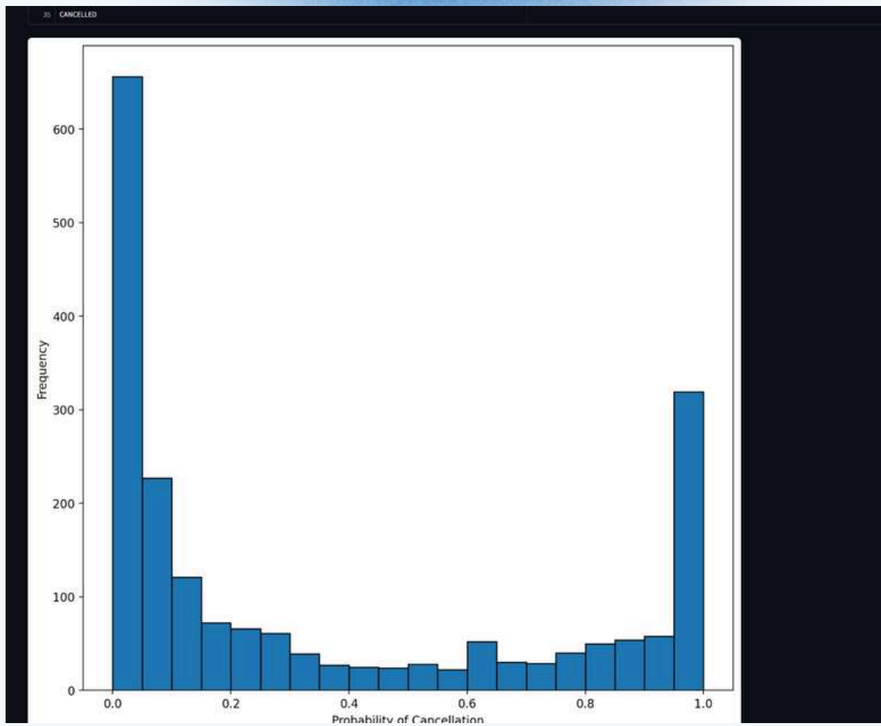
Second paper metrics

Model	Accuracy	Recall	Precision	F1
RF	0.8717	0.7750	0.8646	0.8173
After Tuning RF	0.8725	0.7826	0.8606	0.8197

Table 3.5: Hold-Out Evaluation Metrics for Final Models

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Random Forest	0.844	0.794	0.781	0.788	0.914
LightGBM	0.835	0.792	0.752	0.772	0.902
XGBoost	0.831	0.784	0.751	0.767	0.900
CatBoost	0.827	0.785	0.736	0.759	0.892
KNN	0.828	0.768	0.769	0.768	0.894

GRAPHIC USER INTERFACE



REFERENCES

- **Dataset**:Nuno Antonio, Ana de Almeida, and Luis Nunes. Hotel Booking Demand Datasets. Published in Data in Brief, Volume 22, Pages 41–49.

DOI: https://doi.org/10.1016/j.dib.2018.11.126

- Nuno Antonio, Ana Maria De Almeida, and Luís Nunes. Predicting Hotel Bookings Cancellation with a Machine Learning Classification Model. 2017 IEEE International Conference on Machine Learning and Applications (ICMLA). DOI: 10.1109/ICMLA.2017.00-11
- Zharfan Akbar Andriawan, Ricko, Feri Wijayanto, Satriawan Rasyid Purnama, Adi Wibowo, Adam Sukma Darmawan, and Aris Sugiharto. Prediction of Hotel Booking Cancellation using CRISP-DM. 2020 4th International Conference on Informatics and Computational Sciences (ICICoS). DOI: 10.1109/ICICoS51170.2020.9299011

THANK YOU