Predicting Booking Cancellations

Kyle Reedy

24%

Global Hotel Booking Cancellation Rate

(https://experience-crm.fr/en/where-do-cancellations-come-from/#:~:text=Average%20cancellation%20rates&text=The%20average%20percentage%20of%20canceled,no%20room%20for%20nasty%20surprises.)

41.3%

2017 OTA average booking cancellation rate in Europe:

(https://hospitalitytech.com/global-cancellation-rate-hotel-reservations-reaches-40-average)

Executive Overview

1. Part I
    i. Problem and Motivation of Study
    ii. Stakeholders

2. Part II

    iii. Dataset
    iv. Cleaning and Wrangling
    v. EDA
    vi. ML Models

3. Part II

    vii. Conclusion
    viii. Recommendations

Motivation of Study

Understanding and predicting hotel booking cancellations is crucial for optimizing revenue and resource management in the hospitality industry. As someone who has worked for a hotel front desk in the past, this is an important project for the hospitality industry. By leveraging predictive modeling techniques, I aim to develop accurate and reliable algorithms that empower hotels to do the following:

Manage their inventory

Deliver exceptional guest experiences.

Enhance operational efficiency and stabilize revenue

Stakeholders and Clients

Within Hotel

- Hotel Managers
- Operational Staff
- Front Desk Staff

Corporate and Other

- Data Scientists and Analysts
- Revenue Management
- Marketing Team
- Investors and Shareholders

Setting of Data

The Algarve, Portugal's southernmost region, is known for its Atlantic beaches and golf resorts with whitewashed fishing villages on low cliffs overlooking sandy coves.

Data Set Overview

The dataset encompasses booking details for both a city hotel and a resort hotel in Portugal, spanning from 2015 to 2017.

Resort Hotel

City Hotel

Dataset Dictionary

1. Hotel: Type of hotel (Resort Hotel, City Hotel)
2. is_canceled: Reservation cancellation status (0 = not canceled, 1 = canceled)
3. lead_time: Number of days between booking and arrival
4. arrival_date_year: Year of arrival
5. arrival_date_month: Month of arrival
6. arrival_date_week_number: Week number of the year for arrival
7. arrival_date_day_of_month: Day of the month of arrival
8. stays_in_weekend_nights: Number of weekend nights (Saturday and Sunday) the guest stayed or booked
9. stays_in_week_nights: Number of week nights the guest stayed or booked
10. adults: Number of adults
11. children: Number of children
12. babies: Number of babies
13. meal: Type of meal booked (BB, FB, HB, SC, Undefined)
14. country: Country of origin of the guest
15. market_segment: Market segment designation
16. distribution_channel: Booking distribution channel
17. is_repeated_guest: If the guest is a repeat customer (0 = not repeated, 1 = repeated)
18. previous_cancellations: Number of previous bookings that were canceled by the customer
19. previous_bookings_not_canceled: Number of previous bookings that were not canceled by the customer
20. reserved_room_type: Type of reserved room
21. assigned_room_type: Type of assigned room
22. booking_changes: Number of changes made to the booking
23. deposit_type: Type of deposit made (No Deposit, Refundable, Non Refund)
24. agent: ID of the travel agent responsible for the booking
25. company: ID of the company responsible for the booking
26. days_in_waiting_list: Number of days the booking was in the waiting list
27. customer_type: Type of customer (Transient, Contract, Transient-Party, Group)
28. adr: Average Daily Rate
29. required_car_parking_spaces: Number of car parking spaces required
30. total_of_special_requests: Number of special requests made
31. reservation_status: Last reservation status (Check-Out, Canceled, No-Show)
32. reservation_status_date: Date of the last reservation status

(https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand)

The more important features from this dataset

119,390 Observations
32 Variables

Cleaning : Wrangling

- Duplicate features removed
- Dropping Null Values
  - Company: 94.3% missing
  - Agent: 13.7% missing

```
hotel_demand[hotel_demand.duplicated(keep=False)]
```

```
hotel_demand.drop(columns=['company'], inplace=True)
```

```
hotel_demand.drop(columns=['agent'], inplace=True)
```

```
hotel_demand.columns
```

```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type',
       'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

Cleaning : Wrangling (Continued)

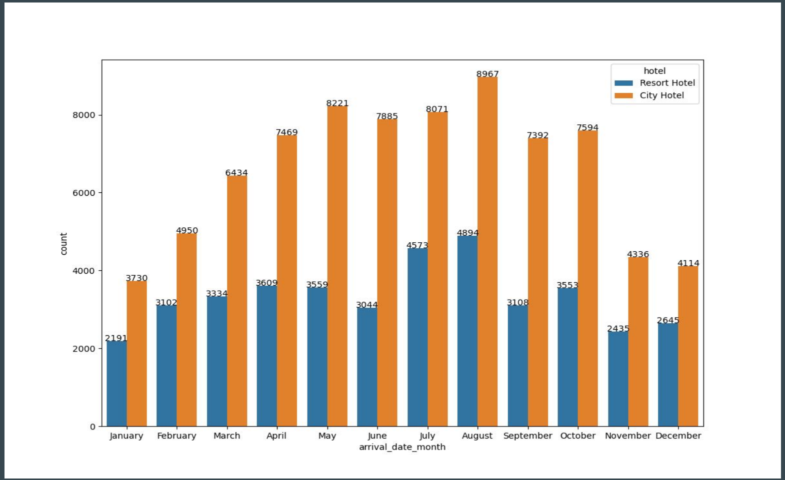
- Changing inconsistencies
  - ‘Country’ null values to 0
  - ‘Children’ and ‘Baby’ count 10 to Median Values
  - Any missing guest count to median values
  - Dropping rows where all adults, children and baby count are 0
    - 180 rows dropped

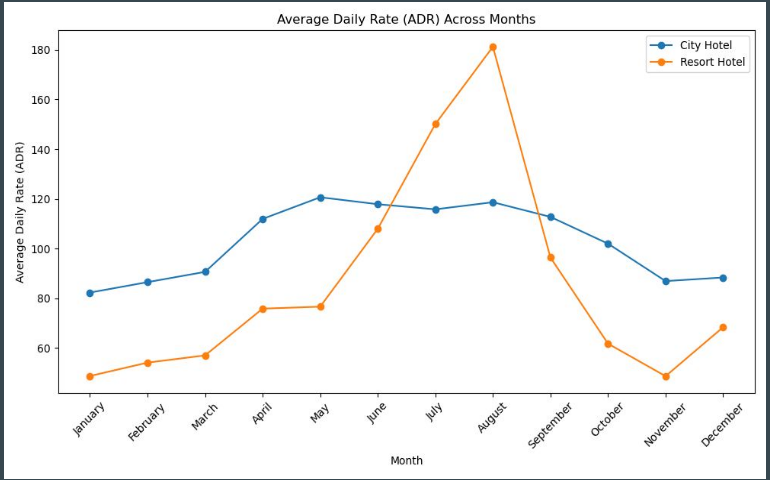
```
filter = (hotel_demand['adults'] == 0) & (hotel_demand['children'] == 0) & (hotel_demand['babies'] == 0)
hotel_demand[filter]
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weeken
2224	Resort Hotel	0	1	2015	October	41	6	
2409	Resort Hotel	0	0	2015	October	42	12	
3181	Resort Hotel	0	36	2015	November	47	20	
3684	Resort Hotel	0	165	2015	December	53	30	
3708	Resort Hotel	0	165	2015	December	53	30	
...	...	...	...	...	...	...	...	
115029	City Hotel	0	107	2017	June	26	27	
115091	City Hotel	0	1	2017	June	26	30	
116251	City Hotel	0	44	2017	July	28	15	
116534	City Hotel	0	2	2017	July	28	15	
117087	City Hotel	0	170	2017	July	30	27	

180 rows × 30 columns

Data Visualization





Busiest - August

City: 8967

Resort: 4894

Least Busiest - January

City: 3730

Resort: 2191

Highest Rate - August

City: 118.7

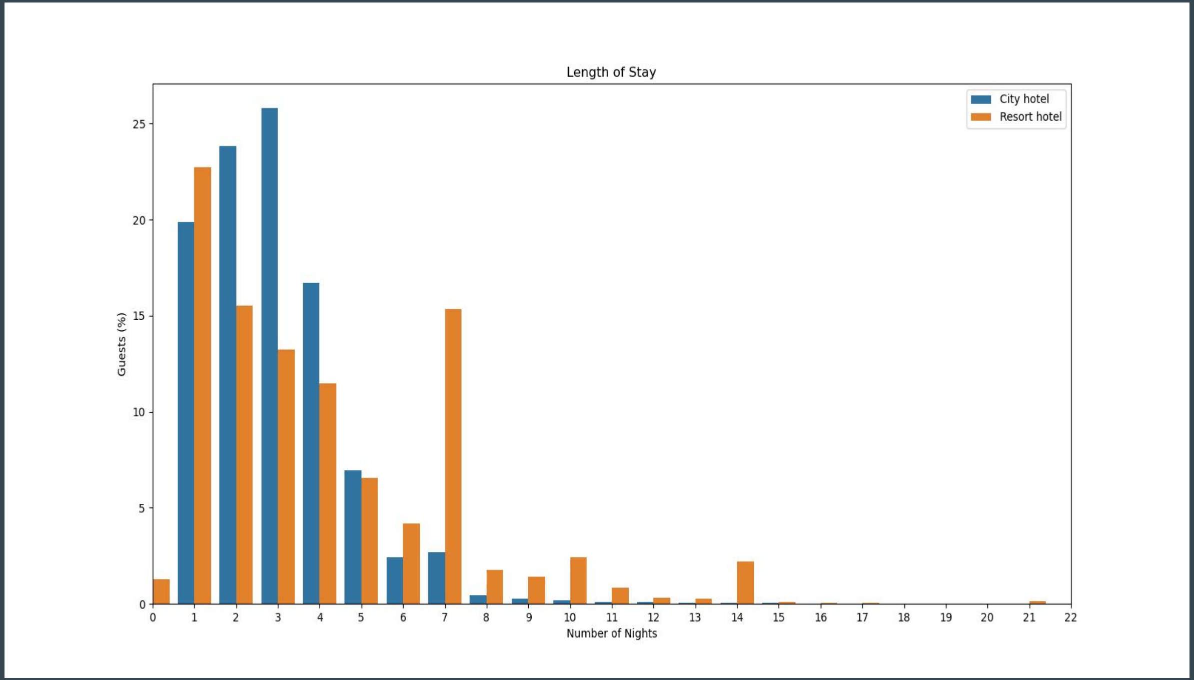
Resort: 181.2

Lowest Rate - January

City: 48.7

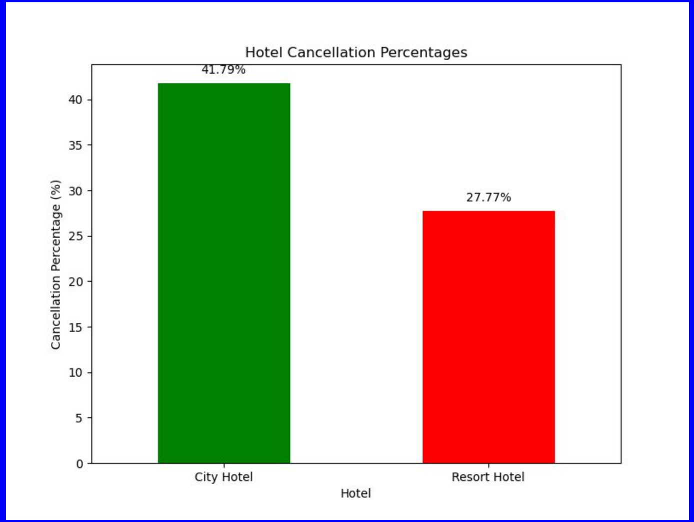
Resort: 82.3

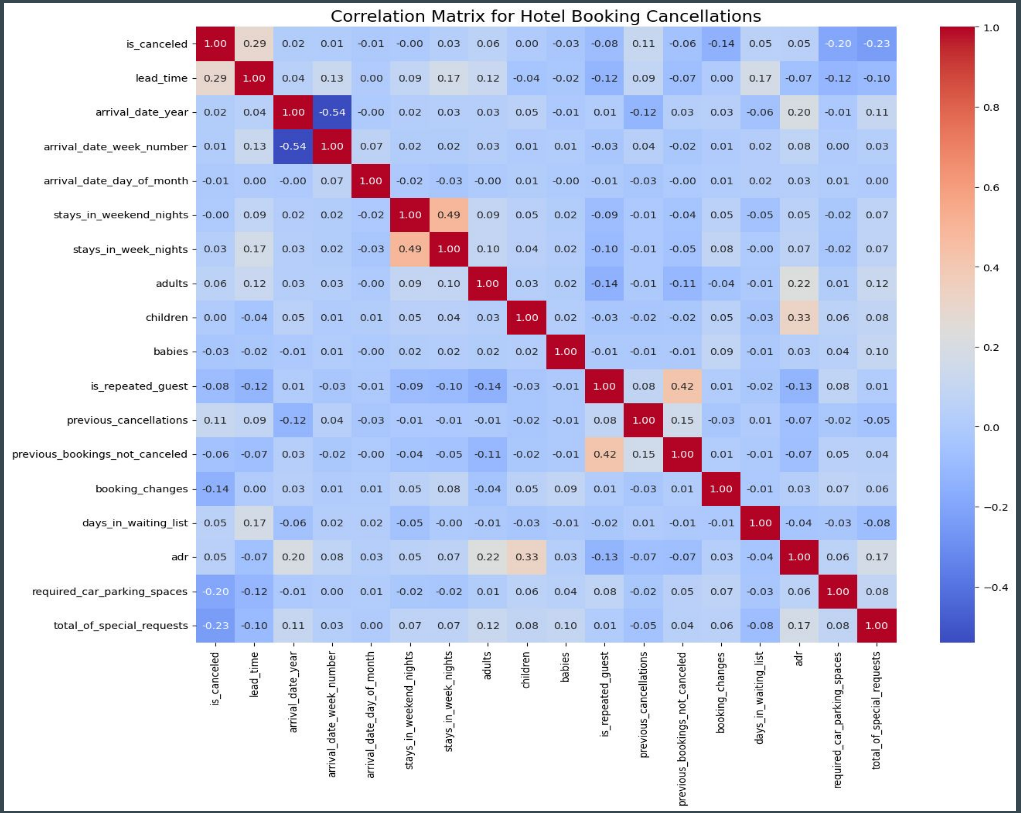
EDA



Most customers tend to stay between 1-7 nights. A large set of resort customer stay for one week and a another portion stay for 14 days.

EDA





Dataset target variable comparison:

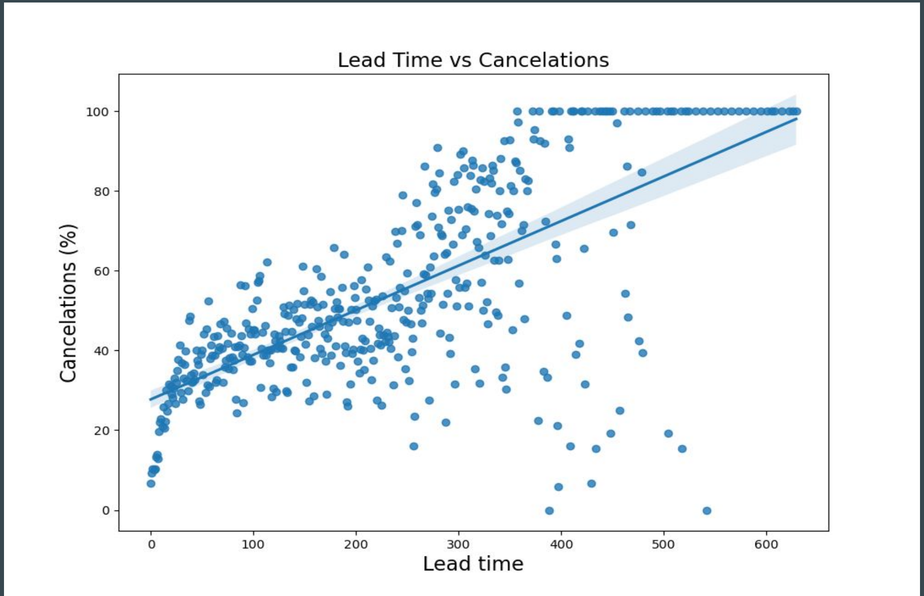
Resort is_canceled Vs City is_canceled

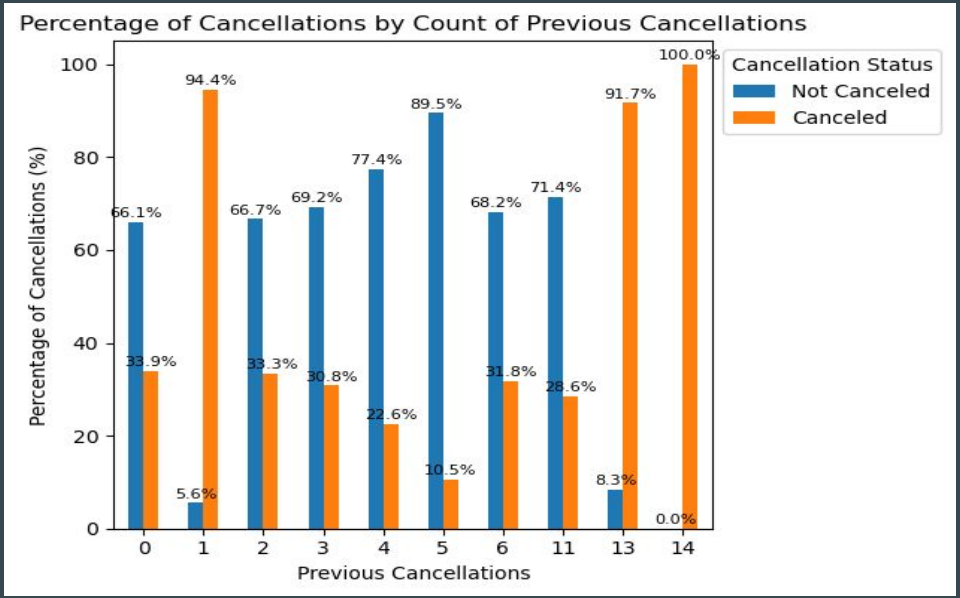
Correlation Matrix of Features with Target Variable: is_canceled

Relevant correlations:

lead_time, total_of_special_requests, required_car_parking_spaces

EDA





Bookings made a few days before their arrival date are rarely canceled. Bookings made a year in advance have a higher probability of cancellations.

94.4% cancellation for those who have canceled in the past. Decreases significantly to 33.3% after two previous cancellations.

Preprocessing

```
category_cols = [col for col in df2.columns if df2[col].dtype == 'O']
category_cols
```

```
['hotel',
 'arrival_date_month',
 'meal',
 'country',
 'market_segment',
 'distribution_channel',
 'reserved_room_type',
 'assigned_room_type',
 'deposit_type',
 'customer_type',
 'reservation_status',
 'reservation_status_date']
```

-Outliers taken care of

-Categorical variables switched to numerical

    -Total of 12 changed

-3 variables dropped

    -reservation_status

    -country

    -assigned_room_type

Machine Learning: Models Compared

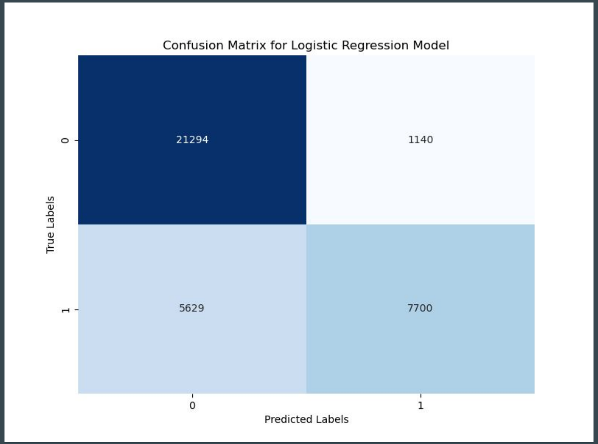
	Model	Scores
1	XGBoost	0.982831
2	Random Forests	0.957442
3	Decision Trees	0.950004
4	Ada Boost	0.949333
5	Gradient Boost	0.917513
6	KNeighbors Classifier	0.865140
7	Logistic Regression	0.804183

The XGboost model had the highest accuracy score with 0.98 while the Logistic Regression model had the lowest with 0.80

To further analyze these accuracy scores in a visual, we will take a look at the confusion matrix for each model:

Modeling - Confusion Matrix

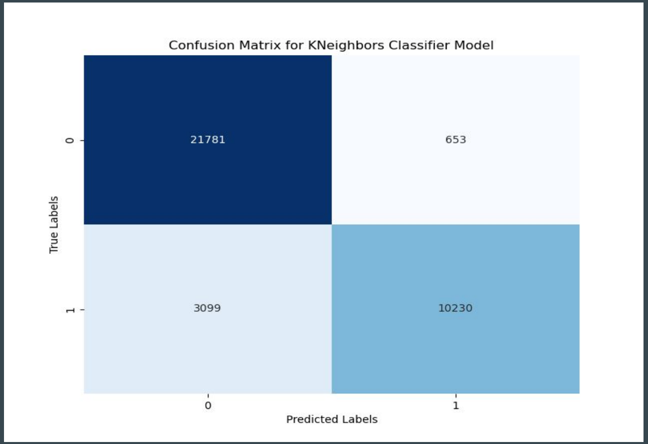
Logistic Regression



Correct Predictions:
Will cancel -> 21,294
Will not cancel -> 7,700

Incorrect Predictions:
Will cancel -> 5,629
Will not cancel -> 1,140

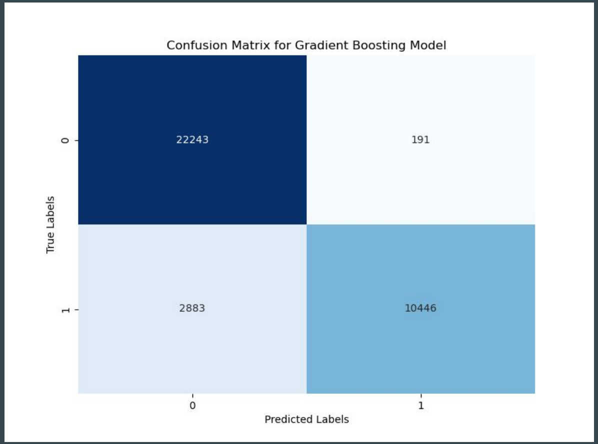
KN-Neighbors



Correct Predictions:
Will cancel -> 21,781
Will not cancel -> 10,230

Incorrect Predictions:
Will cancel -> 3,099
Will not cancel -> 653

Gradient Boosting

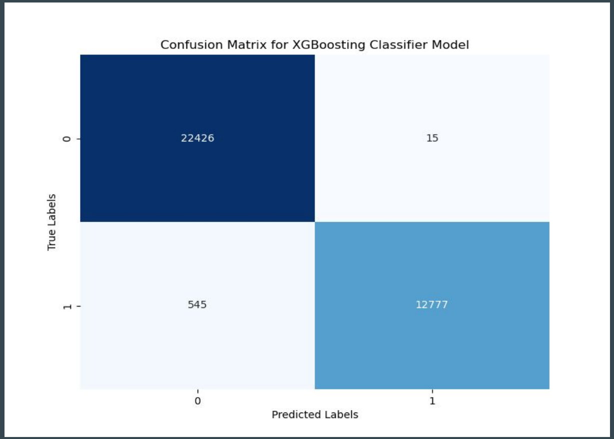


Correct Predictions:
Will cancel -> 22,243
Will not cancel -> 10,446

Incorrect Predictions:
Will cancel -> 2,883
Will not cancel -> 191

Modeling - Confusion Matrix

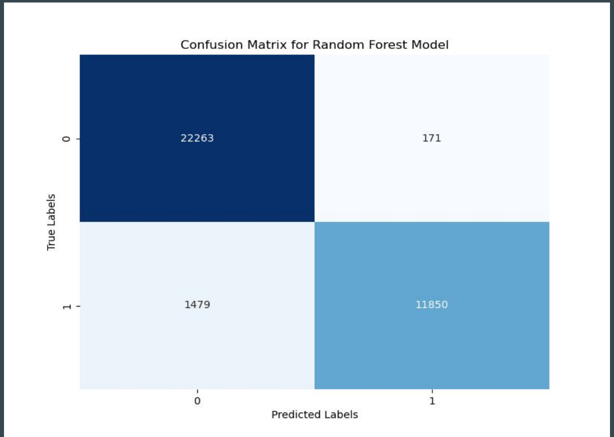
XGBoosting



Correct Predictions:
- Will cancel -> 22,426
- Will not cancel -> 12,777

Incorrect Predictions:
- Will cancel -> 545
- Will not cancel -> 15

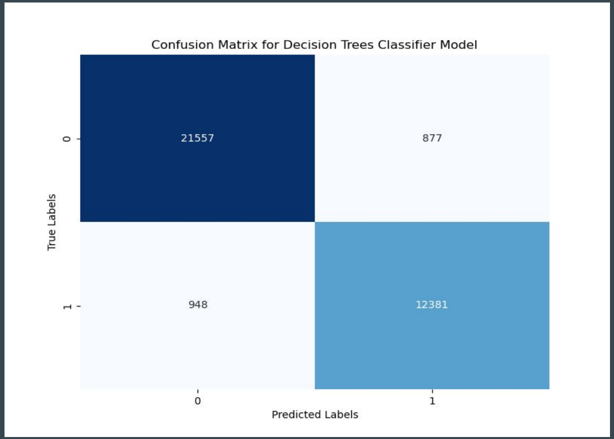
Random Forests



Correct Predictions:
- Will cancel -> 22,263
- Will not cancel -> 11,850

Incorrect Predictions:
- Will cancel -> 1,479
- Will not cancel -> 171

Decision Trees



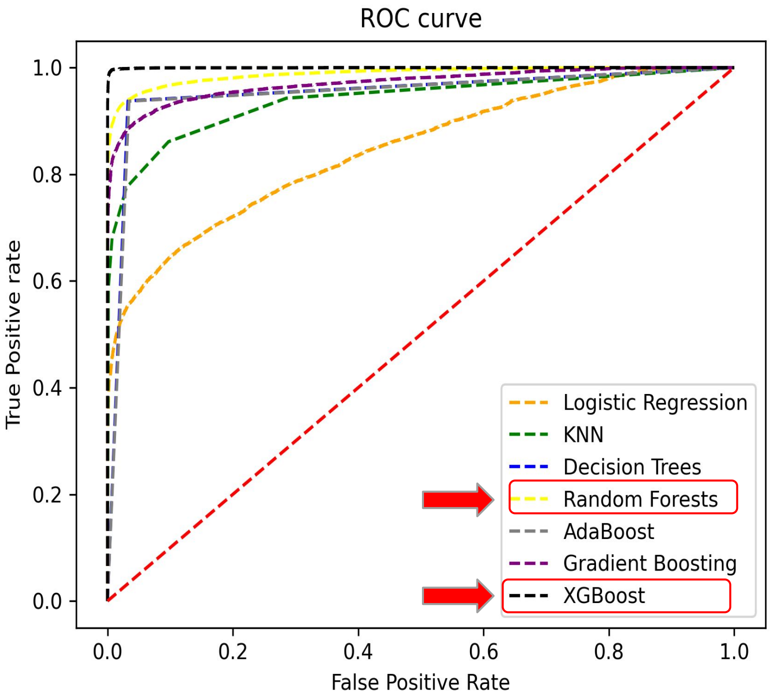
Correct Predictions:
- Will cancel -> 21,557
- Will not cancel -> 12,381

Incorrect Predictions:
- Will cancel -> 948
- Will not cancel -> 877

Improvements

Focusing on High-Performing Models

- Utilize **XGBoost** and **Random Forests** for prediction tasks due to their higher accuracy scores compared to other models. These models are more reliable in identifying potential cancellations and can assist in proactive management strategies.



Recommendations

1. Targeted Marketing and Accomodations:

Tailor marketing strategies based on customer characteristics and behavior.

- Target repeat customers with personalized offers or incentives and special deals for transient guests to encourage loyalty and retaining guests
- Additionally, offering extra services such as tours and rentals, and planning accommodations for guests with special requests will lower cancellation rates

2. Deposit Policy Optimization:

- Adjust deposit policies especially for refundable deposits, and offer incentives for non-refundable bookings to give greater customer satisfaction.

3. Lead Time Management:

- Encourage guests to book well in advance by offering early booking discounts or special promotions.
- Imply stricter cancellation policies as arrival date gets closer.

Random Forest Feature Importance