Enhancing Hotel Management: Predicting Cancellations with Machine Learning

Team RISM
Irene Na, Min Yang, Stanley Yin, Rob Horrobin
Summer 2024

Audience: Stakeholders in a hotel chain and resident Data Scientists

Berkeley
UNIVERSITY OF CALIFORNIA

Agenda

- Motivation and Summary
- Overall Approach
- Exploratory Data Analysis
- Models and Experiments
- Detail on Winning Model
- Conclusions and Recommendations

Motivation and Summary of Results

- Overall Idea:
  - Hotel Cancellations are costly in both:
    - Lost revenue
    - Reputational risk from overbooking
  - Can we predict cancellation such that we can:
    - Develop a "confirmation list" to minimize last minute cancellations or;
    - Create dynamic pricing model at time of booking to "gross up" for cancellation cost

- This is an existing problem; other researchers have been able to predict cancellation within a 7-day window (80% accuracy)[1]

- Summary of results: We were able to achieve an Accuracy of 89.5% and F1 of 83.0% on Test data after testing several models

Berkeley
UNIVERSITY OF CALIFORNIA

Overall Approach

- Exploratory Data Analysis and Feature Engineering
- Baseline design of Logistic Regression
- Tested several models:
  - Random Forest
  - KNN
  - Deep Neural Net
  - XG Boost
- Compared against F1 Score as metric of interest
- Selected a winning model for application

Data Size, Source and Features

Data is 36,275 Rows x 19 Columns

Group Characteristics
no_of_adults
no_of_children

Booking Characteristics
no_of_weekend_nights
no_of_week_nights
type_of_meal_plan
required_car_parking_space
room_type_reserved
market_segment_type
avg_price_per_room
no_of_special_requests

Date/Time Characteristics
lead_time
arrival_year
arrival_month
arrival_date
day_name (derived)

Customer Records
repeated_guest
no_of_previous_cancellations
no_of_previous_bookings_not_canceled

Exploratory Data Analysis (EDA)

Data Cleansing and Evaluation

- Checking shape, labels
- Checking data types
- Checking for null values
- Checking for primary keys
- One date (2/29) where no leap year; updated to 2/28
- Identifying our variable of interest " booking_status"
- merge entries with >=3 no_children in one group due to low value counts
- Shuffle, split data into Train/Test/Validate

Data Visualization

- Split into Numeric/Categorical
- Checking for correlations
- Looking for outliers and imbalanced data
- Evaluating cancellation rate for smaller populations for potential binning

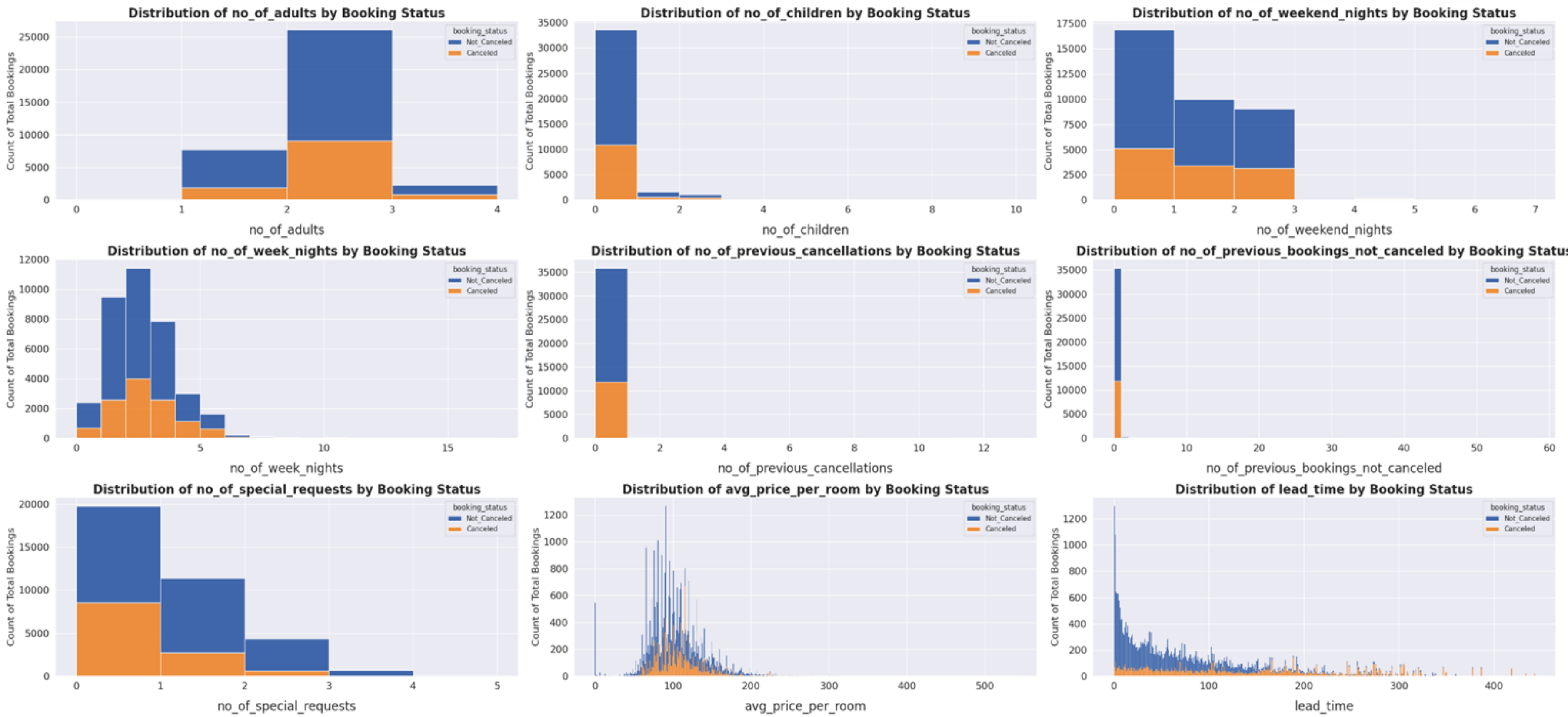
Feature Engineering

- One hot encoding
- Standardizing numerical data
- Oversampling for imbalance data
- Dropping data
- Adding "Day of Week" feature

- **In general our data has minimal issues and is well fit for our purpose**

Berkeley
UNIVERSITY OF CALIFORNIA

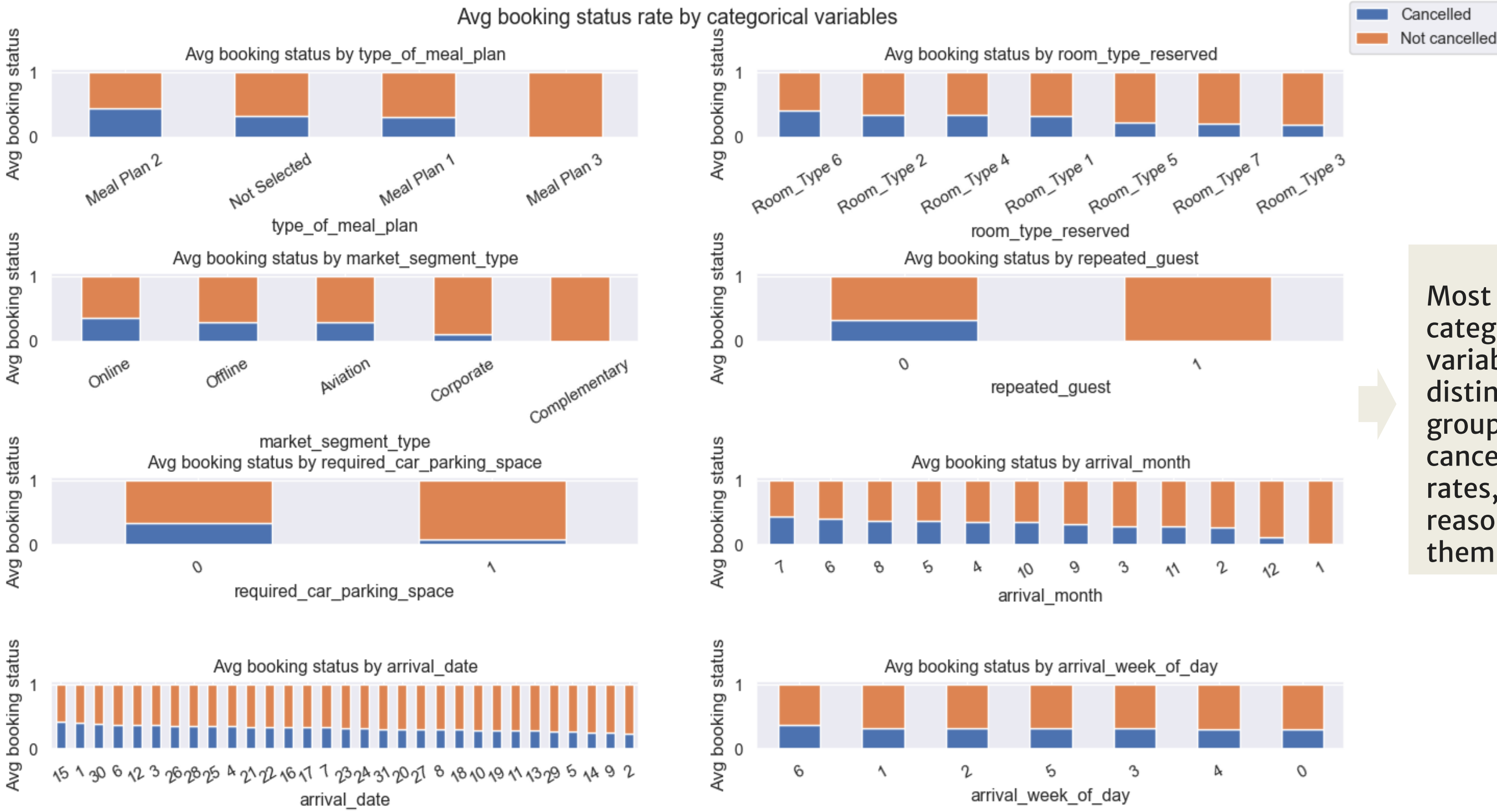
EDA - Numerical Variables - Histogram



- **<u>We performed some consolidation and outlier treatment to improve model performance while observing lead time's positive correlation with cancellation</u>**

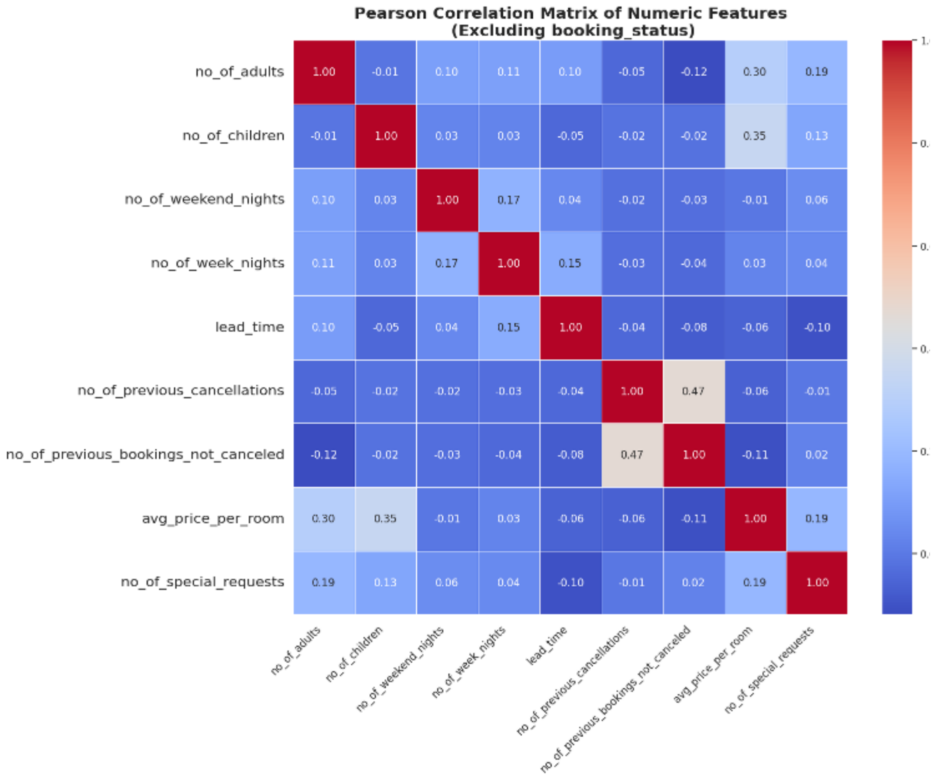
EDA - Categorical Variables

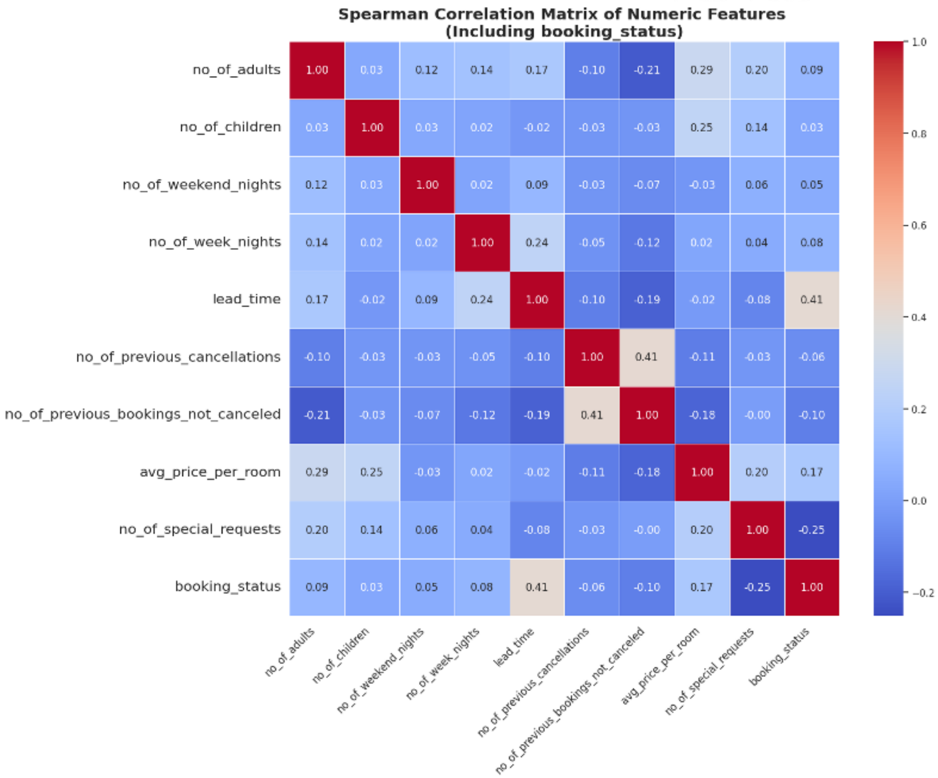
Cancel vs Non-cancel Rate by Sub-group



Most categorical variables have distinct sub-group cancellation rates, giving us reason to keep them

EDA- Numerical Variables - Pearson & Spearman Correlation Matrix





- Did not see any features with significant correlation
- Addressed concerns for multicollinearity
- We did see the highest correlation with Booking Status as follows:
  - Lead time
  - Number of special requests
  - Avg. room price

Berkeley
UNIVERSITY OF CALIFORNIA

Model Selection Road Map

- Overall plan:
  - We will start from baseline model, followed by various advanced models with consistent evaluation metrics.
  - Finally we will put all models side by side and choose one as recommendation
- Metrics we chose to tune each model for hyperparameters:
  - **F1**: most relevant to the concerned question given the business context
  - Both missing a cancellation and predicting cancellations wrongly cost the hotel meaningfully, we want to strike a balance between precision and recall.
- Metrics we chose to evaluate models – three metrics for three purposes
  - Performance: F1 on test data (we want a balanced recall vs precision)
  - Model generalization: overall accuracy on train vs validation vs test
  - Model fairness: accuracy per group on test data (minority class matters)

Berkeley
UNIVERSITY OF CALIFORNIA

Baseline Model Briefing

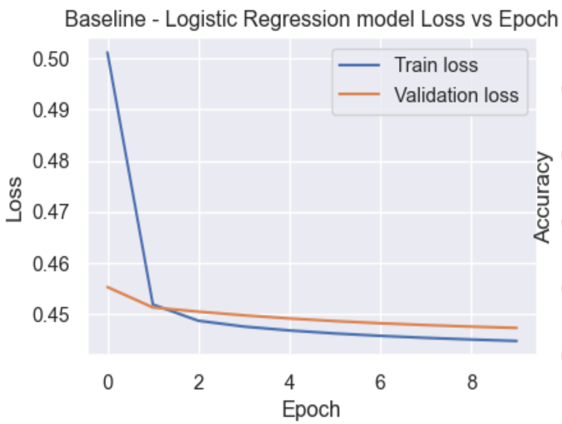
Modeling related assumptions and feature decisions

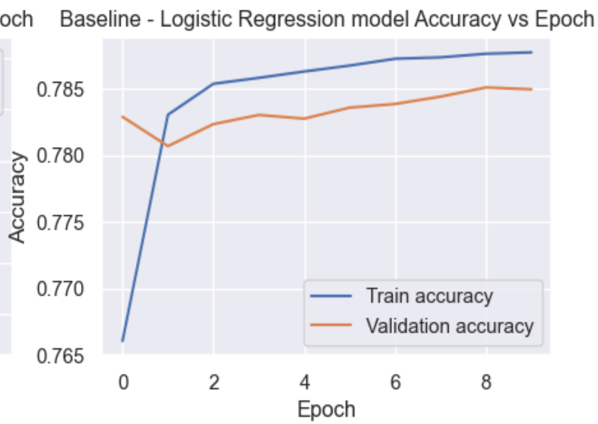
- Treat data as cross-sectional: despite that arrival time spans 2017 and 2018, given the context, we assume no time trend.
- Keep categorical variables with small group counts, but with warning (e.g. room type 3), given we observed distinct average cancellation rate by groups.
- Adopt oversampling to balance target label groups, due to unfair model results, if otherwise.

Baseline model: Logistic regression

- Keep most features except for Booking_ID, year; drop one column per categorical variable in one-hot representation, to avoid perfect multicollinearity.
- In hyperparameter tuning, we use **F1** to choose the best performing model

Based on hyperparameter tunings, to balance the performance based on F1 and convergence efficiency, we chose the model with below hyperparameters: *learning_rate = 0.005, batch_size = 50, epoch=10.*



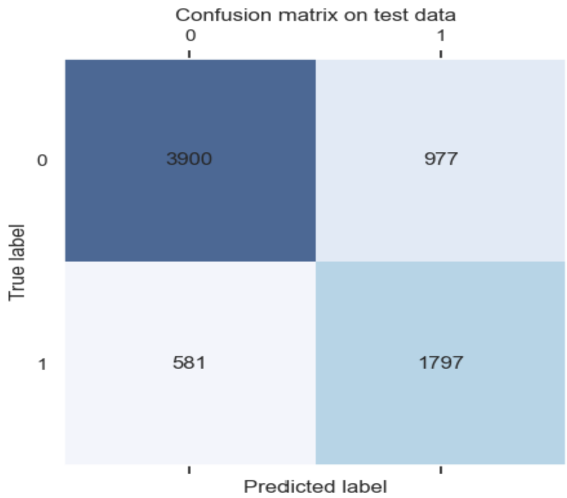


Berkeley
UNIVERSITY OF CALIFORNIA

Baseline Model Evaluation

Evaluation methodologies

- We evaluated the model through three major sets of metrics for different purposes:
  - F1 (performance); overall accuracy (generalization); per group accuracy (fairness)



	Non_cancellation	Cancellation
Accuracy by class	80.0%	75.6%
Precision	87.0%	64.8%
Recall	80.0%	75.6%
F1	83.4%	69.8%

- **Performance – Test F1/ recall /precision: (Cancellation group):**
  - F1: 69.8%
  - Recall: 75.6%
  - Precision: 64.8%
  - Balanced and desirable (we care to catch cancellation more than being precise)
- **Generalization – overall accuracy:**
  - Train: 78.8%
  - Validation: 78.5%
  - Test: 78.5%
  - Model generalizes well
- **Model fairness – Test accuracy per group:**
  - non-cancellation: 80.0%
  - cancellation: 75.6%
  - Model is relatively fair thanks to over-sampling
  - Would have been 88% vs 66% (non__cancel vs cancel) without over-sampling

Berkeley UNIVERSITY OF CALIFORNIA

Random Forest Model Briefing

Modeling related assumptions and feature decisions:

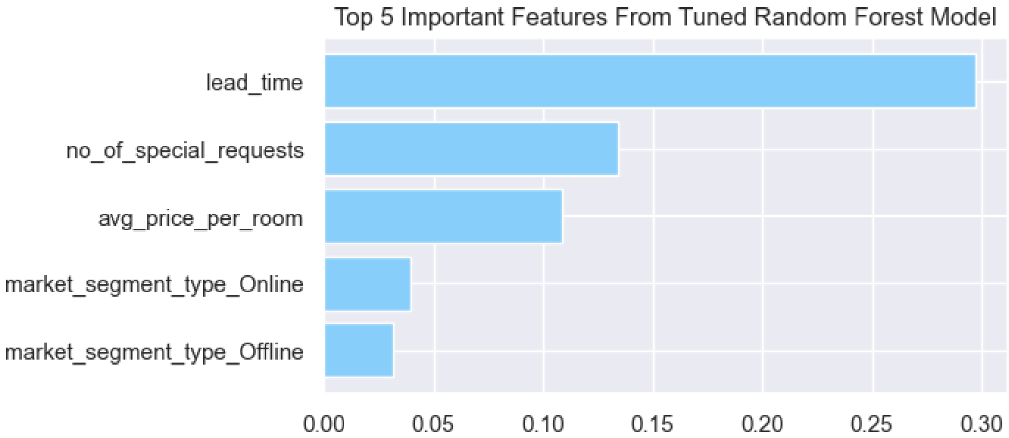
- Unlike logistic regression model, we included all categorical variables columns to train the tree model, given it does not have multicollinearity concern.

Model Results Summary

- In hyperparameter tuning, we range over different combinations of n_estimators, max_depth, max_features and min_sample_split in order to the best performing model based on F1.
- The resulting model's top five important features are both intuitive and consistent with EDA observations (lead_time, no of special requests, average room price, online or offline order).

Based on hyperparameter tunings, to balance the performance based on F1 and convergence efficiency, we chose the model with below hyperparameters:

```
RandomForestClassifier
RandomForestClassifier(max_depth=17, n_estimators=30, n_jobs=3,
                       random_state=1234)
```



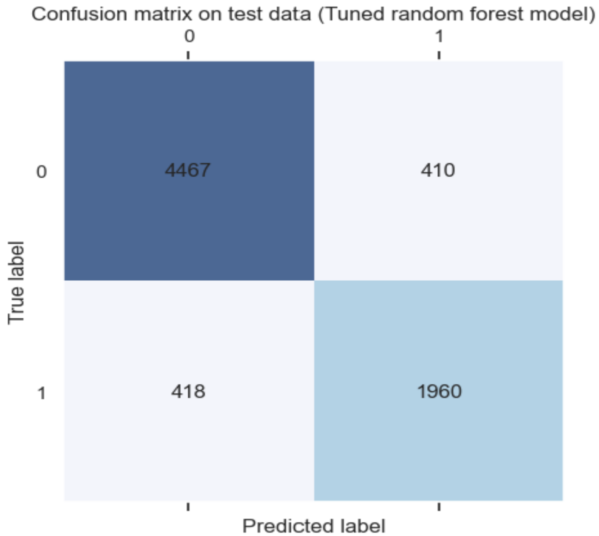
Note: Bar length indicates the magnitude of the importance, without direction

Berkeley
UNIVERSITY OF CALIFORNIA

Random Forest Model Evaluation

Evaluation methodologies

- We evaluated the model through three major sets of metrics for different purposes:
  - F1 (performance); overall accuracy (generalization); per group accuracy (fairness)



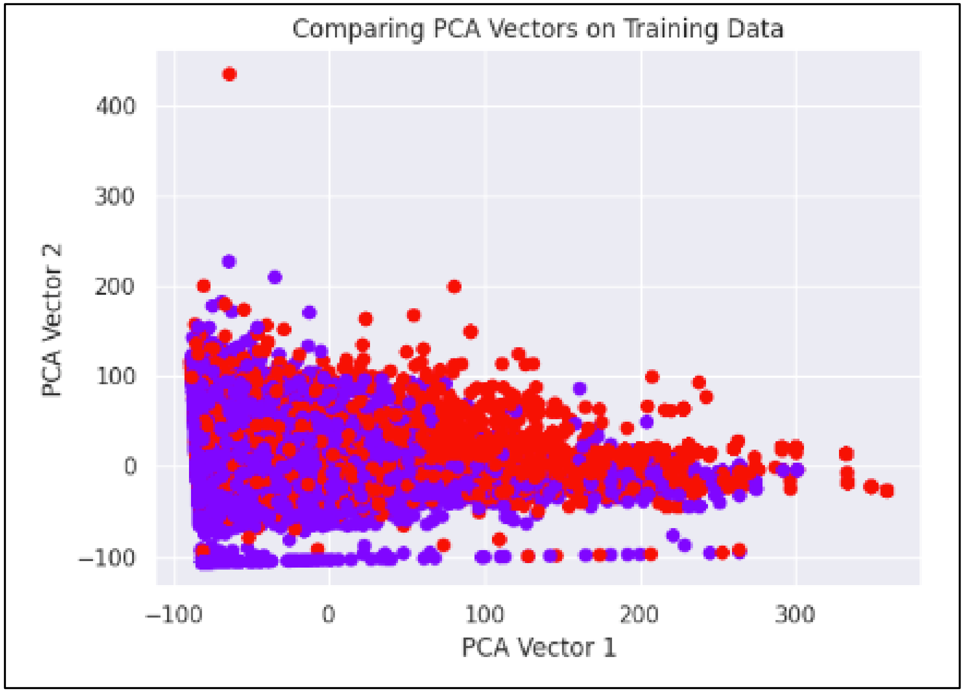
	Non_cancellation	Cancellation
Accuracy by class	91.6%	82.4%
Precision	91.4%	82.7%
Recall	91.6%	82.4%
F1	91.5%	82.6%

- **Performance – Test F1/ recall /precision: (Cancellation group):**
  - F1: 82.6%
  - Recall: 82.4%
  - Precision: 82.7%
  - Very balanced and meaningfully improved over baseline model
- **Generalization – overall accuracy:**
  - Train: 92.3%
  - Validation: 87.8%
  - Test: 88.6%
  - Model generalizes well
- **Model fairness – Test accuracy per group:**
  - non-cancellation: 91.6%
  - cancellation: 82.4%
  - Model is relatively fair

KNN Model Briefing

Modeling related assumptions and feature decisions:

- Dedimensionalized and viewed using Principal Components Analysis (PCA)
- Binary Classifier approach using K-neighbor Classifier for Cancelled/Did Not Cancel
- Removed Categorical variables and only used continuous in KNN
- Used Grid Search to check number of neighbors [2:50] with aim to maximize F1 score
- Resulted in 2 neighbors as optimal
- Otherwise, leveraged oversampling methodology from baseline model
- For robustness, tested KNN with and without categorical features as inputs



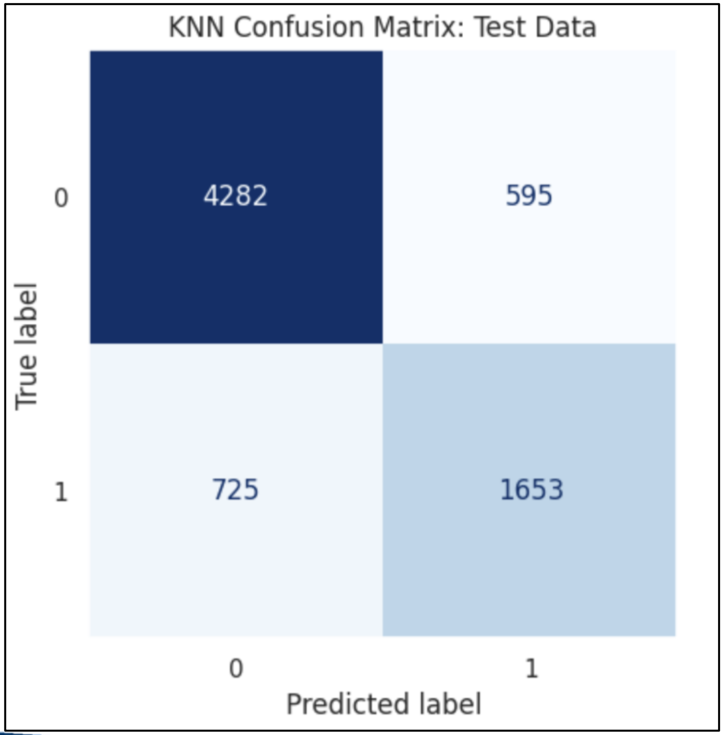
Did Not Cancel Cancelled

Berkeley UNIVERSITY OF CALIFORNIA

KNN Model Evaluation

Evaluation methodologies

- We evaluated the model through three major sets of metrics for different purposes:
  - F1 (performance); overall accuracy (generalization); per group accuracy (fairness)



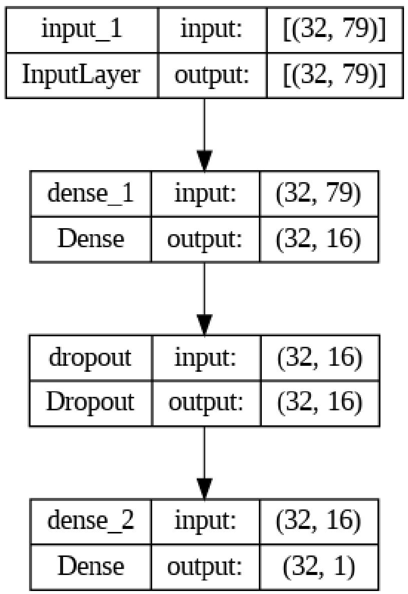
- **Performance – Test F1/ recall /precision: (Cancellation group):**
  - F1: 71.5%
  - Recall: 69.5%
  - Precision: 73.5%
  - Generally balanced
- **Generalization – overall accuracy:**
  - Train: 96.6%
  - Validation: 82.1%
  - Test: 81.8%
  - Model shows some evidence of overfit
- **Model fairness – Test accuracy per group**
  - non-cancellation: 87.8%
  - cancellation: 69.5%
  - Model is somewhat unfair, favoring non-cancellation

Berkeley
UNIVERSITY OF CALIFORNIA

Neural Net Model Briefing

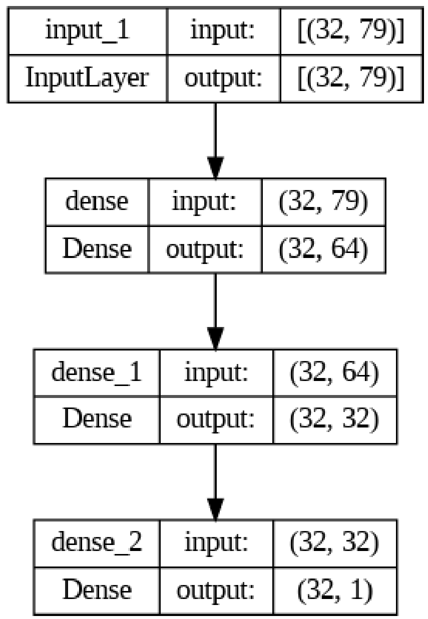
Model 0

val_f1: 0.71
val_precision: 0.63
val_recall: 0.82
val_binary_accuracy: 0.78



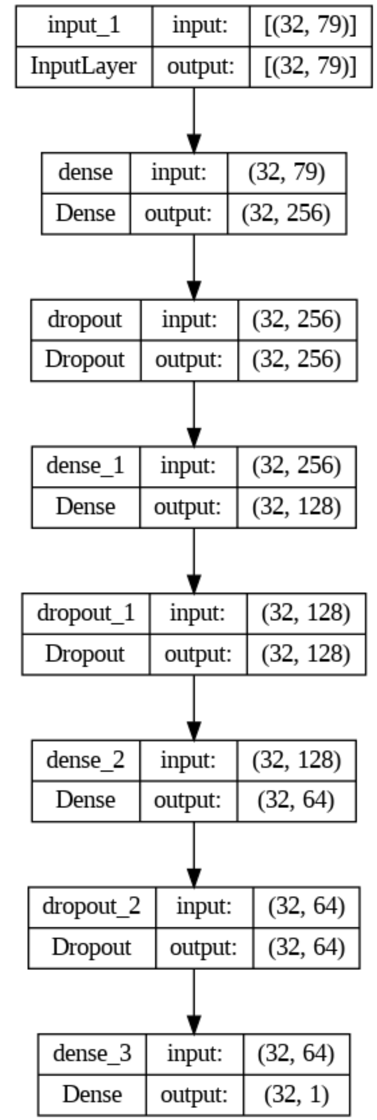
Model 1

val_f1: 0.71
val_precision: 0.64
val_recall: 0.82
val_binary_accuracy: 0.79



Model 2

val_f1: 0.76
val_precision: 0.69
val_recall: 0.85
val_binary_accuracy: 0.83



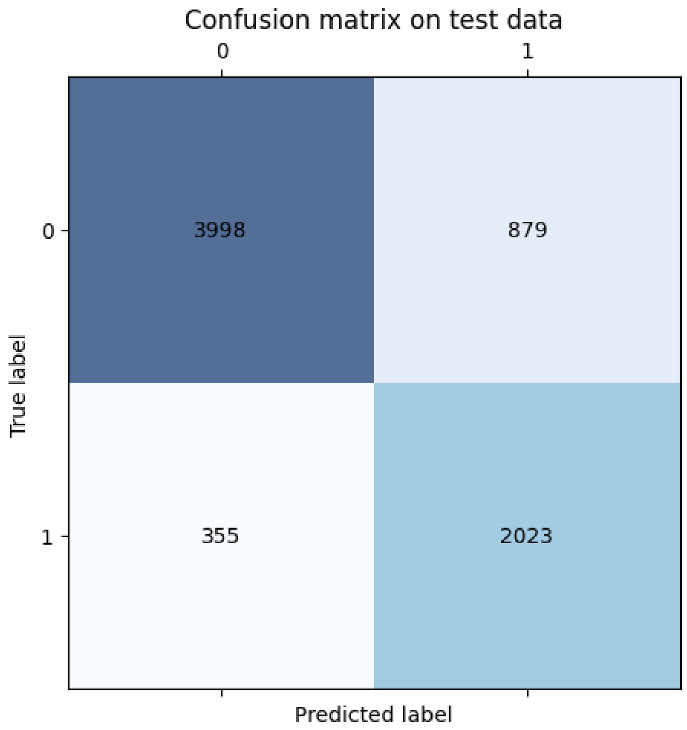
Dropout layers and kernel regularization was applied where necessary to reduce overfitting

Berkeley
UNIVERSITY OF CALIFORNIA

Neural Net Model Evaluation

Evaluation methodologies

- We evaluated the model through three major sets of metrics for different purposes:
  - F1 (performance); overall accuracy (generalization); per group accuracy (fairness)



- **Performance – Test F1/ recall /precision: (Cancellation group):**
  - F1: 76.6%
  - Recall: 85%
  - Precision: 69.7%
  - Some bias towards Non-Cancellation
  - Meaningful improvement over baseline

- **Generalization – overall accuracy:**
  - Train: 87.6%
  - Validation: 82.7%
  - Test: 85%
  - Model generalizes well

- **Model fairness – Test accuracy per group:**
  - non-cancellation: 81.9%
  - cancellation: 85%
  - Model is relatively fair

Berkeley
UNIVERSITY OF CALIFORNIA

XGBoost Model Briefing and Hyperparameters Tuning

- Offers extensive hyperparameter tuning options to optimize for best model performance. Regularization features help prevent the model overfitting issue.
- **Suitable for Hotel Cancellation Prediction:** Its ability to model non-linear dependencies and feature interactions makes it highly effective for predicting events such as hotel cancellations.

XGBoost Model Random Data Split - Training (60%) /Validation (20%) /Test (20%) SMOTE Oversampling						
Model Hyperparameters	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**
Max_depth	**6**	[3, 6, 9]	[3, 6, 9]	[3, 6, 7]	[3, 6, 7]	[9]
Learning_rate	**0.1**	[0.01, 0.1, 0.2]	[0.01, 0.05, 0.1]	[0.01, 0.05, 0.1]	[0.01, 0.05, 0.1]	[0.1]
n_estimators	**100**	[50, 100, 200]	[20, 50, 100, 200]	[220]	[100, 200]	[300]
Colsample_bytree	No	[0.3, 0.7]	[0.3, 0.7]	[0.7, 0.8]	[0.7, 0.8]	[0.8]
GridSearch	No	Yes	Yes	Yes	Yes	Yes
Eearly-stopping-rounds	**10**	No	No	15	15	No
K-fold Cross Validation	No	3-fold	5-fold	5-fold	5-fold	8-fold
Subsample	No	No	No	[0.6, 0.7]	[0.6, 0.7]	No
L2 Regularization - lambda	No	No	[1, 1.5, 2]	[2.5, 3]	[2.5, 3]	No
L1 Regularization - alpha	No	No	No	[0.3, 0.4]	[0.3, 0.4]	[0.3]
gamma Regularization	No	No	No	No	[0.2, 0.3, 0.5]	[0.3]
Best Model Parameter	N/A	Max_depth: 9 Learning_rate: 0.2 n_estimators: 200 Colsample_bytree: 0.7	Max_depth: 9 Learning_rate: 0.1 n_estimators: 200 Colsample_bytree: 0.7 lambda: 1	Max_depth: 7 Learning_rate: 0.1 n_estimators: 220 Colsample_bytree: 0.7 subsample: 0.7 lambda: 2.5 alpha: 0.3	Max_depth: 7 Learning_rate: 0.1 n_estimators: 200 Colsample_bytree: 0.8 subsample: 0.7 lambda: 2.5 alpha: 0.3 gamma: 0.2	Max_depth: 9 Learning_rate: 0.05 n_estimators: 300 Colsample_bytree: 0.5 alpha: 0.3 gamma: 0.3

Berkeley
UNIVERSITY OF CALIFORNIA

XGBoost Model Evaluation

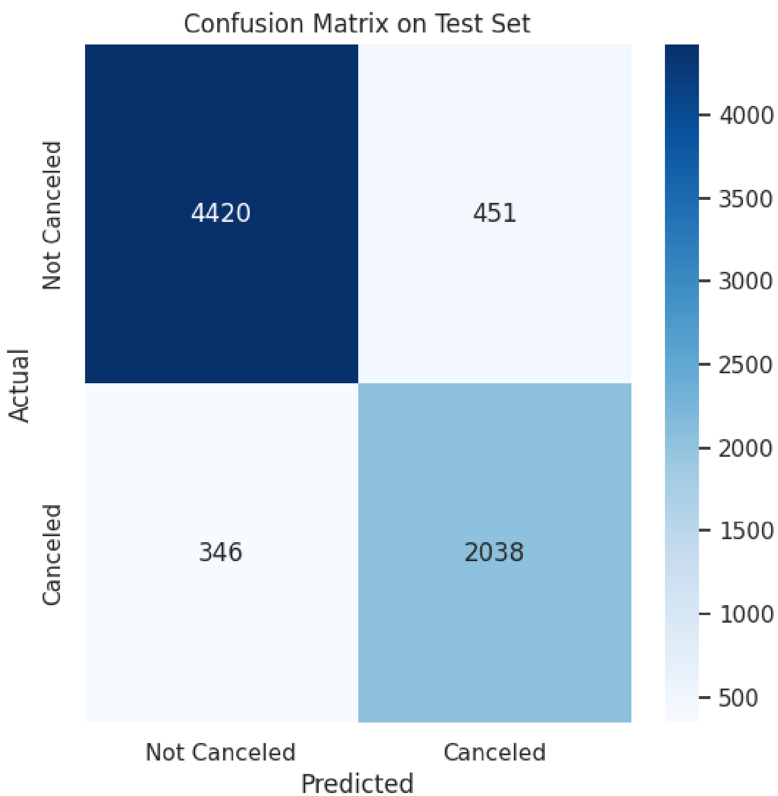
XGBoost Model Random Data Split - Training (60%) /Validation (20%) /Test (20%) SMOTE Oversampling							
Model Hyperparameters		Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Model Performance Measurement							
Training F1		90.4%	97.7%	95.0%	94.99%	92.4%	93.85%
Validation F1		87.5%	89.4%	89.1%	89.08%	88.5%	89.02%
Test F1		87.6%	89.5%	89.2%	89.15%	88.6%	89.07%
Not_Canceled	Training F1	90.5%	97.7%	95.03%	95.03%	92.5%	93.9%
	Test F1	90.8%	92.3%	92.03%	92.03%	91.6%	92.0%
Canceled	Training F1	90.3%	97.7%	94.95%	94.95%	92.3%	93.8%
	Test F1	81.1%	83.6%	83.27%	83.27%	82.5%	83.6%
Model Summary		Initial model with simple parameters; demonstrated basic performance but lacked optimization and advanced features to improve F1.	Introduced GridSearch and expanded hyperparameters but experienced overfitting, suggesting high complexity without adequate regularization	Incorporated L2 regularization to address overfitting, improving generalization but not optimally balancing overfitting with performance	Incorporated both L1 and L2 regularization, added early stopping, applied more targeted hyperparameters , but still not optimally balancing overfitting with performance	Introduced gamma regularization, applied more targeted hyperparameters, achieving a good balance between overfitting reduction and performance enhancement, but still not optimal	**Winning Model** Introduced gamma regularization, applied more targeted hyperparameters, achieving the best balance between overfitting reduction and performance enhancement

Berkeley
UNIVERSITY OF CALIFORNIA

XGBoost Winning Model Evaluation

Evaluation methodologies

- We evaluated the model through three major sets of metrics for different purposes:
  - F1 (performance); overall accuracy (generalization); per group accuracy (fairness)



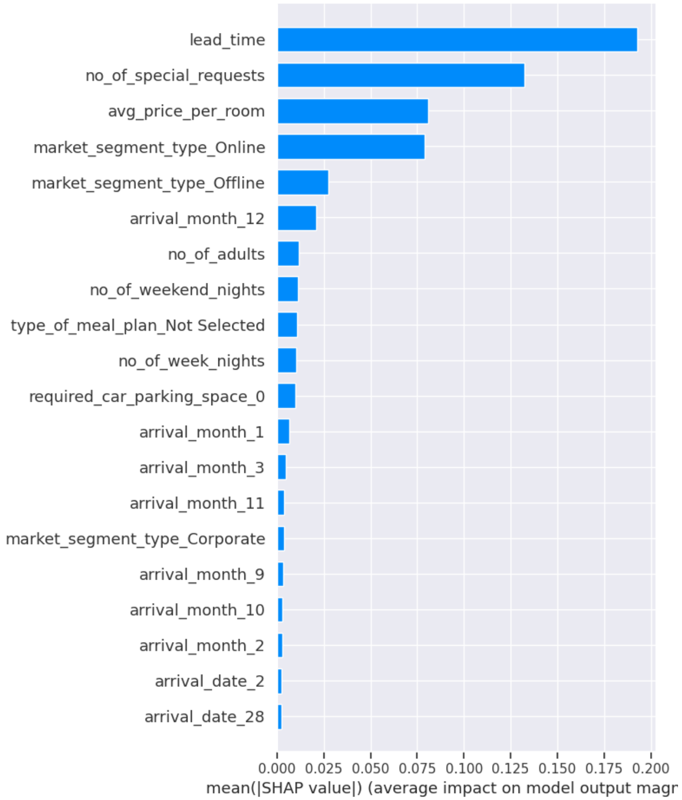
- **Performance - Test F1/ recall /precision: (Cancellation group):**
  - F1: 83.6%
  - Recall: 85.5%
  - Precision: 81.9%
  - Some bias towards Non-Cancellation
  - Meaningful improvement over baseline

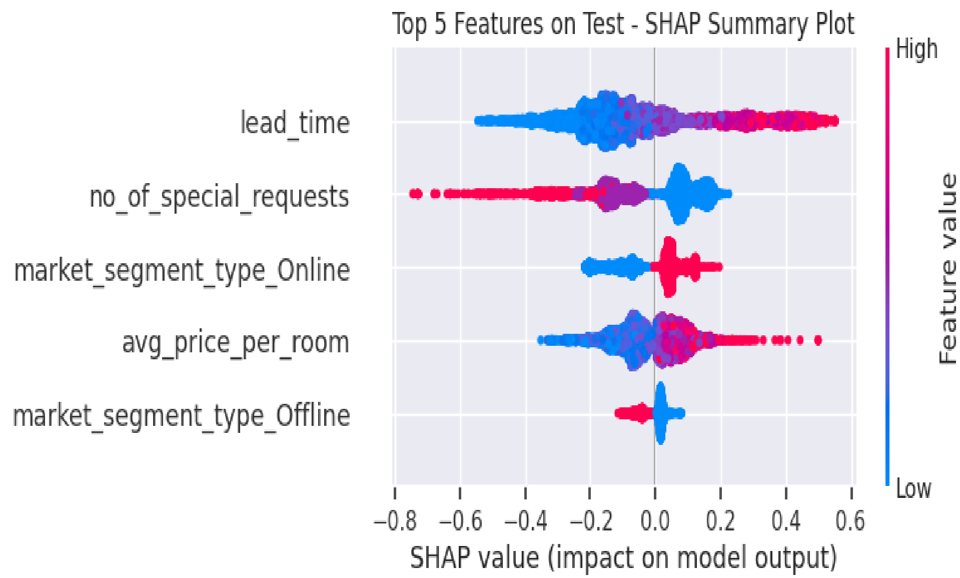
- **Generalization - overall F1:**
  - Train: 93.9%
  - Validation: 89.0%
  - Test: 89.0%
  - Model generalizes well

- **Model fairness - Test F1 per group:**
  - non-cancellation: 92.0%
  - cancellation: 83.6%
  - Model is relatively fair

Berkeley
UNIVERSITY OF CALIFORNIA

Feature Importance: SHAP





Lead Time: High lead times (red) increase the likelihood of cancellation (positive SHAP value), whereas low lead times (blue) decrease the likelihood of cancellation (negative SHAP value).

Number of Special Requests: More special requests (red) increase the likelihood of cancellation, whereas fewer special requests (blue) decrease the likelihood of cancellation.

Berkeley
UNIVERSITY OF CALIFORNIA

Side by Side: Model Test Performance Comparison

Metrics	Baseline (Logistic regression)	KNN	Neural Net	Random Forest	XGboost
Overall Accuracy	78.5%	81.8%	83.0%	88.6%	**89.0%**
Cancellation accuracy	75.6%	69.5%	**85.1%**	82.4%	85.5%
Cancellation F1 – train	78.6%	96.5%	87.9%	92.1%	93.8%
Cancellation F1 – test	69.8%	71.5%	76.6%	82.6%	**83.6%**
Cancellation Recall	75.6%	69.5%	**85.1%**	82.4%	81.2%
Cancellation Precision	64.8%	73.5%	69.7%	**82.7%**	85.0%

Most balanced

Best performer

Berkeley
UNIVERSITY OF CALIFORNIA

Conclusion/Recommendations

- We were able to improve our performance by over 10% on Accuracy and 13% on F1 through XGBoost
- Lessons learned included the need to allocate sufficient time to do hyperparameter tuning and thoughtful application of feature engineering
- While we generally used F1 for key performance metrics in hyperparameter tuning, we used Accuracy to assess generalization and model fairness; in future iterations we could use F1 to assess generalization and fairness as well, to make the evaluation system even more consistent
- This is a PoC so future enhancements would include building a model pipeline allowing for dynamic pricing based on likely cancellations
- In addition to dynamic pricing we could also export out a daily "call list" list of reservations in next several weeks

Berkeley
UNIVERSITY OF CALIFORNIA

Citation

(1) Eleazar C. Sánchez a, et al. “Identifying Critical Hotel Cancellations Using Artificial Intelligence.” *Tourism Management Perspectives*, Elsevier, 13 July 2020, www.sciencedirect.com/science/article/abs/pii/S2211973620300854.

Berkeley
UNIVERSITY OF CALIFORNIA