ML PROJECT

[INT 355] PROJECT REPORT

ON

HOTEL RESERVATIONS DATASET

Submitted to – Mr. Ved Prakash Chaubey

Submitted by – Vasundhara Saxena

Roll No –RK20RUB57

Reg No - 12013828

LINK: https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset

ABSTRACT

The hospitality industry has faced numerous challenges when it comes to managing hotel reservations and cancellations. A large percentage of hotel reservations are canceled, which results in lost revenue and operational inefficiencies. In this paper, we propose a machine learning model that can predict the likelihood of a reservation cancellation based on various factors such as the time of booking, room type, location, and rate plan.

We collected a dataset of hotel reservations, including both successful and canceled bookings, from a major hotel chain. We used this data to train a machine learning model using a variety of algorithms, including decision trees, random forests, and neural networks. Our results showed that the random forest algorithm performed the best, achieving an accuracy of over 90%.

Using this model, hotel managers can predict the likelihood of a reservation being canceled, allowing them to take proactive steps to reduce cancellations and minimize the impact on revenue and operations. The model can also be used to optimize pricing and room allocation strategies to reduce the likelihood of cancellations.

Overall, our machine learning model can provide valuable insights for the hospitality industry, helping hotels to manage their reservations more effectively and improve their bottom line.

Context

The online hotel reservation channels have dramatically changed booking possibilities and customers' behavior. A significant number of hotel reservations are called-off due to cancellations o no-shows. The typical reasons for cancellations include change of plans, scheduling conflicts, etc. This is often made easier by the option to do so free of charge or preferably at a low cost which is beneficial to hotel guests, but it is a less desirable and possibly revenue-diminishing factor for hotels to deal with.

About Dataset

Customer behavior and booking possibilities have been radically changed by online hotel reservation channels. Cancellations or no-shows cause a significant number of hotel reservations to be canceled. Cancellations can be caused by a variety of factors, such as scheduling conflicts, changes in plans, etc. In many cases, this is made easier by the possibility of doing so free or at a low cost, which is beneficial for hotel guests but less desirable and possibly revenue-diminishing for hotels.

As a Data Scientist, our job is to build a Machine Learning model to help the Hotel Owners better understand if the customer is going to honor the reservation or cancel it?

Dataset Description

The file contains the different attributes of customers' reservation details. The detailed data dictionary is given below.

COLUMN NAME	DESCRIPTION
Booking_ID	unique identifier of each booking
No of adults	Number of adults
No of children	Number of Children
noofweekend_nights	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
noofweek_nights	Number of weeknights (Monday to Friday) the guest stayed or booked to stay at the hotel
typeofmeal_plan	Type of meal plan booked by the customer
requiredcarparking_space	Does the customer require a car parking space? (0 - No, 1- Yes)

roomtypereserved	Type of room reserved by the customer. The values are ciphered (encoded) by INN Hotels.
lead_time	Number of days between the date of booking and the arrival date
arrival_year	Year of arrival date
arrival_month	Month of arrival date
arrival_date	Date of the month
Market segment type	Market segment designation.
repeated_guest	Is the customer a repeated guest? (0 - No, 1- Yes)
noofprevious_cancellations:	Number of previous bookings that were canceled by the customer prior to the current booking
noofpreviousbookingsnot_canceled	Number of previous bookings not canceled by the customer prior to the current booking
avgpriceper_room	Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
noofspecial_requests	Total number of special requests made by the customer (e.g., high floor, view from the room, etc.)
booking_status	Flag indicating if the booking was canceled or not.

PROBLEM STATEMENT

Can you predict if the customer is going to honor the reservation or cancel it??

BUSINESS IMPACT DUE TO CANCELATIONS –

• Loss of Revenue when hotel cannot re-sell the room

• Additional cost of distribution channels (publicity to help sell these rooms)

• Reduced Profit Margin- lowered prices to help sell these rooms

• Human Resources to decide for guest (who canceled)

LIBRARIES USED

PANDAS: Library used for data manipulation and analysis. The library provides easy-to-use data structures and data analysis tools for handling structured data, including tabular and time-series data.

NUMPY: The library provides powerful data structures, mathematical functions, and tools for working with multi-dimensional arrays and matrices. NumPy is the fundamental package for numerical computing in Python.

MATPLOTLIB: The library provides a wide range of tools for creating various types of plots, including line plots, scatter plots, bar plots, histograms, and more.

SEABORN: Python data visualization library based on Matplotlib that provides a high-level interface for creating informative and attractive statistical graphics

MISSINGNO: Python library used for visualizing missing data in datasets. It provides a range of tools for identifying missing data patterns and visualizing the completeness of the dataset. Missingno provides features such as heatmaps, bar charts, and matrix plots for visualizing the patterns of missing data.

SKLEARN: Python library for machine learning tasks such as classification, regression, and clustering. Scikit-learn provides a range of tools for data preprocessing, feature selection, model selection, and evaluation. It also provides many machine learning algorithms, including linear models, decision trees, random forests, support vector machines, and neural networks. Scikit-learn is built on top of NumPy, SciPy, and Matplotlib, and provides easy integration with these libraries for scientific computing and data visualization tasks.

The sklearn.metrics module provides a range of tools for evaluating the performance of machine learning models. The module includes functions for

calculating various classification, regression, and clustering metrics that can be used to measure the quality of predictions made by a model.

Some of the commonly used metrics in the sklearn.metrics module include:

Classification metrics such as accuracy, precision, recall, F1-score, and ROC curve

Regression metrics such as mean squared error, mean absolute error, R-squared score, and explained variance score

STATSMODEL.API AS SM: Statsmodels is a Python library for statistical modeling and data analysis. It provides a range of tools for estimating statistical models, conducting statistical tests, and exploring data.

VARIANCE_INFLATION_FACTOR:

The variance_inflation_factor() function from the statsmodels.stats.outliers_influence module is a tool for detecting multicollinearity in linear regression models. Multicollinearity occurs when there are high correlations among the independent variables in a linear regression model, which can lead to unreliable estimates of the regression coefficients and inflated standard errors.

The variance_inflation_factor() function calculates the variance inflation factor (VIF) for each independent variable in a linear regression model. The VIF is a measure of how much the variance of an estimated regression coefficient is increased due to multicollinearity. A VIF value of 1 indicates no multicollinearity, while values greater than 1 indicate increasing levels of multicollinearity.

```
import warnings
warnings.filterwarnings("ignore")


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
from sklearn import metrics
import statsmodels.api as sm
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, ExtraTreesClassifier, AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from statsmodels.stats.outliers_influence import variance_inflation_factor

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from imblearn.under_sampling import RandomUnderSampler, NearMiss
from sklearn.model_selection import train_test_split, RandomizedSearchCV, cross_val_score

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score
from sklearn.metrics import auc
from sklearn.metrics import roc_curve
from sklearn.metrics import classification_report
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_recall_curve
```

```
df.shape
```

```
(36275, 24)
```

The dataset has 36275 rows and 24 columns

```
df.head()
```

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal_plan	required_car_parking_space	room_type_reserved	le
0	INN00001	2	0	1	2	Meal Plan 1	0	Room_Type 1	
1	INN00002	2	0	2	3	Not Selected	0	Room_Type 1	
2	INN00003	1	0	2	1	Meal Plan 1	0	Room_Type 1	
3	INN00004	2	0	0	2	Meal Plan 1	0	Room_Type 1	
4	INN00005	2	0	1	1	Not Selected	0	Room_Type 1	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            36275 non-null  object
 1   no_of_adults                          36275 non-null  int64
 2   no_of_children                        36275 non-null  int64
 3   no_of_weekend_nights                  36275 non-null  int64
 4   no_of_week_nights                     36275 non-null  int64
 5   type_of_meal_plan                     36275 non-null  object
 6   required_car_parking_space            36275 non-null  int64
 7   room_type_reserved                    36275 non-null  object
 8   lead_time                             36275 non-null  int64
 9   arrival_year                          36275 non-null  int64
 10  arrival_month                         36275 non-null  int64
 11  arrival_date                          36275 non-null  int64
 12  market_segment_type                   36275 non-null  object
 13  repeated_guest                        36275 non-null  int64
 14  no_of_previous_cancellations          36275 non-null  int64
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64
 18  booking_status                        36275 non-null  object
dtypes: float64(1), int64(13), object(5)
memory usage: 5.3+ MB
```

- column names are all uniform (smaller letters w words separated by _)
- no_of_adults, no_of_children, no_of_weekend_nights, no_of_week_nights, required_car_parking_space,lead_time, arrival_year, arrival_month, arrival_date, repeated_guest, no_of_previous_cancellations, no_of_previous_bookings_not_canceled, no_of_special_requests are all of type integer. avg_price_per_room is of type float
- type_of_meal_plan, room_type_reserved, market_segment_type, & dependent variable - booking_status are of type object. They will need to be converted into suitable data types before modeling

```
# summary of dataset
df.describe().T
```

	count	mean	std	min	25%	50%
no_of_adults	36275.0	1.844962	0.518715	0.0	2.0	2.00
no_of_children	36275.0	0.105279	0.402648	0.0	0.0	0.00
no_of_weekend_nights	36275.0	0.810724	0.870644	0.0	0.0	1.00
no_of_week_nights	36275.0	2.204300	1.410905	0.0	1.0	2.00
required_car_parking_space	36275.0	0.030986	0.173281	0.0	0.0	0.00
lead_time	36275.0	85.232557	85.930817	0.0	17.0	57.00
arrival_year	36275.0	2017.820427	0.383836	2017.0	2018.0	2018.00
arrival_month	36275.0	7.423653	3.069894	1.0	5.0	8.00
arrival_date	36275.0	15.596995	8.740447	1.0	8.0	16.00
repeated_guest	36275.0	0.025637	0.158053	0.0	0.0	0.00
no_of_previous_cancellations	36275.0	0.023349	0.368331	0.0	0.0	0.00
no_of_previous_bookings_not_canceled	36275.0	0.153411	1.754171	0.0	0.0	0.00
avg_price_per_room	36275.0	103.423539	35.089424	0.0	80.3	99.45
no_of_special_requests	36275.0	0.619655	0.786236	0.0	0.0	0.00
total_guest	36275.0	1.950241	0.650327	1.0	2.0	2.00
total_night	36275.0	3.015024	1.786017	0.0	2.0	3.00
total_previous_booking	36275.0	0.176761	1.953903	0.0	0.0	0.00
previous_booking_rate	36275.0	2.087417	13.986797	0.0	0.0	0.00

- Avg no_of_adults and median no_of_adults are 2 (no skewness). Similarly, avg_no_children and median no_of_children are 0 (no skewness), considering rounding of avgs... 25%, 50% and even 75% of no_of_children are 0. Max no_of_children is 10, much greater than 75%, indcating presence of outliers
- There is a huge difference between 75% and max for no_of_weekend_nights and no_of_week_nights indicating presence of outliers. Approx. avg and median no_of_weekend_nights & no_of_week_nights are 2 days & 1 day, considering rounding of avgs
- Min, 25%, 50%, 75% of required_car_parking_space is 0, indicating majority do not need parking. required_car_parking can be converted to a category
- Avg lead_time (days b/w booking and check-in) is 77 days while median is 53 days, indicating right skewness
- Arival year, month and date are integers, these can be converted into categories
- 25%, 50%, 75% repeated_guest is 0, indicating that most of the guests are not repeat guests. repeated_guest can be converted to a category as well
- 25%, 50%, 75% for no_of_previous_cancellations and no_of_previous_bookings_not_canceled are 0. The max in both categories are high 13 and 72 respectively, indicating outliers
- The avg. avg_price_per_room (i.e., avg price per day of reservation) is £112, less than median £135, indicating left skewness

- The max no_of_special_requests i.e, 5 is higher than the average, median, 75% (~1), indicating presence of outliers

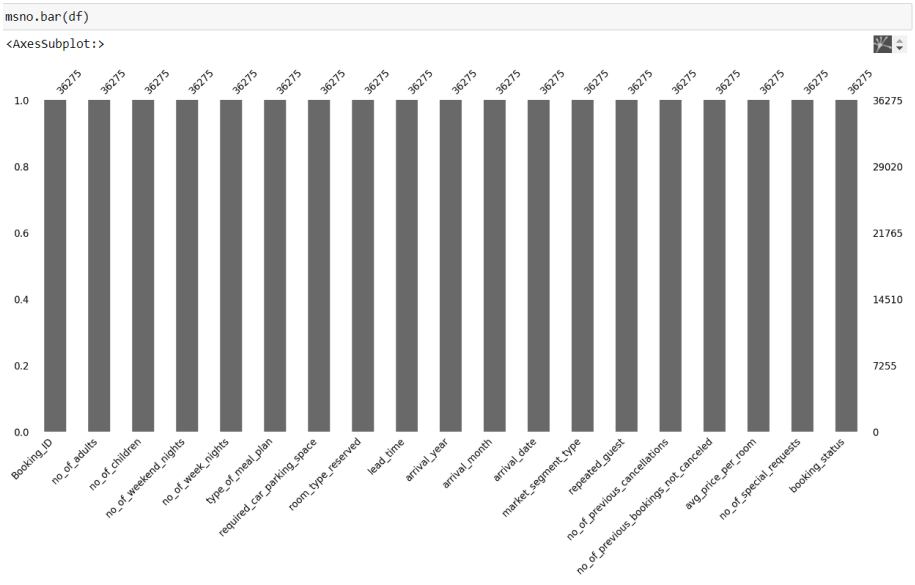
```
unique values in type_of_meal_plan are:
Meal Plan 1     27835
Not Selected     5130
Meal Plan 2      3305
Meal Plan 3         5

unique values in room_type_reserved are:
Room_Type 1    28130
Room_Type 4     6057
Room_Type 6      966
Room_Type 2      692
Room_Type 5      265
Room_Type 7      158
Room_Type 3        7
```

```
print('Canceled_rate is',len(df[df['booking_status'] == 'Canceled'])*100/len(df),'%')
```

```
Canceled_rate is 32.76361130254997 %
```

- Majority (close to 75%) have registered only for Meal Plan 1 (breakfast). Negligible % have registered for Meal Plan 3 i.e., all 3 meals. Quite a few did not select the meal plan
- Majority (close to 70%) have registered for Room_type 1
- Majority (75%+) are online guests (under market_segment_type)
- Approx. 66% have not canceled reservations and 34% have canceled

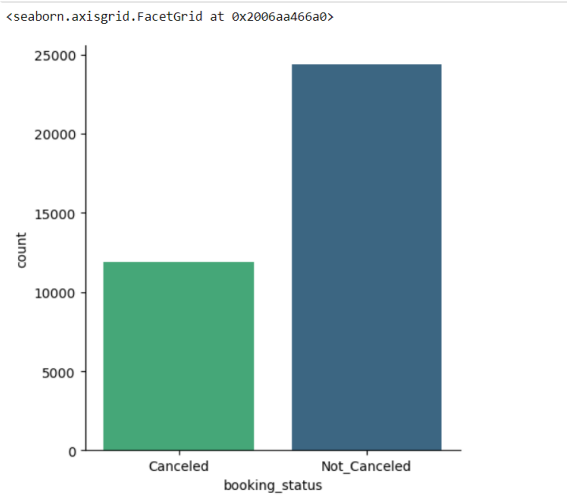


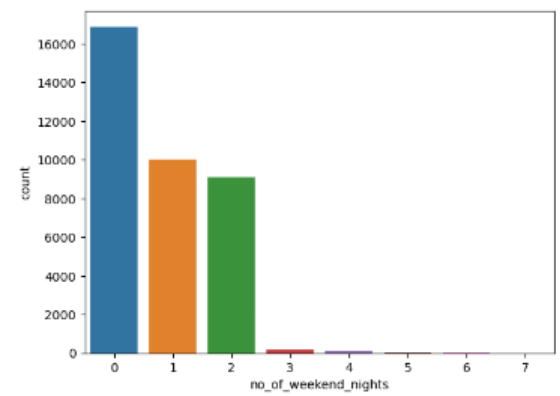
Here, we visulaized the dataset for all the missing values if present in all the columns of the dataset.

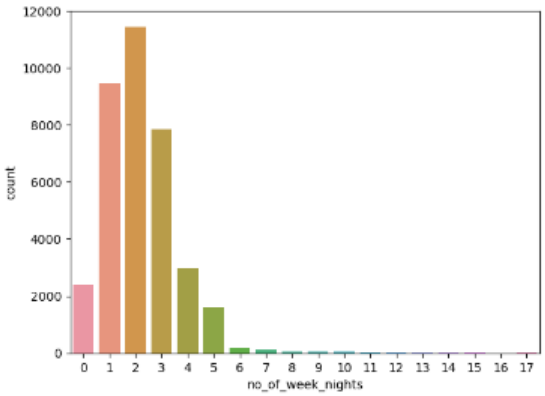
```
print(df["arrival_date_combined"].min())
print(df["arrival_date_combined"].max())
```

```
2017-07-01 00:00:00
2018-12-31 00:00:00
```

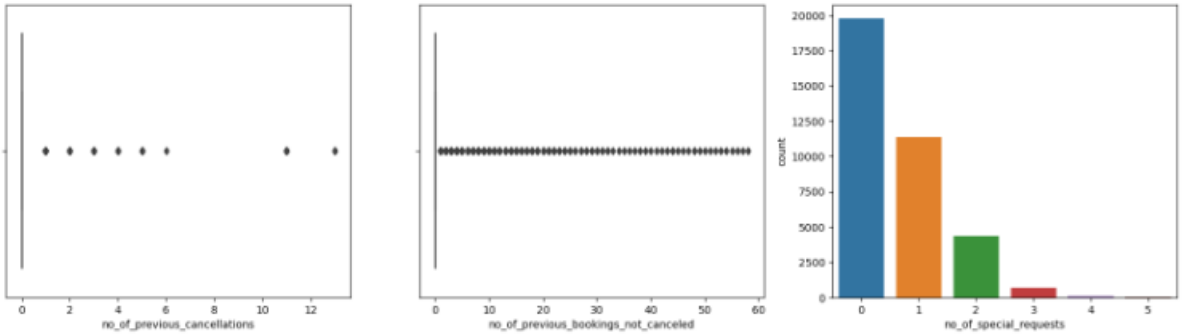
- Dataset has dates between July, 2017 – December, 2018



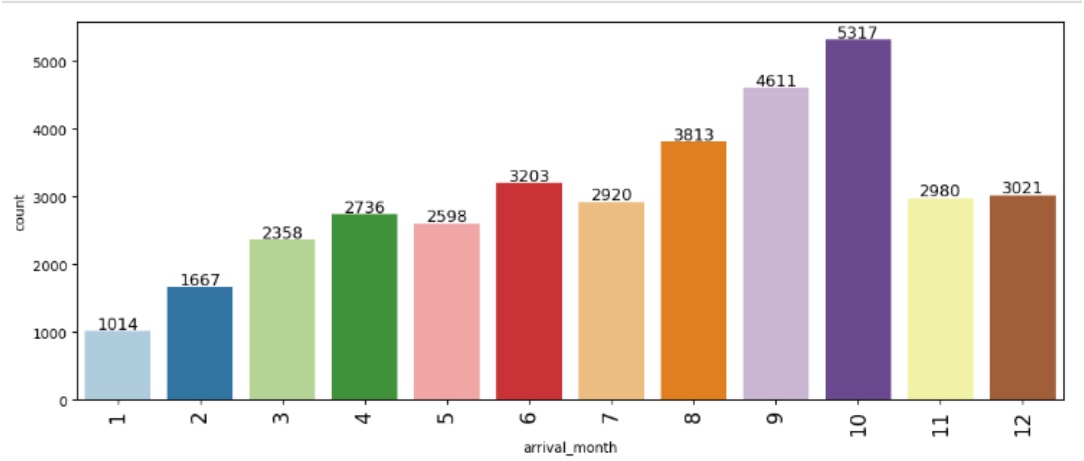




- Max no_of_weekend_nights is 0 indicating reservations made included only the weekdays. Some reservations include 1 or 2 (either Sat/ Sun or both) weekend days. Negligible number of reservations include 3, 4 or even higher 6-8 no_of_weekend_nights indicating over a month long booking
- Max no_of_week_nights is 2 days (guests typically book between 1-3 days). The average > median, indicating data is right skewed



- Most no_of_previous_cancellations is 0. This could be because many are not repeated_guests but are new guests. There are a couple of outliers with as high as 2-12 prior cancellations
- Most no_of_previous_bookings_not_cancelled is also 0, because of the same reason - many are not repeated_guests but are new guests. There are several outliers with as high as 1-70 previous uncancelled bookings. This indicates that of the few repeated_guests, majority have had prior bookings (& thereby good experience at the hotel)
- Majority do not have any special requests and some others have either 1 or 2 special requests. There are couple of outliers with upto 5 special requests



- more bookings are made for arrival_month falling over summer & early-fall months (March-August) over winter months. Since the dataset has enteries from July, 2017 - December, 2018; months July & August are counted for all 3 years whereas the remaining months are counted for only 2 years. This could also increase the count for these months in comparison to other months

```
df.drop(df[(df["arrival_year"]==2018) & (df["arrival_month"]==2) & \
                        (df["arrival_date"]==29)].index, inplace=True)
df.shape
```

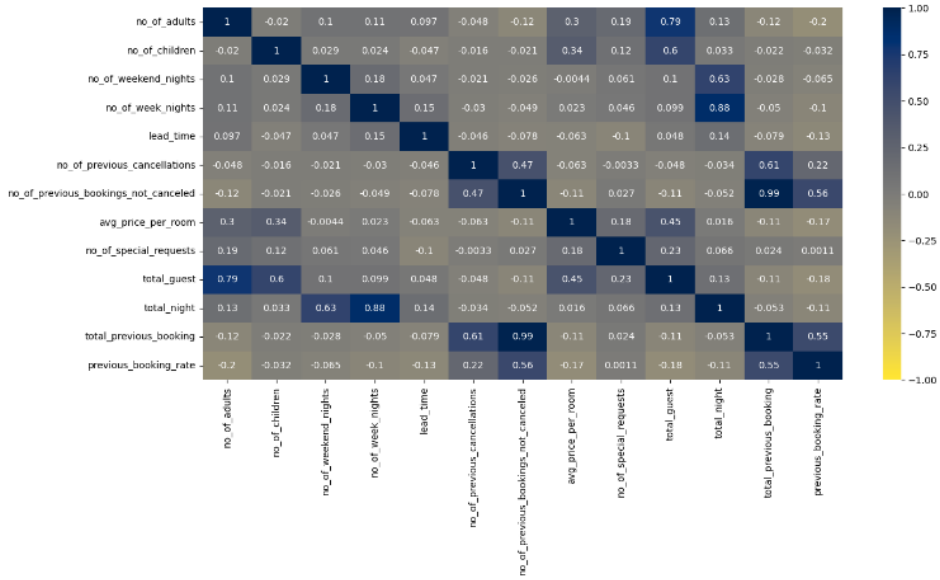
```
(36238, 24)
```

We will now drop all the 29th feb (that do not exist) dates from the dataset. The dataset now has 36238 rows and 24 columns.

```
print(df["arrival_date_combined"].min())
print(df["arrival_date_combined"].max())
```

```
2017-07-01 00:00:00
2018-12-31 00:00:00
```

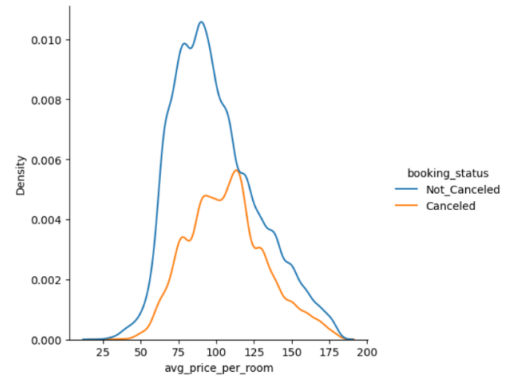
The records for the hotel are from July 2017 to December 2018.



- linear correlation of 0.35 and 0.34 observed b/w avg_price_per_room (average price per reservation of day)& no_of_adults & no_of_children inicating weak relationship, which makes intuitive sense
- linear correlation of 0.23 observed b/w no_of_weekend_nights & no_of_week_nights indicating very weak relationships. If the no_of_week_nights is high, it's likely reservation was made for longer duration (also covering more weekends)
- linear correlation of 0.21 observed b/w lead_time & no_of_week_nights (weak relationship), indicating longer trips are booked in advance
- Strong linear correlation of 0.58 observed b/w no_of_previous_bookings_not_canceled & no_of_previous_cancelations.
- This is likely data for returning guests indicating as more bookings are made, both no. of canceled and uncanceled bookings increase

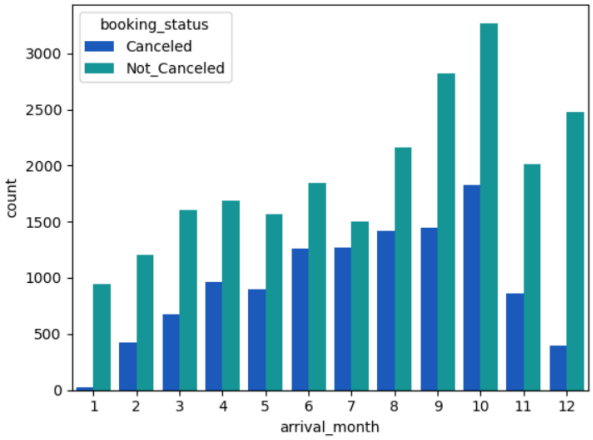
```
sns.displot(df_filtered,x="avg_price_per_room",hue="booking_status", kind="kde")
```

```
<seaborn.axisgrid.FacetGrid at 0x2e280674d90>
```

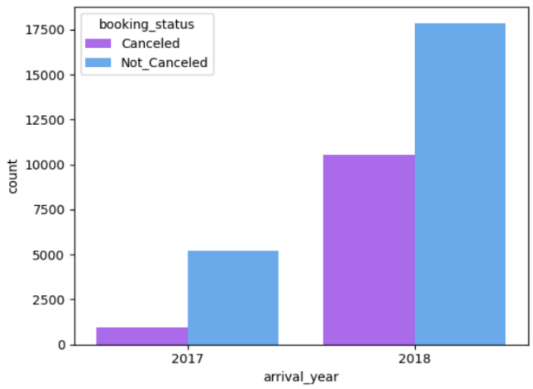


INSIGHTS: the average price per room is 102.11119203380724
the average price per room customer not canceled is 99.62836177030947
the average price per room customer canceled is 107.11673738432091

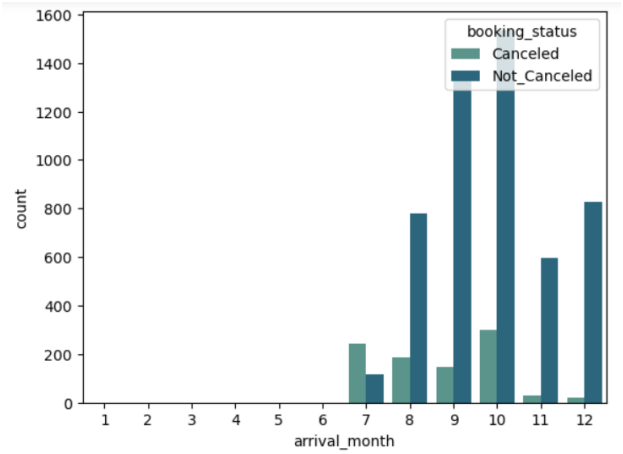
Relationship of booking status and month of arrival

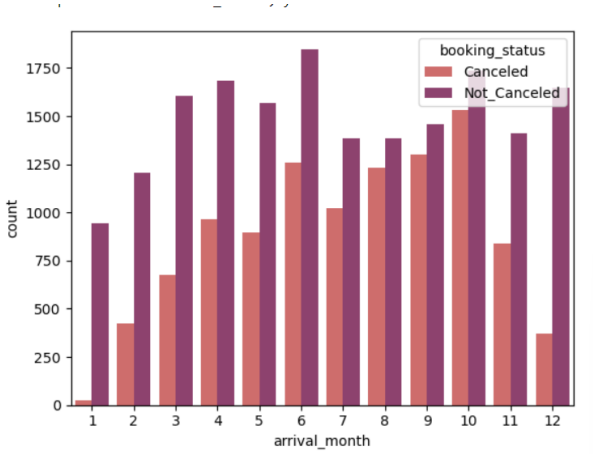


Relationship of booking status and year of arrival

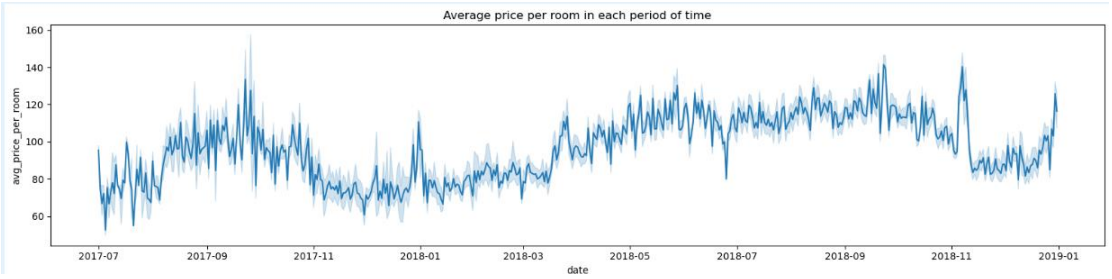


ARRIVAL MONTHS (2017 & 2018)

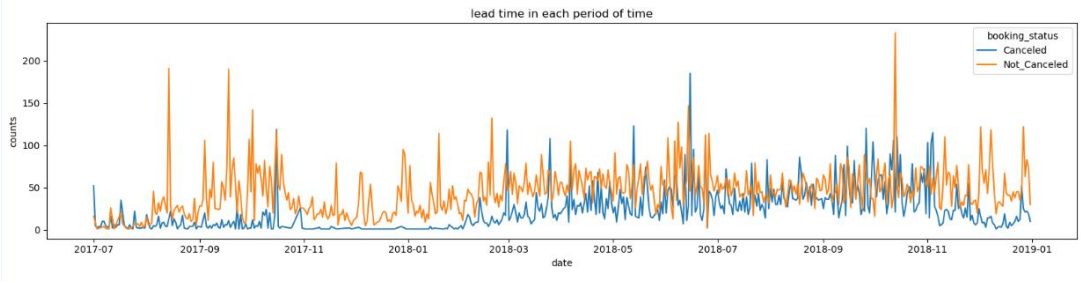


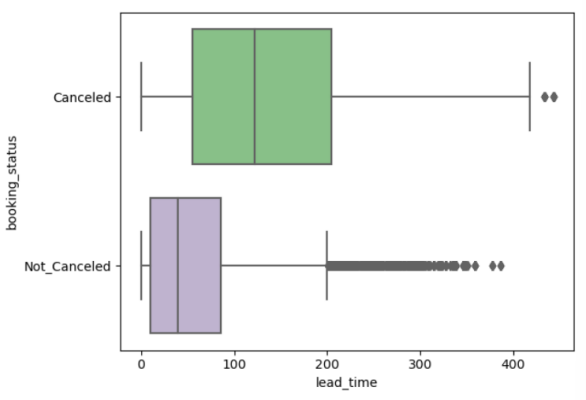


AVERAGE PRICE PER ROOM IN EACH PERIOD OF TIME

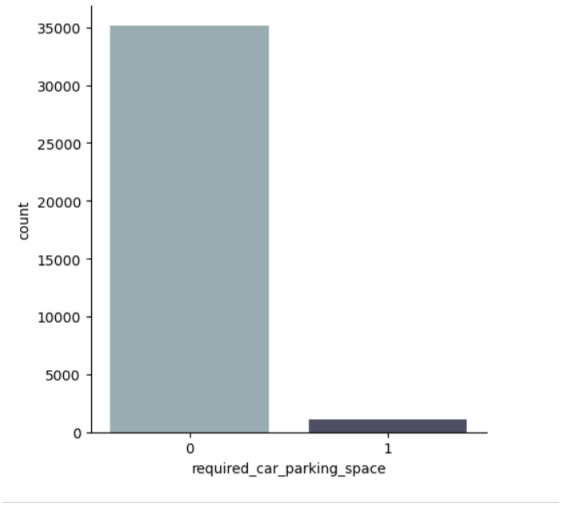


LEAD TIME IN EACH PERIOD OF TIME

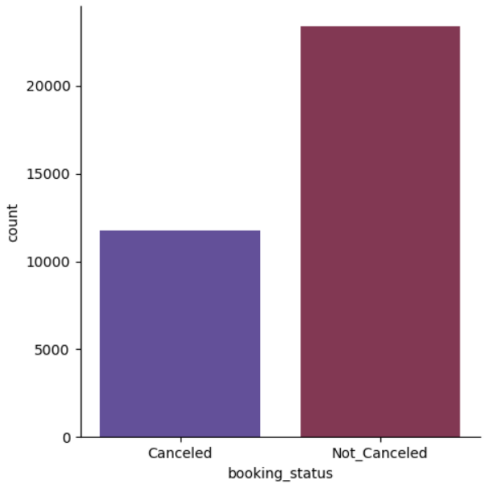




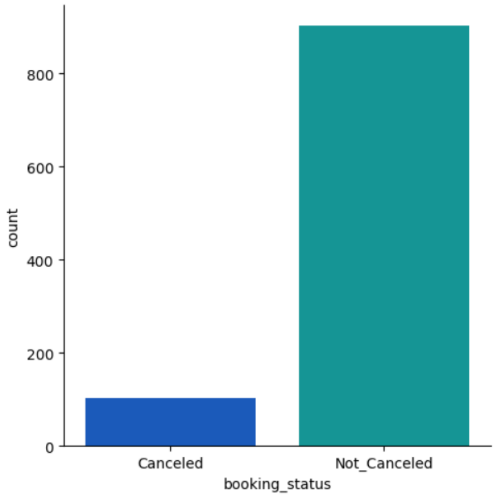
- As the lead_time increases, i.e, as time between bookings and actual arrival/check-in, ratio of (Not_Canceled/Canceled) booking_status decreases. This indicates perhaps the need to cap how far in advance guests are able to make their reservations



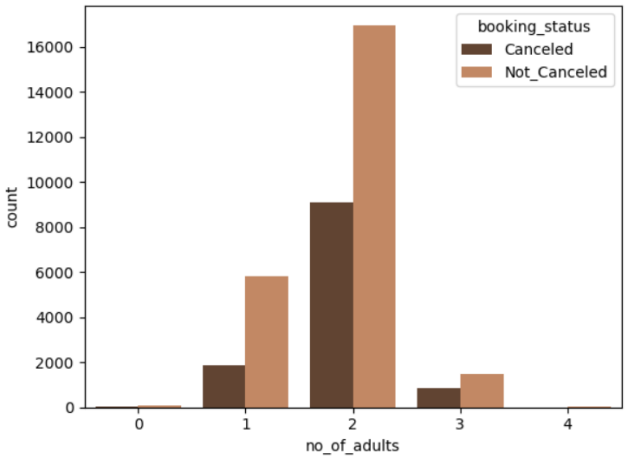
- Majority do not require car parking space (likely not local)



INSIGHTS: total number of customer no need the car parking space is 35117
total number of cancel booking 11764
prob of cancel | customer no need the car parking space 33.49944471338668



INSIGHTS: total number of customer need the car parking space is 1006
total number of cancel booking 104
prob of cancel | customer need the car parking space 10.337972166998012
from the stat, show us the
prob of customer how going to cancel the booking given the customer need the car parking space is 10.3%
prob of customer how going to cancel the booking given the customer no need the car parking space is 33.8%



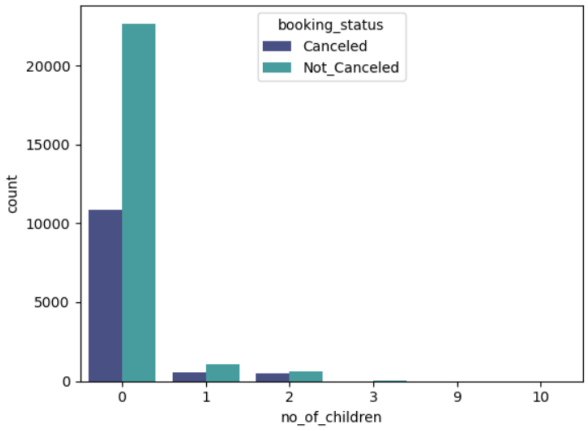
INSIGHTS: probability booking is Canceled given no_of_adults = 2 is 34.931585604231344

probability booking is Canceled given no_of_adults = 1 is 24.153204794163628

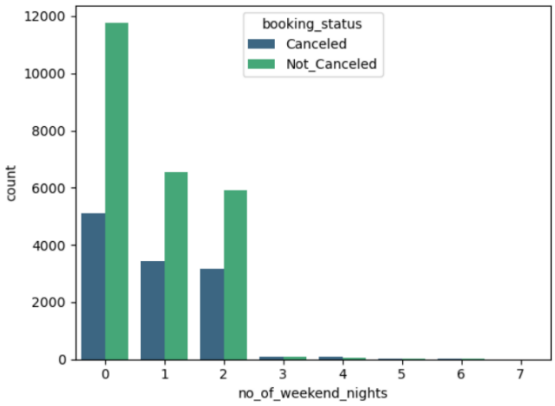
probability booking is Canceled given no_of_adults = 3 is 37.26252158894646

probability booking is Canceled given no_of_adults = 0 is 31.654676258992804

probability booking is Canceled given no_of_adults = 4 is 18.75



- Most bookings are made for 2 adults & 0 children



probability booking is Canceled given no_of_weekend_nights = 1 is 34.39100311276233

probability booking is Canceled given no_of_weekend_nights = 2 is 34.803219049718884

probability booking is Canceled given no_of_weekend_nights = 0 is 30.18610715979137

probability booking is Canceled given no_of_weekend_nights = 4 is 64.34108527131782

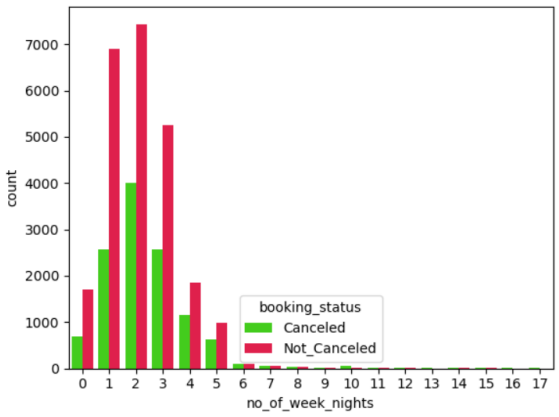
probability booking is Canceled given no_of_weekend_nights = 3 is 48.68421052631579

probability booking is Canceled given no_of_weekend_nights = 6 is 80.0

probability booking is Canceled given no_of_weekend_nights = 5 is 85.29411764705883

probability booking is Canceled given no_of_weekend_nights = 7 is 100.0

insight more number of day customer at hotel , the more likely the customer is going to cancel the booking



probability booking is Canceled given no_of_weekend_nights = 1 is 34.39100311276233

probability booking is Canceled given no_of_weekend_nights = 2 is 34.803219049718884

probability booking is Canceled given no_of_weekend_nights = 0 is 30.18610715979137

probability booking is Canceled given no_of_weekend_nights = 4 is 64.34108527131782

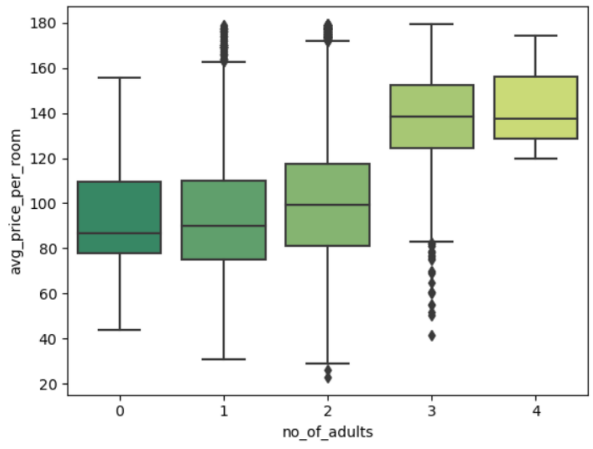
probability booking is Canceled given no_of_weekend_nights = 3 is 48.68421052631579

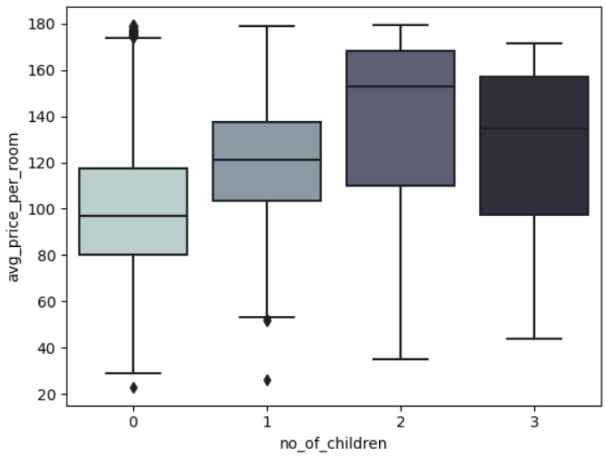
probability booking is Canceled given no_of_weekend_nights = 6 is 80.0

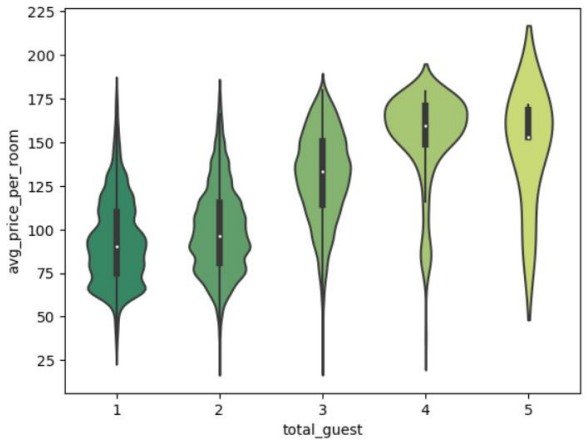
probability booking is Canceled given no_of_weekend_nights = 5 is 85.29411764705883

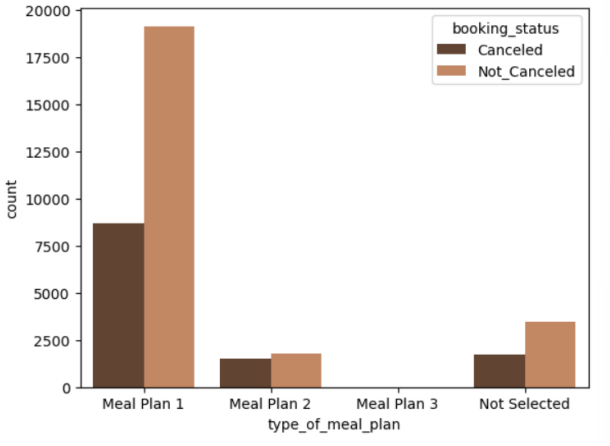
probability booking is Canceled given no_of_weekend_nights = 7 is 100.0

- Majority of the bookings are made over the weekdays (spread over 1-3 days) in comparison to weekends

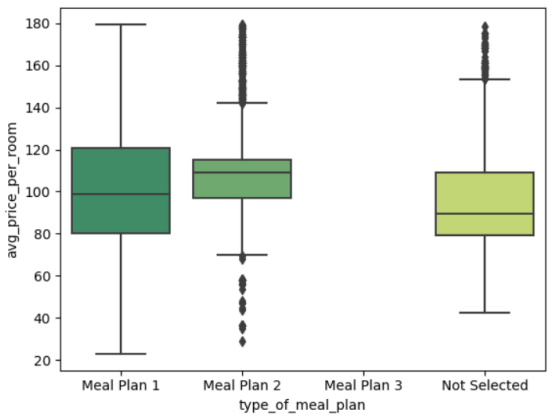


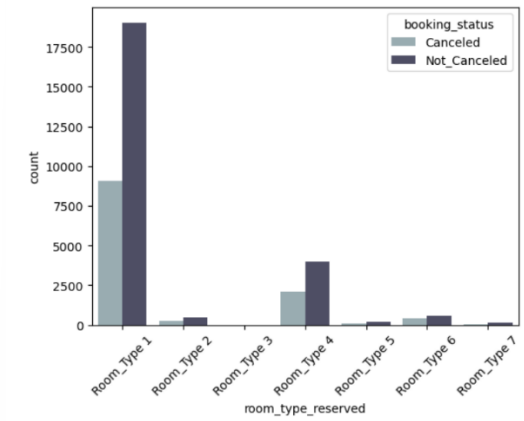




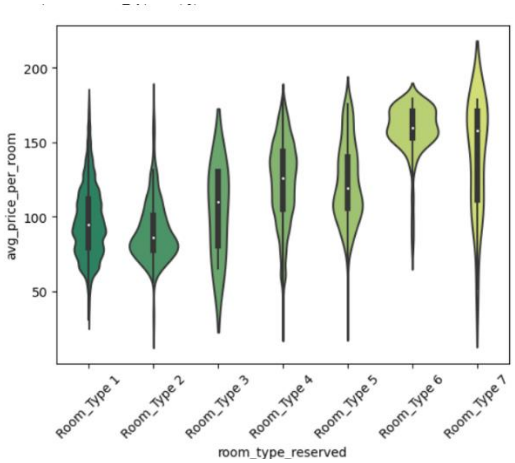


- Most booking are made for Meal Plan 1 (i.e., breakfast only) and negligible for Meal Plan 3 (i.e, all 3 meals). Several do not select the meal plan

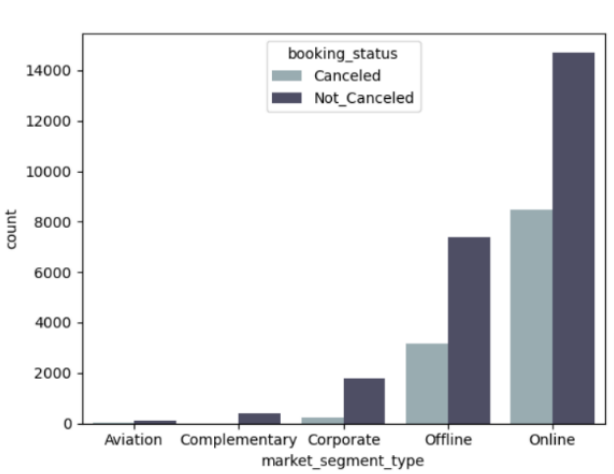


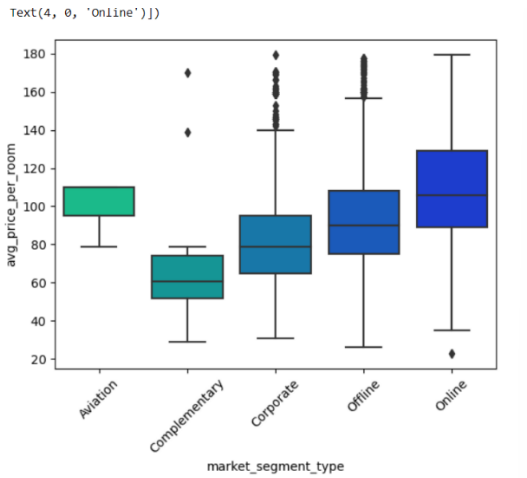


- Most bookings are made with preference for Room_type 1, followed by Room_type 4 & Room_type 6



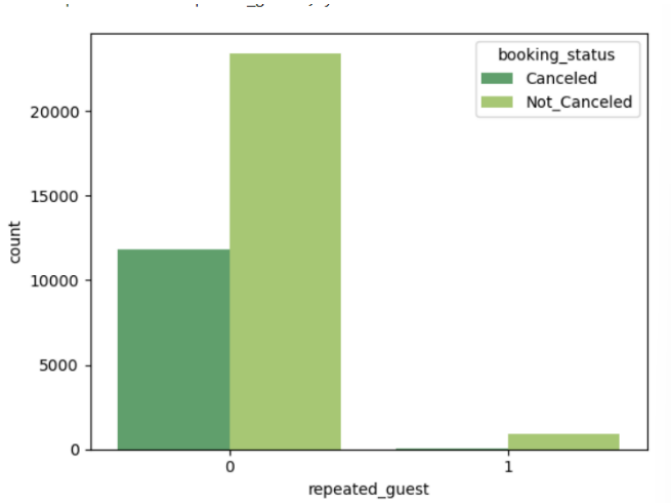
- Majority of bookings were made with Room_Type1, and then followed by Room_Type4 and Room_Type6. However, higher fractions of cancellations wer made in the order Room_Type6 (>50%), Room_Type4 & Room_Type1(30%)



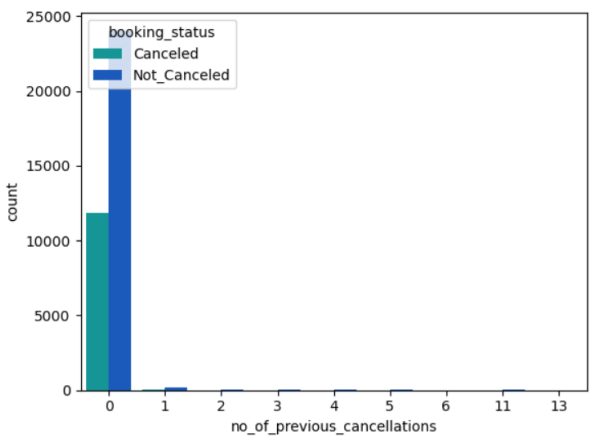


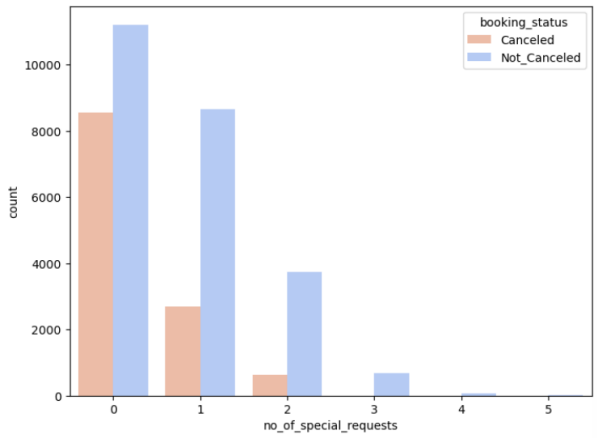
- Most bookings made are online, followed by offline, corporate & then complimentary

- Room prices are dynamic in nature. The min, median & max avg_price_per_room for online market segment is higher than other segments such as aviation, corporate & offline
- Across all market segments, min, median & max avg_price_per_room are higher where bookings have been canceled than when bookings have not been canceled
- Complimentary rooms have the lowest avg_price_per_room across market segments, and have 0 cancellations

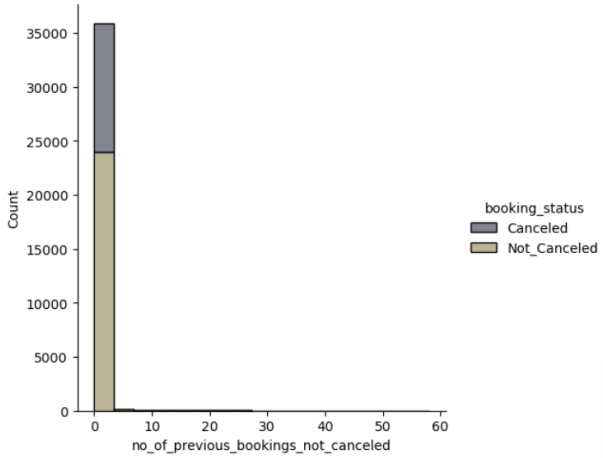


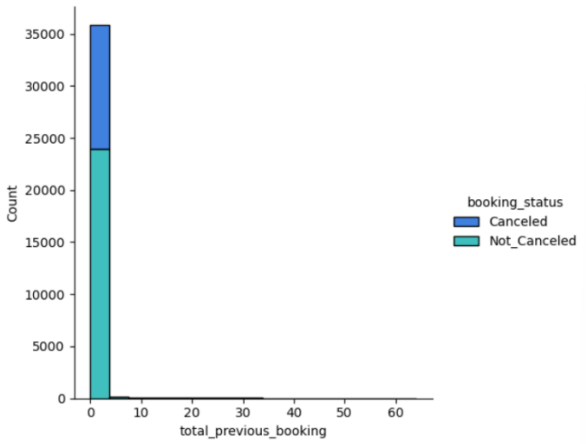
- Majority are not repeated_guests





- More bookings are cancelled when no special_requests are made. Bookings with 3 or more no_of_special_requests have 0 cancellations





EDA SUMMARIZED

- Most bookings are made for 2 adults & 0 children.
- Majority of the bookings are made over the weekdays (spread over 1-3 days) in comparison to weekends.
- Majority of guests book closer to check in-date with both average & median falling under 3 months (90 days). Lead time (time between booking & check-in) is right skewed with several outliers booking more than 6 months (240 days) in advance. The lead time was transformed via a cube-root transformation (to treat skewness). As the lead time increases, it was observed that the odds of (Cancelation: No Cancelation) increases as well.

    - Hotel should introduce policies to restrict how far in advance a booking can be made to decrease the odds for cancelations.
- Most guests have no special requests. Some have 1 or 2 requests and only a minority of guests must make 5 special requests. More bookings are canceled when no special requests are made. Bookings with 3 or more special requests have 0 cancelations.

- The average price per room is skewed right with outliers in the range >£ 200.

- Out of 40,000+ guests, less than 1500 guests indicated needing a parking spot. Out of 40,000+ guests, less than 1500 guests were found to be repeat guests.Out of the <1500 repeat guests, more than 60% have 0 prior canceled bookings and only less than 10% have more than 1 prior canceled bookings

- Dataset has entries between July 2017- August 2019; Summer (March-August) & Winter (September-February). More bookings are made over the summer months (26K+) than the winter months (15K+). About 40% and 20% of all bookings are canceled in summer and winter months respectively

- Majority of customers have the following room order preference: Room type 1 > type 4 > type 6. Cancelation follows the following room order preference: Room type 1 < type 4 < type 6.

    - Hence, a guest preferring a room type 1 is less likely to cancel. Hotel needs to communicate these findings to market each room appropriately.
- Room prices are dynamic in nature. Prices are higher in the online market segment than other segments like aviation, corporate and offline. Across all segments, bookings have been canceled in instances where prices are higher & not canceled when prices are lower. There are no cancelations in Complimentary category.

- Correlations:

    - Correlation observed b/w price per room & no of adults & children which makes intuitive sense.
    - Correlation observed no on weeknights and weekend nights (as longer stays will cover more of both)
    - Linear correlation observed b/w lead time & no on weeknights indicating longer trips are booked in advance.

- Strong relationship observed between previous bookings not canceled & no of previous cancelations (verified by statistical tests)
- Weak correlation observed b/w lead time & price with odd of cancelations being high for both high lead time and high price

MODEL EVALUATION CRITERION

Model can make a wrong prediction as:

- Predicting a person will cancel a booking when a person does not cancel the booking (False Negatives).

This will result in a loss of potential revenue & business for the hotel chain.

- Predicting a person will not cancel a booking, when a person will cancel the booking (False Positives)

This will result in last-minute cancellations -loss of revenue due to hiring of human resources for guests who will no longer come, as well as profit-margin loss in case of trying to price the room cheap to get last minute bookings.

- f1_score should be maximized, the greater the f1_score higher the chances of identifying both the classes correctly.
- The model_performance_classification_statsmodels function will be used to check the model performance of models.
- The confusion_matrix_statsmodels function will be used to plot confusion matrix.

Training set performance:

	Accuracy	Recall	Precision	F1
0	0.778961	0.791561	0.769958	0.78061

Test set performance:

	Accuracy	Recall	Precision	F1
0	0.767564	0.768938	0.764731	0.766828

OBSERVATIONS

The F1 score on the training and testing sets are 0.843 and 0.845, which means model is showing generalized performance on the dataset.

Training performance:

	Accuracy	Recall	Precision	F1
0	0.778961	0.791561	0.769958	0.78061

Feature Selection For our project, we are trying to predict the cancellation status of a booking. After accessing the data, we found that is Canceled and Reservation Status are completely correlated with each other. When cancellation status is 0 then reservation status would be Checkout otherwise cancellation status would be 1. The feature importance generated by decision tree in Figure proves our guess. As only one prediction variable is needed, we decide to drop Reservation Status feature and use is Canceled as a binary label target.

FEATURE EXTRACTION

To enhance the feature description and richness, based on the domain knowledge, three new features are created: • IsFamily: A binary indicator describing whether the hotel guests come as a family or not.

IsFamily =1 if Adults > 0 and [Children > 0 or Babies > 0]

Otherwise

• CustomerNumber:

The total number of customers. CustomerNumber = Adults + Children + Babies

• NightNumber:

The total number of staying nights. NightNumber = StaysInWeekendNights + StaysInWeekNights

DATA PREPROCESSING

- **StandardScaler**

StandardScaler is a preprocessing class from the sklearn.preprocessing module in scikit-learn library that is used to standardize the data by scaling each feature to have a mean of 0 and a variance of 1. This transformation is also known as "standardization" or "Z-score normalization".

The StandardScaler class performs the following steps:

1. Computes the mean and standard deviation of each feature from the training set.
2. Subtracts the mean from each feature.
3. Divides each feature by its standard deviation.

This scaling ensures that each feature has the same scale and prevents features with large values from dominating the model. It is particularly useful for models that are sensitive to the scale of the features, such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM).

- **LabelEncoder**

LabelEncoder is a preprocessing class from the sklearn.preprocessing module in scikit-learn library that is used to convert categorical labels (text-based) into numerical labels. This transformation is necessary for some machine learning algorithms that can only handle numerical data.

The LabelEncoder class performs the following steps:

1. Assigns a unique integer to each unique category in the label column, starting from 0.
2. Replaces the text labels with their corresponding integers.

- **train_test_split**

train_test_split is a function in the Python library scikit-learn that is commonly used for splitting a dataset into two subsets: a training set and a testing set.

The purpose of splitting the data into these two sets is to be able to train a machine learning model on the training set and evaluate its performance on the testing set.

This helps to prevent overfitting of the model to the training data and gives a more accurate estimation of the model's performance on new, unseen data.

The train_test_split function randomly shuffles the data and splits it into two subsets based on a specified ratio, typically 70/30 or 80/20, where the larger subset is used for training and the smaller subset is used for testing.

accuracy_score: Computes the accuracy of the predicted values compared to the true values.

confusion_matrix: Computes a confusion matrix which is a table that summarizes the performance of a classification model.

roc_curve: Computes the Receiver Operating Characteristic (ROC) curve, which is a graphical representation of the performance of a binary classifier.

roc_auc_score: Computes the Area Under the ROC Curve (AUC-ROC), which is a measure of the performance of a binary classifier.

precision_score: Computes the precision of a classification model, which is the ratio of true positives to the total number of predicted positives.

classification_report: Generates a report that includes precision, recall, and f1-score for each class in a classification model.

recall_score: Computes the recall of a classification model, which is the ratio of true positives to the total number of actual positives.

f1_score: Computes the f1-score of a classification model, which is the harmonic mean of precision and recall.

precision_recall_curve: Computes the precision-recall curve, which is a graphical representation of the trade-off between precision and recall.

METHODOLOGIES

Logistic Regression (Baseline)

The baseline model is set as the benchmark for the performance comparison. Since this is a binary classification problem and logistic regression can map all data points into a value between 0 and 1 by the sigmoid function, then it’s used as the baseline model.

Package: sklearn.linear model.LogisticRegression

Decision Tree

The decision tree can break down the problem like binary classification into a bunch of subsets with homogeneous values. The splitting procedure is helpful to show the importance of each feature and thus provide us some insights. Afterall, a big advantage of decision tree is its interpretability which is close to human-being's decision making process.

Package: sklearn.tree.DecisionTreeClassifier

Random Forest

Random forest is an ensemble learning method which can be used for classification. It's comprised by a multitude of decision trees but without the cost of the overfitting prone and thus may have higher accuracy than decision tree.

Package: sklearn.ensemble.RandomForestClassifier

Extra Trees

Extra trees is also ensembled by decision trees. However, different from random forests's subsampling and optimal splitting points, extra trees uses the whole original sample and split nodes by random. In other words, extra trees has a higher degree of randomness but also keeps optimization. Therefore, extra trees may executes faster.

Package: sklearn.ensemble.ExtraTreesClassifier

Support Vector Machine

Support vector machine is a good general-purpose classification algorithm. It aims to find the best decision boundary that splits a dataset into two or more classes by maximum margin. Despite the disadvantage of the O (n2p) complexity, it's still feasible in our problem given the relatively small scale of dataset.

Package: sklearn.svm.SVC

AdaBoost

AdaBoost is a meta-learning method that is initially created to increase the efficiency of binary classifiers and thus perfectly suits this problem. The adaptive

behavior allows it to focus on the mistake of weak classifier and then turn them into stronger ones. As a result, it can seize the depth of detail thoroughly.

Package: sklearn.ensemble.AdaBoostClassifier

GradientBoostingClassifier

Type of machine learning algorithm that belongs to the ensemble learning family. Ensemble learning refers to the process of combining multiple models to improve the overall performance and accuracy of a machine learning algorithm. The key advantages of GBC are its ability to handle both categorical and continuous data, its high accuracy and its robustness to outliers in the data.

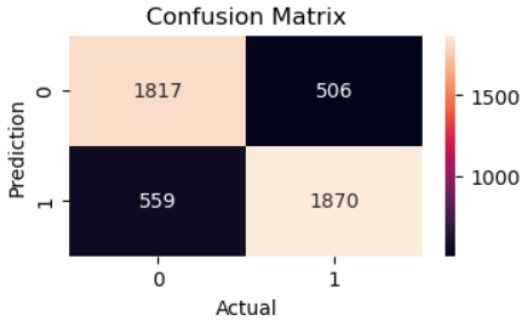
Package: sklearn.ensemble.GradientBoostingClassifier

KNeighborsClassifier

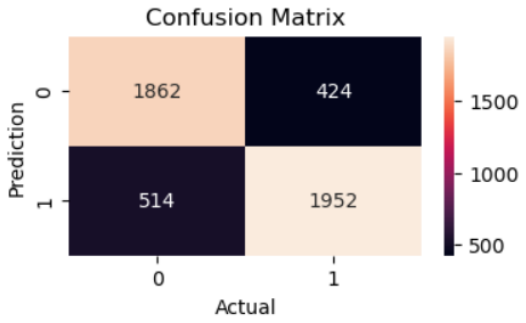
Supervised machine learning algorithm used for classification and regression tasks. KNN is a non-parametric algorithm that works by finding the k nearest neighbors to a given data point in the feature space and then predicting the class of the new data point based on the classes of its nearest neighbors.

Package: sklearn.neighbors.KNeighborsClassifier

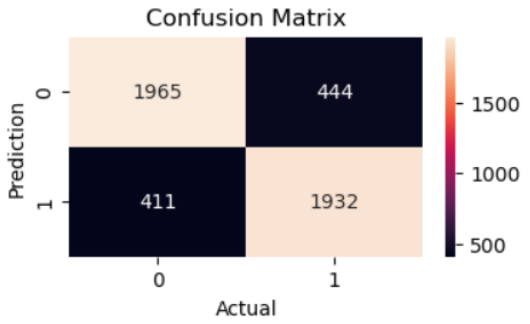
```
Model LogisticRegression
Accuracy: 0.7758838383838383
Precision: 0.7698641416220667
Recall: 0.7870370370370371
F1 Score: 0.7783558792924038
```



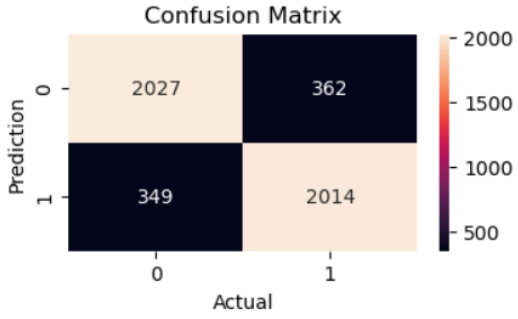

```
Model SVC
Accuracy: 0.8026094276094277
Precision: 0.7915652879156528
Recall: 0.8215488215488216
F1 Score: 0.8062783973564643
```



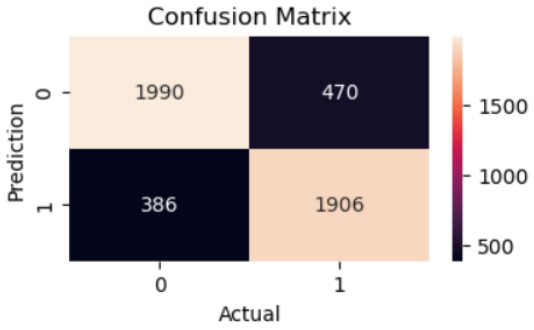
Model KNeighborsClassifier
Accuracy: 0.8200757575757576
Precision: 0.8245838668373879
Recall: 0.8131313131313131
F1 Score: 0.8188175460902734

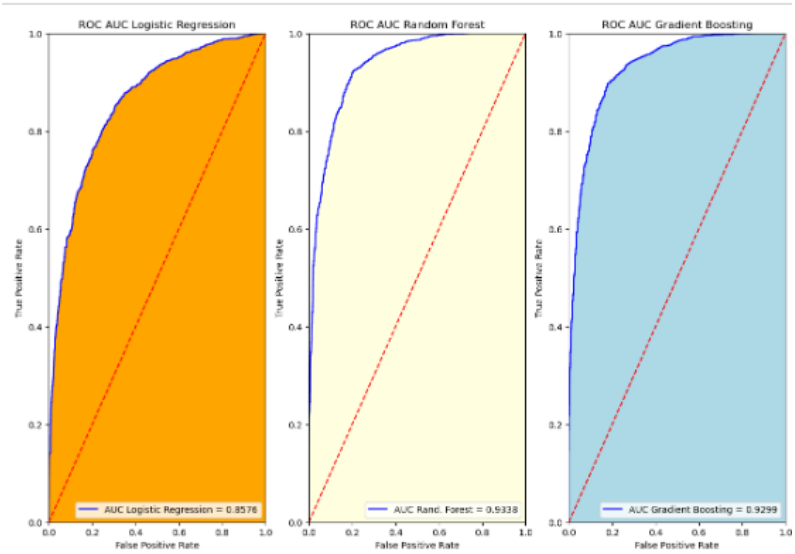


Model RandomForestClassifier
Accuracy: 0.8503787878787878
Precision: 0.8523063901819721
Recall: 0.8476430976430976
F1 Score: 0.8499683477526905



Model DecisionTreeClassifier
Accuracy: 0.8198653198653199
Precision: 0.8315881326352531
Recall: 0.8021885521885522
F1 Score: 0.8166238217652101





CONCLUSION

1. This dataset is a supervised classification dataset.

2. In this dataset I have used many popular Machine Learning algorithms like Logistic Regression, Decision Tree, Random Forest Classifier, AdaBoost Classifier, ExtraTrees Classifier, SVM, GradientBoosting Classifier and KNeighbors Classifier. To predict the cancellation chances.

3. Random forest has the best accuracy among all algorithm that We tried from all the evaluation matrix to predict hotel cancellation classification case, we see that Random Forest Classifier has the best accuracy when it comes to predicting hotel cancellation based on certain features (85.03%).

RECOMMENDATIONS

- ML model is able to predict cancelations or no cancelations for bookings with a confidence of ~86%. Hotel policies for staffing, publicity and dynamic room pricing need to take into consideration the odds for cancelations & have contingency plans in place
- Lead time was identified as the most important feature with a longer lead time increasing the odds for cancelations. Policies need to be introduced to restrict how far in advance bookings can be made before the check in date
- Similarly, hotel policies need to restrict the length of hotel stay as bookings made for longer stay periods were also found to have increased odds of cancelations
- The repeat guests (although few) were identified to have lower odds of cancelations. Hotel policies need to incentivize current & previous guests to increase conversion as repeated guests
- More bookings (as well as more cancelations) were found to occur over months (March - August) than months (September - February). Broadly policies and plans can be formulated estimating business on this biannual basis
- Majority of customers preferred Room Type 1. As well, this room has a pattern of not having as many bookings cancelled. The room has to be adequately marketed, and priced in order to capitalize on its strengths
- Across all market segments, avg price per room has been higher in instances where bookings have been canceled than in instances where bookings have not been canceled. More competition information is required to ensure that our pricing is competitive to retain guests