Hotel Booking Cancelation Prediction using Machine Learning Models

Problem Identification and Project Specification

Farida Simaika
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
faridasimaika@aucegypt.edu

Katia Gabriel
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
katiag@aucegypt.edu

Introduction

The hospitality industry faces a huge problem when it comes to hotel booking cancellations by customers, particularly when they happen close to the scheduled arrival date. According to data on the European market, 49.8% of reservations made on the online booking platform Booking.com were canceled in 2018.

The high prevalence of booking cancellations are without a doubt one of the major concerns for any hotel industry, as the cancellation of hotel reservations that cannot be fulfilled by other guests results in financial losses, failure in optimizing the available resources, and put a huge burden on the revenue management system as it poses high opportunity costs due to the sudden existence of vacant rooms that were not accounted for. Moreover, since the number of guests defines the operational setup for a hotel, an inaccurate estimate of the number of guests might result in an overstaffing or understaffing of personnel and insufficient supplies, which further adds to the problem. Hotel managers usually resort to the implementation of strict cancellation rules and high overbooking limits to resolve the issue at hand, however these terms and conditions may deter customers from ever booking a reservation in a certain hotel, which leads to underbooking and might cause a certain hotel to have a bad reputation in comparison to its competitors.

Thus, there is a critical need to identify potential customers that will most probably cancel their bookings, predict their behavior and manage the risk posed on hotels due to booking cancellations. Tackling this issue is very important as it addresses the continued success of the hotel businesses and is a key factor in the better utilization of the available resources, increasing revenues and reducing opportunity costs, as well as facilitating better business decisions and maintaining a competitive advantage.

The main aim of this project is to leverage various machine learning techniques and models to help managers accurately forecast the demand for hotel reservations. The development and use of precise cancellation forecasting models will help managers identify the motives behind the customer's cancellation decisions and will also reveal various patterns and trends when it comes to customer cancellation. This will help the hotel industry to optimize overbooking restrictions, forecast an accurate net revenue and create appropriate cancellation rules. It will also help managers in understanding the root cause behind customer cancellation behavior which mandates novel solutions that can improve the revenue management system, booking policies and accommodate customer needs thus improving the overall customer satisfaction.

Keywords

Hotel, Booking, Cancellation, Prediction, Forecasting, Machine Learning, Model, Revenue Management System

Literature Review

Hotel industries rely heavily on revenue management systems as their most common business practice. Revenue management is responsible for managing the balance between demand and supply, by effectively allocating the available resources, while maximizing business revenues. Hotel revenue management systems are based on a very crucial assumption about customer demand. Failure to accurately forecast this demand due to booking cancellations causes overstatements of revenues

which poses a risk on the industry (Ivanov, 2014, p. 16). Accordingly, the use of machine learning models plays a vital role in empowering revenue managers by providing them with accurate cancellation prediction models to help them in making informed decisions.

Andriawan et al. (2020) investigated how machine learning can be used to anticipate which customers will cancel their hotel reservations. The researchers used the CRISP-DM methodology. The data used in this study came from two hotels in Portugal, one in Algarve and the other in Lisbon. The study used four different tree-based models. The best performing tree-based model in the study was Random Forest. It had an accuracy rate of 87% . The researchers also highlighted that lead-time (the number of days between the day of booking and the arrival date) was the variable that had the largest impact on the Random Forest algorithm. The article's conclusion states that machine learning can be a useful tool for minimizing revenue losses for hotels.

Antonio, de Almeida and Nunes (2019) created an automated system in collaboration with hotels in Portugal that predict whether customers will cancel their hotel reservations or not. Their model was trained using reservation data from these hotels. The automated model had an accuracy rate of 84% and is based on the XGBoost algorithm. The algorithm made it easy to spot probable cancellations. This allowed the hotel to get in touch with these customers to try and deter them from canceling their reservations. As an attempt to retain them, the hotel management offered these customers complimentary gifts such as free breakfast. The algorithm's reduction in cancellations resulted in annual savings for the hotels of 39 million euros, which clearly shows how accurate forecasting can mitigate revenue losses.

A broader perspective has been adopted by Chen et al. (2022) by developing and comparing three different machine learning models to predict hotel booking cancellations. The data used in developing these various models was provided by Antonio et al. The research also substantiated the fact that as the lead time increases, the cancellation rate recheases 50%. Moreover, the study has identified that there is a positive correlation between lead time and previous cancellations. Three models (Logistic Regression Classifier, k-Nearest-Neighbor(kNN), and CatBoost Classifier) have been developed on the dataset, with CatBoost being the model with the highest accuracy score of 100% in comparison to the kNN model with an accuracy score of 99.8% and Logistic Regression Classifier with an accuracy score of 99.3%. Some limitations were identified during the course of the research, which were the need for more information on the customers' booking changes and past cancellations. Moreover, the researchers pointed out the need for more data about hotel features as well since most of the data reflects only customer behaviors. The addition of this data will help in developing a more credible prediction model. However, the study's main limitation is that the chosen sample data only captures the hotel reservations over the span of two years from July 2015 up until August 2017. The data is not recent which fails to represent economic fluctuations and political instability that might alter customer behaviors in terms of hotel booking cancellation.

Sanchez-Medina et al.(2020) adopted a new approach in order to estimate the likelihood of cancellations in the hotel business. Their study only used 13 explanatory variables which is significantly lower than the average for studies of this type. The variables used were routinely requested from customers by online booking platforms such as Booking and Tripadvisor. Several machine learning techniques (Support Vector Machine, Random Forest, ANN and C5.0) were applied on data from a hotel in Spain. Results indicate that the C5.0 algorithm produces better results in terms of the area under the ROC curve (AUC). The Random Forest algorithm exhibits higher accuracy. It was also observed that the SVM performs worse than the Random Forest algorithm in terms of accuracy, precision and F-score. The ROC curve shows that when ANN is optimized with GA, this approach gives all metrics (accuracy,precision, F-score, specificity and AUC) with values over 0.95 and has the best performance overall. The researchers highlighted that accurate cancellation forecasting encourages hoteliers to make wise managerial choices while offering a significant competitive edge.

Overall, the literature sheds light on the importance of applying machine learning techniques to accurately predict customer behavior and potential customer cancellation to support the revenue management with the needed information.

Methodology

The project will be divided into three main procedures which are data preparation, machine learning model development and evaluation, as well as model comparison that will be executed using Python.

The data preparation phase will entail the exploration of the data and understanding its features. This step involves the preprocessing of the data by identifying the existence of missing data and the best way in dealing with them. Moreover, various encoding techniques should be carried out in dealing with categorical data, as well as standardization techniques. Descriptive statistics of various numerical variables will be calculated to have a closer look on the nature of the variables and to determine the existence of any relationship between variables that could be of any relevance in further steps. After the data preparation phase the data should be ready for data analysis and the development of machine learning models. The data set will be divided into a training set and a validation set.

The machine learning model development and evaluation phase will encompass the development of various machine learning models to help in accurately predicting future hotel booking cancellation. Previous studies, as discussed in the literature review have based their prediction on machine learning models such as tree-based models, XGBoost algorithm, Logistic Regression Classifier, kNN, and CatBoost Classifier. This project will further develop other machine learning models in an attempt to find the most relevant features in combination with the best machine learning model in forecasting the hotel booking cancellations. Once the machine learning model is trained, the model will be evaluated using a validation data set to detect possible overfitting problems. Different performance indicators will then be used on the trained model in order to assess its overall performance, strength and weaknesses.

In the last phase all machine learning models will be put in comparison to choose the machine learning model that yields the best outcome.

Data Overview

A thorough investigation was conducted in order to find a data set that is appropriate for a machine learning project. Two data sets regarding hotel cancellation were found. The selected data sets have the same source and contain the same features. However, only one appeared to be better suited for a machine learning project and aligned best with our goal. The data set that was disregarded contained only 16 variables which is not enough to train a good model.

The chosen data set captures some of the most relevant information about hotel reservation processes. It contains 32 predictor variables and encompasses 119,390 observations. The label, which is the variable to be predicted, is the “is_canceled” variable, which identifies the status of each reservation as either canceled or not canceled. Some of the features included in the dataset that will be further explored in predicting the label are lead-time, country, hotel type, room type, etc.

Variable	Description
hotel	Categorical variable that differentiates between two hotel types (H1: Resort Hotel and H2: City Hotel)
is_canceled	Binary variable indicating if the booking was canceled (1) or not (0)
lead_time	Numerical variable indicating the number of days that elapsed between the day of booking and the arrival date.
arrival_date_year	Year of arrival date (2015-2017).
arrival_date_month	Categorical variable indicating the month of the arrival date.
arrival_date_week	Numerical variable indicating the year’s week number for the arrival date.

arrival_date_day	Numerical variable indicating the day of arrival date.
stays_in_weekend_nights	Numerical variable indicating the number of weekend nights (Sunday and Saturday) the guest booked.
stays_in_week_nights	Numerical variable indicating the number of week nights (Monday to Friday) the guest booked.
adults	Numerical variable indicating the number of adults to stay at the hotel.
children	Numerical variable indicating the number of children to stay at the hotel.
babies	Numerical variable indicating the number of babies to stay at the hotel.
meal	Categorical variable differentiating between the type of meal booked (HB: half board, BB: bed and breakfast, SC: self-catering, FB: full board)
country	Categorical variable indicating the guest's country of origin (in ISO format)
market_segment	Categorical variable that differentiates between market segment designations (TA: travel agents, TO: Tour operators)
distribution_channel	Categorical variable that differentiates between booking distribution channels (TA: travel agents, TO: Tour operators)
is_repeated_guest	1
previous_cancellation	Numeric variable that indicates the number of any previous cancellation made by hotel guests.
previous_booking_not_canceled	Numeric variable that indicates any previous bookings made by a guest at a certain hotel and was fulfilled.
reserved_room_type	Categorical variable that indicates the various room types that were reserved by hotel guests. This variable has 10 levels, with each room type assigned a letter from the alphabet (A, B, C, D, E, F, G, H, L, and P).
assigned_room_type	Categorical variable that indicates if a hotel guest got assigned a different room type than the one reserved. This variable has 10 levels , with each room type assigned a letter from the alphabet (A, B, C, D, E, F, G, H, L, and P).
booking_changes	Numerical Variable that indicates the number of booking changes made by hotel guests.
deposit_type	Categorical variable that indicates the deposit type. This variable has 3 levels. (no deposit: during room reservation the

	guest is not required to pay a deposit, No refund: in case of a cancellation made by the guest the amount of money paid in advance is non-refundable, Refundable: in case of a cancellation made by the guest the amount of money paid is refundable).
agent	Numeric variable that indicates the hotel agent sitting on the front desk of a hotel who is responsible for booking the rooms for the customers, verifying the reservation, and collecting payments.
company	Numeric variable that gives each reserving company an ID.
days_in_waiting_list	Numeric variable that indicates the number of days a customer has to wait before he can get waitlisted when all hotel rooms are fully-booked.
customer_type	Categorical variable that indicates the type of customers that are booking rooms in the hotel. The variable has 4 levels. (transient: individuals that need a short, mostly urgent stay at a hotel, Transient-party: individuals that need a short, mostly urgent stay at a hotel but are associated with at least another transient booking, Group: guests that

	request an accommodation at a hotel as a group, e.g. families, tourist groups, school groups, etc., Contract: employees that request a reservation at a hotel through a business to business contract, which offers them rooms at discounted rates).
adr	Numerical variable indicating the average daily rate of a hotel which measures the rental revenue of a room per day.
required_car_parking_spaces	Numerical variable that indicates how many car parking spots a guest needs at a certain hotel.
total_of_special_requests	Numerical variable which indicates the total amount of special requests made by customers during their stay such as extra beds.
reservation_status	Categorical variable with 3 levels that states the reservation status of each guest. (Check-Out: the guest has fulfilled his stay at the hotel, Canceled: the guest has canceled his reservation prior to the date of arrival, No-Show: the guest neither fulfilled nor canceled his reservation and did not show up on the day of check in).
reservation_status_date	Variable that gives the

	day, month, and year at which the reservation status was changed from reserved to checked-out, canceled, or no-show.

Bibliography

Andriawan, Z. A., Purnama, S. R., Darmawan, A. S., Wibowo, A., Sugiharto, A., Wijayanto, F. et al. (2020), Prediction of hotel booking cancellation using crisp-dm, in '2020 4th International Conference on Informatics and Computational Sciences (ICICoS)', IEEE, pp. 1–6.

Antonio, N., de Almeida, A. & Nunes, L. (2017), 'Predicting hotel booking cancellations to decrease uncertainty and increase revenue', Tourism & Management Studies 13(2), 25– 39

El Hadad, R., Roper, A., & Jones, P. (2008). The Impact of Revenue Management Decisions on Customers' Attitudes and Behaviors: A Case Study of a Leading UK Budget Hotel Chain. *EuroCHRIE 2008 Congress,*

Ivanov, S. (2014). *Hotel Revenue Management: From Theory to Practice*. Zangador.

Nuno Antonio, Ana de Almeida, Luis Nunes, Hotel booking demand datasets, Data in Brief, Volume 22, 2019, page 41-49 doi.org/10.1016/j.dib.2018.11.126

Sanchez-Medina, A. J., Eleazar, C. et al. (2020), 'Using machine learning and big data for efficient forecasting of hotel booking cancellations', *International Journal of Hospitality Management* 89, 102546.

Hotel Booking Cancelation Prediction using Machine Learning Models

Data Preparation, Cleaning and feature Engineering

Farida Simaika
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
faridasimaika@aucegypt.edu

Katia Gabriel
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
katiag@aucegypt.edu

Description of The Data Set

A thorough investigation was conducted in order to find a data set that is appropriate for a machine learning project. Two data sets regarding hotel cancellation were found. The selected data sets have the same source and contain the same features. However, only one appeared to be better suited for a machine learning project and aligned best with our goal. The data set that was disregarded contained only 16 variables which is not enough to train a good model.

The chosen data set captures some of the most relevant information about hotel reservation processes. It contains 32 predictor variables and encompasses 119,390 observations. The feature variables are of type categorical variables and numerical variables. The categorical variables are: “hotel”, “meal”, “country”, “market segment”, “distribution channel”, “is repeated guest”, “reserved room type”, “assigned room type”, “deposit type”, “customer type”, and “reservation status”. The numerical variables are: “lead time”, “arrival date year”, “arrival date month”, “arrival date week”, “arrival date day”, “stays in weekend nights”, “stays in week nights”, “adults”, “children”, “babies”, “previous cancellations”, “previous booking not canceled”, “booking changes”, “days in waiting list”, “adr”, “required car parking spaces”, and “total of special requests”. The label, which is the variable to be predicted, is the “is_canceled” variable, which identifies the status of each reservation as either canceled or not canceled. The features in this dataset will be further explained and analyzed, in an attempt to identify the most important features that are useful in predicting future hotel booking cancellation. A detailed description of each variable in this dataset is displayed in the following table:

Variable	Description
hotel	Categorical variable that differentiates between two hotel types (H1: Resort Hotel and H2: City Hotel)
is_canceled	Binary variable indicating if the booking was canceled (1) or not (0)
lead_time	Numerical variable indicating the number of days that elapsed between the day of booking and the arrival date.
arrival_date_year	Year of arrival date (2015-2017).
arrival_date_month	Categorical variable indicating the month of the arrival date.
arrival_date_week	Numerical variable indicating the year’s week number for the arrival date.
arrival_date_day	Numerical variable indicating the day of arrival date.
stays_in_weekend_nights	Numerical variable indicating the number of weekend nights (Sunday and Saturday) the guest booked.
stays_in_week_nights	Numerical variable

	indicating the number of week nights (Monday to Friday) the guest booked.
adults	Numerical variable indicating the number of adults to stay at the hotel.
children	Numerical variable indicating the number of children to stay at the hotel.
babies	Numerical variable indicating the number of babies to stay at the hotel.
meal	Categorical variable differentiating between the type of meal booked (HB: half board, BB: bed and breakfast, SC: self-catering, FB: full board)
country	Categorical variable indicating the guest's country of origin (in ISO format)
market_segment	Categorical variable that differentiates between market segment designations (online TA (travel agent), offline TA, groups, direct, corporate, complementary and aviation.)
distribution_channel	Categorical variable that differentiates between booking distribution channels (TA: travel agents, TO: Tour operators Direct, corporate and GDS)
is_repeated_guest	1

previous_cancellation	Numeric variable that indicates the number of any previous cancellation made by hotel guests.
previous_booking_not_ canceled	Numeric variable that indicates any previous bookings made by a guest at a certain hotel and was fulfilled.
reserved_room_type	Categorical variable that indicates the various room types that were reserved by hotel guests. This variable has 10 levels, with each room type assigned a letter from the alphabet (A, B, C, D, E, F, G, H, L, and P).
assigned_room_type	Categorical variable that indicates if a hotel guest got assigned a different room type than the one reserved. This variable has 10 levels , with each room type assigned a letter from the alphabet (A, B, C, D, E, F, G, H, L, and P).
booking_changes	Numerical Variable that indicates the number of booking changes made by hotel guests.
deposit_type	Categorical variable that indicates the deposit type. This variable has 3 levels. (no deposit: during room reservation the guest is not required to pay a deposit, No refund: in case of a cancellation made by the guest the amount of money paid in advance is

	non-refundable, Refundable: in case of a cancellation made by the guest the amount of money paid is refundable).
agent	Numeric variable that indicates the hotel agent sitting on the front desk of a hotel who is responsible for booking the rooms for the customers, verifying the reservation, and collecting payments.
company	Numeric variable that gives each reserving company an ID.
days_in_waiting_list	Numeric variable that indicates the number of days a customer has to wait before he can get waitlisted when all hotel rooms are fully-booked.
customer_type	Categorical variable that indicates the type of customers that are booking rooms in the hotel. The variable has 4 levels. (transient: individuals that need a short, mostly urgent stay at a hotel, Transient-party: individuals that need a short, mostly urgent stay at a hotel but are associated with at least another transient booking, Group: guests that request an accommodation at a hotel as a group, e.g. families, tourist groups, school groups, etc., Contract: employees

	that request a reservation at a hotel through a business to business contract, which offers them rooms at discounted rates).
adr	Numerical variable indicating the average daily rate of a hotel which measures the rental revenue of a room per day.
required_car_parking_ spaces	Numerical variable that indicates how many car parking spots a guest needs at a certain hotel.
total_of_special_requests	Numerical variable which indicates the total amount of special requests made by customers during their stay such as extra beds.
reservation_status	Categorical variable with 3 levels that states the reservation status of each guest. (Check-Out: the guest has fulfilled his stay at the hotel, Canceled: the guest has canceled his reservation prior to the date of arrival, No-Show: the guest neither fulfilled nor canceled his reservation and did not show up on the day of check in).
reservation_status_date	Variable that gives the day, month, and year at which the reservation status was changed from reserved to checked-out, canceled, or no-show.

Analysis of the Feature Variables

The different features of the data set will be analyzed in terms of statistical distribution, missing values, correlation with the label "is_canceled" and overall relevance.

1. Hotel Feature Analysis

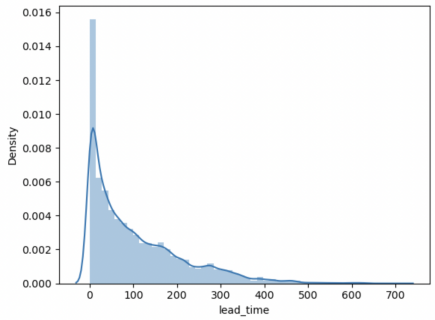
This feature is a categorical variable differentiating between two types of hotels in Portugal: City Hotel and Resort Hotel. The variable did not present any missing values. Graphical representations of this feature allowed us to primarily infer that city hotels have more cancellations than resort hotels. The cancellation rates in both city and resort hotels were calculated. City hotels (encoded with label =1) receive on average more bookings than resort hotels and hence more cancellations. City Hotels have an almost 42% cancellation rate. On the other hand, resort hotels receive less bookings and have a lower cancellation rate of 29%.

hotel	is_canceled	count
0	0	71.990420
	1	28.009580
1	0	58.081655
	1	41.918345

A Chi-Squared test was performed in order to assess the existence of a relationship between the feature and the label. The hypotheses were the following: H0: there is no correlation and H1: a correlation exists between the variables. The output of the Chi-Squared test was very close to 0 which is less than the significance level. We reject the null hypothesis, there is a significant relationship between the variables.

2. Lead Time Feature Analysis

The lead time feature is a numerical variable indicating the number of days that elapsed between the day of booking and the arrival date. The variable presented no missing values. It had the highest correlation coefficient with the label. The statistical distribution of the feature shows skewness.

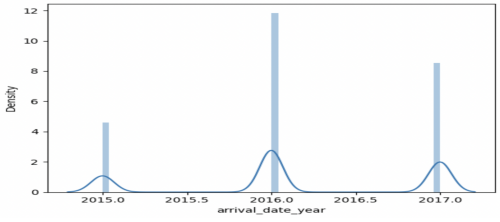


Plotting the feature against the label and calculating cancellation percentages lead us to conclude that the longer the lead time, the more likely the cancellation of the booking. The higher lead time increases the chances of unforeseen circumstances occurring that could derail the guests travel plans. It was observed that bookings with a lead time of more than 7 months had more than a 50% chance of cancellation.

is_canceled	0	1
lead_time_month		
0	81.573543	18.426457
1	63.533944	36.466056
2	60.179641	39.820359
3	55.886009	44.113991
4	56.363185	43.636815
5	53.739086	46.260914
6	55.195941	44.804059
7	53.003210	46.996790
8	44.806517	55.193483
9	36.065097	63.934903
10	30.626366	69.373634
11	29.362416	70.637584
12	42.008197	57.991803
13	37.500000	62.500000
14	27.255639	72.744361
15	35.159011	64.840989
16	17.073171	82.926829
17	18.852459	81.147541
18	25.274725	74.725275

3 and 4. Arrival Year Feature Analysis

The arrival year feature captures the year of the guest's arrival date. The years in this data set range between 2015 and 2017. The statistical distribution of the variable shows that each year seems to be approximately normally distributed. The variable presented no missing values.



In 2017, the Portuguese hotel industry had the highest number of booking cancellations reaching a 39% cancellation rate. However, the cancellation rate remained constant throughout the years going from 35% in 2015 to 39% in 2017. The feature presents a weak correlation with the label.

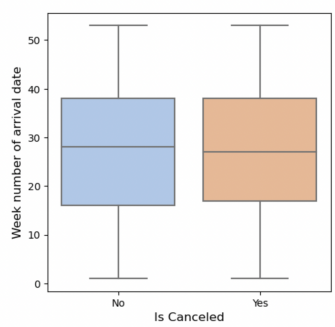
5. Arrival Month Feature Analysis

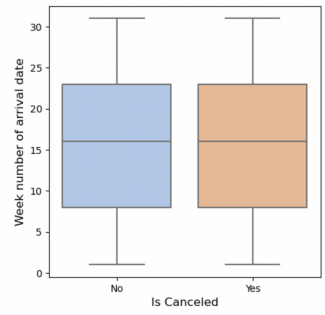
The arrival month feature is a categorical variable with 12 classes (months of the year). The variable identifies the month of the customer's arrival date. The feature presented no missing values as well. A Chi-Squared test (at significance level of 5%) was conducted to determine the existence of a significant relationship between the label and the arrival month. The test yielded to a p-value of less than 5% which resulted in rejecting the null hypothesis that stated that there was no correlation between the label and the feature. Throughout the years, the months of June and July had the highest reservation rates . However, in June, almost half of the hotel bookings were canceled. This might be due to the fact that Portugal is a summer destination and hence the month of June is a summer peak season with increased bookings and cancellations.

arrival_date_month	is_canceled	count
April	0	59.134920
	1	40.865080
August	0	62.140067
	1	37.859933
December	0	64.645258
	1	35.354742
February	0	66.429374
	1	33.570626
January	0	69.298843
	1	30.701157
July	0	62.376002
	1	37.623998
June	0	58.401395
	1	41.598605
March	0	67.646756
	1	32.353244
May	0	60.083675
	1	39.916325
November	0	68.451584
	1	31.548416
October	0	61.666968
	1	38.333032
September	0	60.753457
	1	39.246543

6 and 7. Arrival Week and Day of Month Feature Analysis

The two features are numerical variables that capture respectively the day and the week number of the guest's arrival date. The aforementioned variables have insignificant correlation coefficients with the label. In addition, the number of unique values of each feature is significant. Upon plotting the features, there was no difference in their statistical distributions when plotted against the label.

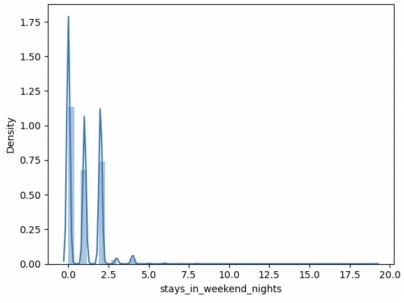




One can infer that these variables are not factors that determine whether or not a guest will cancel their hotel reservation. The arrival year and month are much more significant and provide more insights on the customer's arrival date. These variables will therefore be dropped.

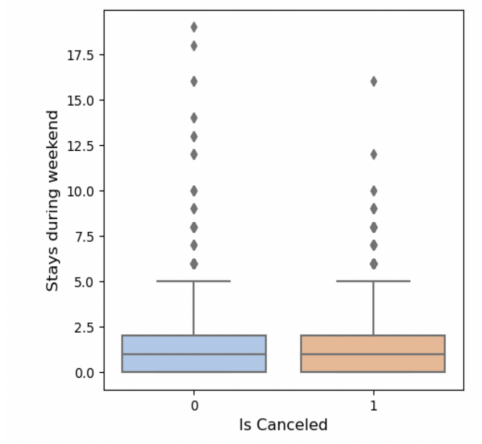
8. Stays in Weekends and Weeknights Feature Analysis

Both features are numerical variables indicating the number of weekend nights (Sunday and Saturday) and week nights (Monday to Friday) the guest has booked. The distribution of the features is heavily skewed.



The two features both have an insignificant correlation coefficient with the label. The heat map of the variables in this data set underscored a

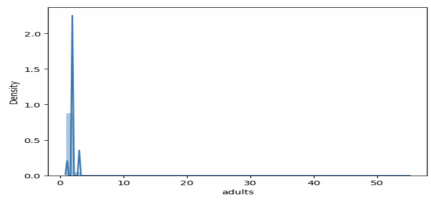
collinearity problem between the two features. The correlation coefficient between the number of stays in weekend and weeknights is 0.52. An appropriate course of action would be to drop one of the variables since the other is basically redundant. Upon plotting the features against the label, one can notice that the statistical distribution of the variables does not change.



We can therefore hypothesize that both variables are insignificant and do not contribute in predicting whether or not the guest will cancel their hotel booking. Instead of deleting both of these variables at the risk of losing significant data, a new feature called "total_stay" will be created combining the aforementioned variables. The feature engineering section of the report explains the significance of this new variable in detail.

9. Adults Feature Analysis

The feature adult is a numerical variable capturing the number of adults that made a hotel reservation. The variable did not present any missing values. The statistical distribution of the variable adult is skewed. Most hotel bookings have around 2 to 3 adults under a single reservation.



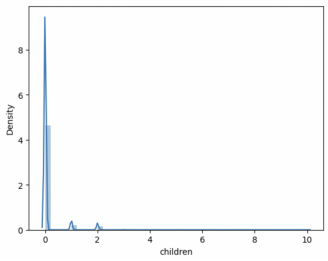
The feature presents a very low correlation coefficient with the label. One can infer that the number of adults does not contribute in predicting whether or not a booking will be canceled. The cancellation percentages according to the number of adults per booking were calculated.

adults	is_canceled	count
1	0	70.808324
	1	29.191676
2	0	60.511602
	1	39.488398
3	0	65.238480
	1	34.761520
4	0	74.193548
	1	25.806452
5	1	100.000000
6	1	100.000000
10	1	100.000000
20	1	100.000000
26	1	100.000000
27	1	100.000000
40	1	100.000000
50	1	100.000000
55	1	100.000000

As previously stated, the number of adults did not seem to interfere with the cancellation of a booking. An increased number of adults under a single booking did not prevent the guests from canceling it and vice versa. This feature will be used to create another variable in the data set. This new variable and its significance are discussed in detail in the feature engineering section of the report.

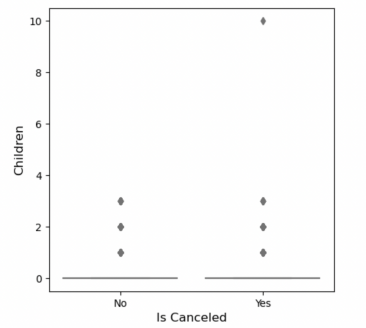
10 and 11. Children and Babies Feature Analysis

The feature children is a numerical variable that indicates the number of children in a booking reservation. The feature babies is a numerical variable capturing the number of babies under a single booking. The source of the data set does not include differentiation between children and kids. The statistical distribution of the children variable seems to be skewed. Only 8% of adults in this data set have children.



The same applies to the statistical distribution of the babies variable. It is skewed as well with the mode of the distribution being 0. Most adults in this data set

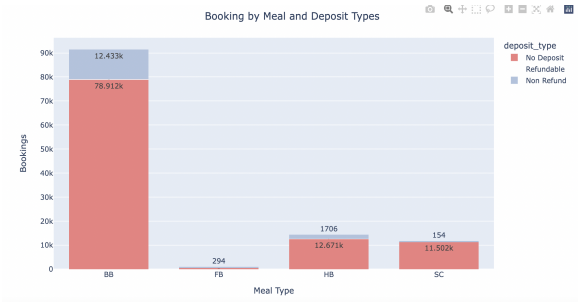
do not have babies. The correlation coefficient between the variable babies and the label is weak. The feature adults are also weakly correlated to the label. Upon calculating cancellation percentages and plotting different visualizations, we were able to hypothesize that in general the number of babies and children did not affect the status of the booking.



In addition, a relationship between the number of children/babies and the number of special requests was examined. It was suspected that the higher the number of children/kids, the more the special requests (such as placing a crib in the room) and thus decreasing the probability of booking cancellation. The initial hypothesis was erroneous, there is no relationship between the number of children/babies and the number of special requests placed under a booking. As mentioned before , both variables have a very weak correlation with the label. The feature engineering section of the report discusses in detail the process that was implemented to address this issue.

12. Meal feature Analysis

The meal feature is a categorical variable differentiating between the type of meal booked. The variable has four categories: HB: half board, BB: bed and breakfast, SC: self-catering and FB: full board. A Chi-Square test was conducted in order to determine the existence of a significant relationship between the variable meal and the label. The Chi-Squared test was performed with a significance level of 5%. The output of the test was lower than the significance level and very close to zero. We therefore rejected the null hypothesis that stated that there was no relationship between the feature and the label.The most common meal type is the BB (bed and breakfast). The reason behind this might be related to the deposit type. As shown below, meals of type BB require no deposit to be made for the booking. Customers generally prefer the no-deposit type allowing them to cancel their booking freely without losing their money.



However, the meal type BB did not have the highest cancellation rate (only 37%). It is the FB (full board) meal type that had the highest cancellation rate of 59%.

meal	is_canceled	count
BB	0	62.406557
	1	37.593443
FB	0	40.025094
	1	59.974906
HB	0	65.347705
	1	34.652295
SC	0	63.730703
	1	36.269297

The main factor behind this high cancellation rate is the deposit type. As can be seen from the above graph, Full Board meals have the highest non-refund deposit ratio. This type of deposit has historically a high cancellation rate.

13. Country Feature Analysis

The variable country is a categorical variable that indicates the guest's country of origin. The variable presented many missing values. The approach that was used to deal with the missing value will be explored in detail in the data preprocessing section of the report. The feature has more than 177 unique variables. The countries of origin are written using the ISO code format. A Chi-Square test was conducted (with a significance level of 5%) in order to assess the existence of a significant relationship between the feature and the label. The Chi-Square test yielded to a p-value less than the chosen significance level. We were then able to reject the null hypothesis that stated that there was no significant relationship between the variable and the

label. In this data set, most guests are Portuguese. The hotels in this data set are located in Portugal, so we can infer that most guests at these hotels are local customers.

A feature with more than 177 unique values is impossible to deal with efficiently. The high number of unique values makes it very difficult to make rigorous and accurate inferences. The data pre-processing section of the report discusses the approach that was implemented to deal with this problem.

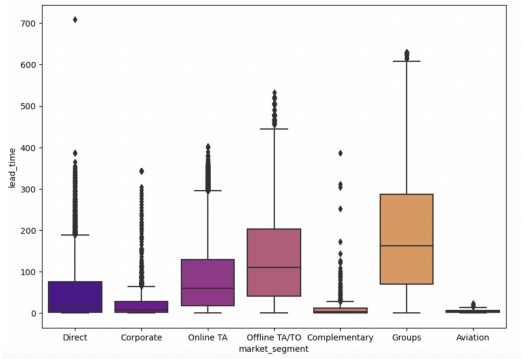
14. Market segment feature analysis

Market segment is a categorical feature that differentiates between different market segment designations. The variable has 7 categories: online TA (travel agent), offline TA, groups, direct, corporate, complementary and aviation. The feature basically explains how the guest made the booking (directly from the website, through a travel agent..).

Market segment is a categorical variable so a Chi-Square test was performed in order to determine the existence of a significant relationship between the feature and the label. A significance level of 5% was chosen. The p-value of the Chi-Square test was less than the significance level of 5%. Since the output of the test was less than the significance level, we reject the null hypothesis that stated that there was no relationship between the feature and label. Cancellation percentages for each category were calculated. The highest cancellation rate (62%) was among the market segment category “groups”. The market segment “direct” and “complementary” had the lowest cancellation rates of 15% and 12% respectively.

market_segment	is_canceled	count
Aviation	0	77.922078
	1	22.077922
Complementary	0	87.904360
	1	12.095640
Corporate	0	81.093481
	1	18.906519
Direct	0	84.523714
	1	15.476286
Groups	0	38.794412
	1	61.205588
Offline TA/TO	0	65.482825
	1	34.517175
Online TA	0	63.091471
	1	36.908529

The reasons behind these cancellation rates were explored. It was found that market segments with a high cancellation rate had a high lead time and vice versa.



As previously mentioned, a high lead time increases the probability of booking cancellation.

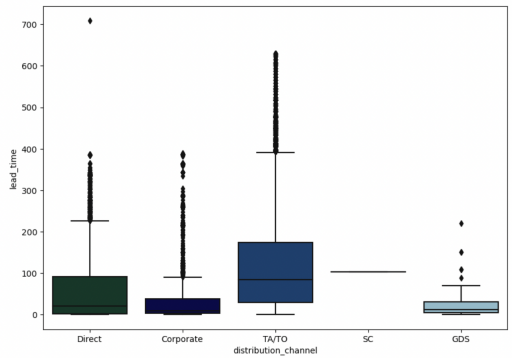
15. Distribution Channel feature analysis

This feature is a nominal feature differentiating between different hotel booking channels. The feature has 5 categories: TA: travel agents, TO: Tour operators, Direct, corporate and GDS (global distribution system). The Chi-Square test (with a significance level of 5%) was conducted in order to assess the existence of a relationship between the variable and the label. The Chi-Square test yielded to a value that was less than the significance level of 5%. We were able to reject the null hypothesis that stated that there was no relationship between the feature and label. There is in fact a significant relationship between them. Most guests in this data set booked their reservation through a tour operator/tour agent. Since tour agents were the main booking channel, it is logical that they also had the highest cancellation rates of almost 42%. This cancellation rate is twice as high as the cancellation rates for other distribution channels.

distribution_channel	is_canceled	count
Corporate	0	77.794654
	1	22.205346
Direct	0	82.368038
	1	17.631962
GDS	0	80.526316
	1	19.473684
SC	0	100.000000
TA/TO	0	58.776056
	1	41.223944

The factors behind the high cancellation rates were explored. Similar to the market segment feature, it

was found that the lead time plays a crucial role in determining these cancellation rates.



The category TA/TO (tour agent and operators) has the highest lead time. Previous analysis demonstrated that higher lead time increased the probability of cancellation.

16. Is Repeated Guest Feature Analysis

This feature variable attempts to identify any existing patterns in the customer behavior based on past events. By observing the total number of returning guests it is obvious that the majority (= 115580 observations) are new guests that haven't booked a room in their respective hotels before. Only (= 3810 observations) are repeated guests.

is_repeated_guest	0	1
is_canceled		
0	71908	3258
1	43672	552

As can be seen from the contingency table above, out of the 3810 repeated guests only 552 guests canceled their booking. Accordingly, it can be assumed that returning guests are most likely to fulfill their bookings. The following customer behavior can be further underscored with the bar plot below.

A chi-square test was performed to be able to determine the existence of a correlation between this feature variable and the label variable. The following Hypotheses were tested:

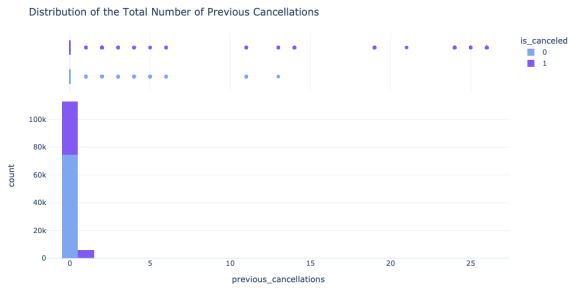
H_0 **:** there exists no correlation

H_1: there exists a correlation

The p-value chosen for this test was 0.05. The Chi-squared test yielded a value that is very close to zero, which is 1.7841252215934033e-188. Since this value is <0.05, the null hypothesis was rejected and there exists a correlation between this feature variable and the label.

17. Previous Cancellations Feature Analysis

This feature variable captures the number of previous cancellations made by repeated guests. The statistical distribution of these variables show that the majority of observations in this dataset are new guests. This is simply because, as mentioned above, this dataset captures information mostly about new guests. However, out of the previously canceled customers, it seems that as the number of previous cancellations increase, the potential of the current reservation to be also canceled is high.



The correlation between this feature variable and the label variable is around 0.11 which shows a weak positive relationship.

18. Previous Booking Not Canceled Feature Analysis

As a continuation of the two aforementioned variables, this variable depicts previous bookings that were not canceled. Similar to the "Is Repeated Guest" feature, the statistical distribution shows that the majority of hotel bookings had no previous bookings that were not canceled.

However, as can be seen from the boxplots, returning guests who had made previous bookings and didn't cancel did not cancel their present bookings. Only a very small percentage of guests with previous bookings not canceled had to cancel their bookings. The correlation between this feature variable and the label variable is around -0.05 which shows a weak negative relationship.

19. Reserved Room Type Feature Analysis

This feature variable shows the room type preference of guests. As can be seen from the barplot below, the room type with the most reservations is room type A, followed by room type D.

Looking at the contingency table below it becomes evident that guests who book room type P always cancel their reservations.

reserved_room_type	A	B	C	D	E	F	G	H	L	P
is_canceled										
0	52364	750	624	13099	4621	2017	1331	356	4	0
1	33630	368	308	6102	1914	880	763	245	2	12

The cancellation rates based on the room type reserved are very similar to each other with cancellation rates ranging from 29% - 41%. The room type with the highest cancellation rate is room type P with 100% cancellation rate. It is evident that the probability of cancellation when the reserved room type is P is very high.

A chi-square test was performed to be able to determine the existence of a correlation between this feature variable and the label variable. The following Hypotheses were tested:

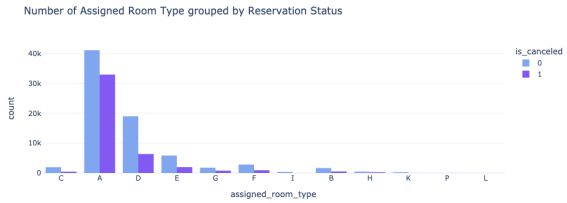
H_0 **:** there exists no correlation

H_1: there exists a correlation

The p-value chosen for this test was 0.05. The Chi-squared test yielded a value that is very close to zero, which is 1.121956218424043e-133. Since this value is <0.05, the null hypothesis was rejected and there exists a correlation between this feature variable and the label.

20. Assigned Room Type Feature Analysis

This feature variable tries to capture the customer behavior in the case where a different room type was assigned than the reserved room type.



Looking at the above barplot, it seems that there are two new room types "I" and "K", which experience no reservations but are assigned to guests.

assigned_room_type	A	B	C	D	E	F	G	H	I	K	L	P
is_canceled												
0	41105	1651	1929	18960	5838	2824	1773	461	358	267	0	0
1	32948	512	446	6362	1968	927	780	251	5	12	1	12

The cancellation rate of the assigned room types ranges from 19% - 44.5 %, with room type A having the highest cancellation rate of 44.5 % and room type C having the lowest cancellation rate of 19%. However, there are some extreme cases which deviate from the above percentage range. Rooms I and K which were not reserved and were only assigned to respective guests experienced a very low cancellation rate of around 2% and 4%, respectively. Even though these rooms were not desired by the guests as their first preference, however, being assigned to these rooms had a very little impact on their cancellation behavior. Moreover, it can be seen that rooms L and P experience a cancellation rate of 100%. It only makes sense to analyze whether the

assignment of different rooms than the ones reserved has a direct impact on the cancellation behavior. Out of the 14917 of reserved guests that were assigned different rooms than the one they booked, 802 of them canceled their bookings. Being assigned a different room than the one reserved leads to a cancellation rate of around 5% which is not that significant but could be also considered a reason as to why guests might cancel their bookings.

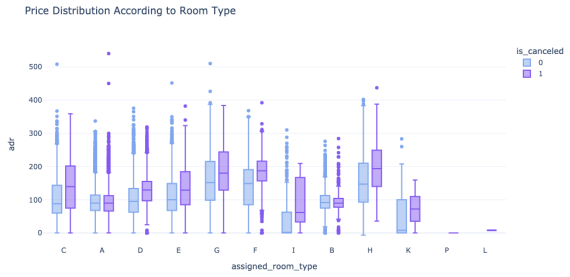
A chi-square test was performed to be able to determine the existence of a correlation between this feature variable and the label variable. The following Hypotheses were tested:

H_0 **:** there exists no correlation

H_1: there exists a correlation

The p-value chosen for this test was 0.05. The Chi-squared test yielded a value that is very close to zero, which is 0.0. Since this value is <0.05, the null hypothesis was rejected and there exists a correlation between this feature variable and the label.

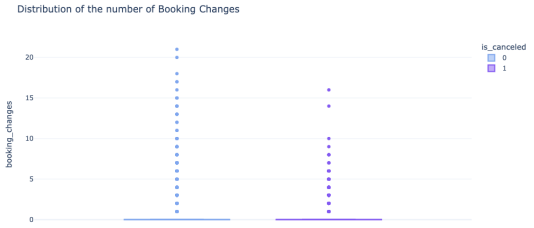
In an attempt to identify the reasons behind the high cancellation rates that some reserved room types face, the distribution of the average daily rate of each room type was taken a closer look at.



It becomes evident from the graph above, that on average the average daily rate of rooms that are canceled is higher than the average daily rate of rooms that were fulfilled. So higher average rate prices result in higher cancellation rates.

20. Booking Changes Feature Analysis

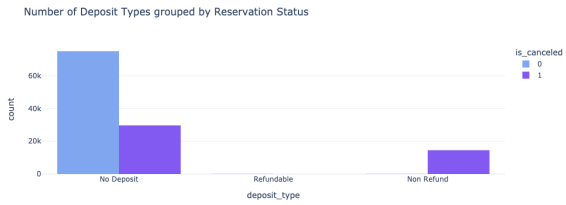
This feature variable tries to monitor the booking cancellation potential based on the number of booking changes made until the booking was fulfilled or canceled.



The distributions above are left skewed, as the majority of reservations are done without any booking changes. Since both distributions are almost identical, one might infer that there is no direct impact of the booking changes on the customer's behavior to cancel or not. Even though the maximum number of booking changes of one instance reached 21, yet this client did not cancel their booking. It is important to note that the customer type with the most booking changes on average are customers of type transient and transient party. The correlation between this feature variable and the label variable is around -0.14438099106132654 which shows a weak negative relationship.

21. Deposit Type Feature Analysis

This dataset captures reservations that do not have any deposit requirements, are refundable and are not refundable. This feature variable tries to observe whether the payment of any kind of deposit retains the guest or not.



The most common deposit type in this dataset is the "no deposit" type, followed by the "non refundable" and the "refundable" type.

deposit_type	No Deposit	Non Refund	Refundable
is_canceled			
0	74947	93	126
1	29694	14494	36

The cancellation rate based on the deposit type is as follows: The highest cancellation rate (67%) is of

deposit type no deposit, followed by deposit type non-refundable (32.7%) and deposit type refundable (0.08%). Although one might assume that non-refundable booking reservations would experience low cancellation rates compared to others because of lost money, it can be concluded that a big portion of guests are willing to cancel their booking with non-refundable deposits.

A chi-square test was performed to be able to determine the existence of a correlation between this feature variable and the label variable. The following Hypotheses were tested:

H_0 **:** there exists no correlation

H_1: there exists a correlation

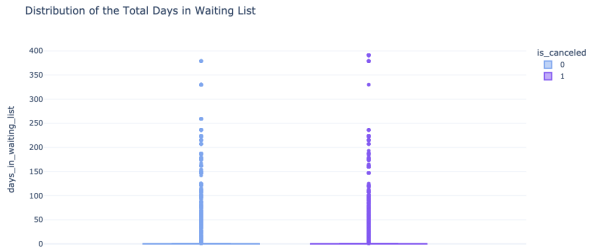
The p-value chosen for this test was 0.05. The Chi-squared test yielded a value that is very close to zero, which is 0.0. Since this value is <0.05, the null hypothesis was rejected and there exists a correlation between this feature variable and the label.

22. Days in Waiting List Feature Analysis

This feature variable tries to depict whether the number of days a guest has to wait in until his reservation is confirmed has a direct impact on his/her cancellation decision or not. The statistical distribution of the variable shows that on average the number of days that guests wait on the waiting list is higher for canceled reservations than for reservations that were fulfilled.

is_canceled	count	mean	std	min	25%	50%	75%	max
0	75166.0	1.589868	14.784875	0.0	0.0	0.0	0.0	379.0
1	44224.0	3.564083	21.488768	0.0	0.0	0.0	0.0	391.0

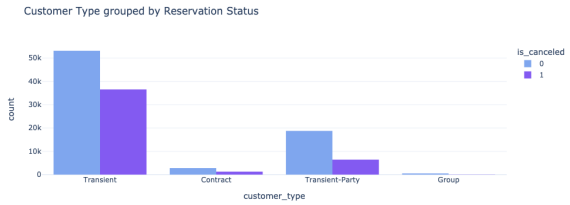
One might infer that as the number of days on the waiting list increases, the reservation is more likely to be canceled. However, looking at the distributions below, such a conclusion is not applicable because both distributions are almost identical.



Taking a closer look into the observations that have high waiting times, it becomes clear that the months (April, May, June, July, and August) have the highest waiting times on average and are therefore considered as high-season. However, the number of days on the waiting list does not have a direct impact on the cancellation behavior. The correlation between this feature variable and the label variable is around 0.054185824117780675 which shows a weak positive relationship.

23. Customer Type Feature Analysis

This feature variable categorizes the type of hotel guests in groups in an attempt to capture customer behavior based on their type. The most common customer type in this dataset are of type transient, which are individuals that need a short stay in a hotel, followed by transient-party customers, contracts and lastly groups. The graph below shows that customers of type transient are the most important and most frequent customer groups and they comprise the most important target group for hotels.



customer_type	Contract	Group	Transient	Transient-Party
is_canceled				
0	2814	518	53099	18735
1	1262	59	36514	6389

According to the above contingency table, the customer group with the highest cancellation rate is transient guests with a cancellation rate of 82%, followed by transient-party customers with a cancellation rate of 14.4% and Contract customers with the lowest cancellation rate of 0.133%. It becomes clear that customer types that are individuals not associated with a contract or a group tend to cancel their reservations more easily and hence have higher cancellation rates.

A chi-square test was performed to be able to determine the existence of a correlation between this feature variable and the label variable. The following Hypotheses were tested:

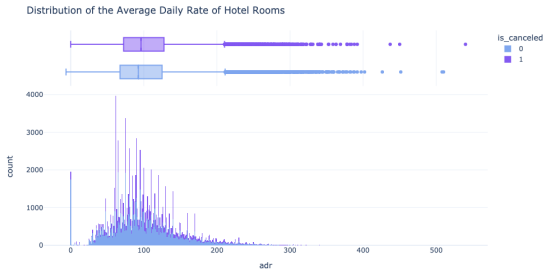
H_0 **:** there exists no correlation

H_1: there exists a correlation

The p-value chosen for this test was 0.05. The Chi-squared test yielded a value that is very close to zero, which is 0.0. Since this value is <0.05, the null hypothesis was rejected and there exists a correlation between this feature variable and the label.

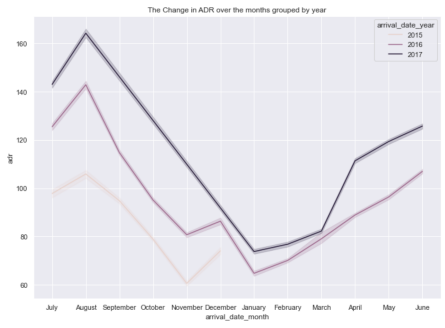
24. Average Daily Rate (ADR) Feature Analysis

This feature variable provides knowledge about the different price rates of the rooms reserved and whether it has a direct impact on hotel booking cancellations or not. The statistical distribution of the ADR variable, depicted below, is left skewed and shows that the average daily rate of reservations that were canceled are higher than those who were fulfilled. The median of the average daily rate of canceled reservations lies around 96.2, while the median of the average daily rate of fulfilled reservation lies around 92.5. The mean ADR of fulfilled bookings lies around 99.987693, while the mean ADR of canceled bookings lies around 104.854438.



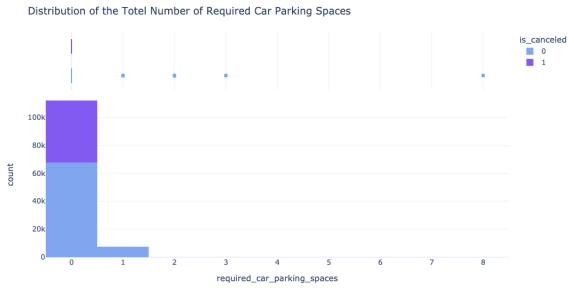
Moreover, the below line plot shows that the ADR increases over the years (2015-2017). Bookings made during the months during summer (April, May, June, July, and August) experience the highest rate relative to the whole year, which makes them a high season. The ADR experiences a huge drop in January.

The correlation between this feature variable and the label variable is around 0.04879048029938 which shows a weak positive relationship.



25. Required Car Parking Spaces Feature Analysis

This feature variable shows the number of required car parking spaces for each reserved room. The unique values in this categorical variable are 0, 1, 2, 3, and 8. More than 94% of the guests do not require any car parking spaces.

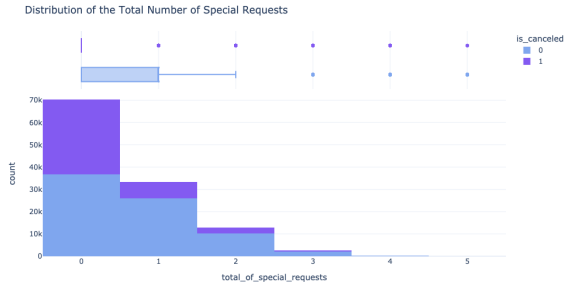


As can be seen from the graph above, the distribution of the variable is left skewed, with the majority of observations, requesting no car parking spaces regardless of them canceling or not. It is evident that there exists two major outliers which represent two reservations that were fulfilled and that requested around 8 car parking spaces. This will be further looked into to try and discover the reason behind it. It can be seen that the guests who requested 8 car parking spaces were of type "Transient-Party". Transient-Party are transient guests that are associated with at least another transient booking and could be viewed as a group that require a short stay at a hotel. Moreover, the above boxplot assumes that as the number of car parking spots required increases, the booking is more likely to be fulfilled. However, this phenomenon is not generalizable as it is inferred from a very small number of guests that could be considered as an exception. Therefore, this variable might not be of huge importance in predicting hotel booking cancellations. The correlation between this feature variable and the label variable is around -0.19549781749449852 which shows a weak negative relationship.

26. Number of Special Requests Feature Analysis

This variable gives insight about the total number of special requests made by each guest and its impact on the reservation status. If we compare the distribution of the total number of special requests based on the reservation status, it becomes clear that the distributions are almost identical, which means that

this feature does not have a huge role in determining any potential cancellations.



It is evident that the distribution of the variable is left skewed. The majority of the reservations did not ask for any special requests. However, the portion of fulfilled reservations have on average more special requests in total. Moreover, there are 3 outliers shown by the boxplots of reservations having 3, 4 and 5 special requests. The correlation between this feature variable and the label variable is around -0.2346577739690245. There exists a weak negative relationship between the total number of special requests and the cancellation status.

27. Reservation Status Feature Analysis

This feature variable shows the reservation status upon the time of expected check-out. There are 3 reservation status, which are canceled, check-out and no-show.

reservation_status	Canceled	Check-Out	No-Show
is_canceled			
0	0	75166	0
1	43017	0	1207

It is evident that there are guests that do not show up on the day of the bookings, which have a huge impact on the revenue management system. It is important to discover the motives behind the customer's no show decision. In an attempt to understand the reason behind no shows, the following table was constructed.

deposit_type	reservation_status	
No Deposit	Canceled	28522
	Check-Out	74947
	No-Show	1172
Non Refund	Canceled	14460
	Check-Out	93
	No-Show	34
Refundable	Canceled	35
	Check-Out	126
	No-Show	1

The relationship between the deposit type and the reservation status suggests that the highest no-shows occur in reservations done without any deposit requirements, followed by refundable deposits. This indicates that as long as the guest did not pay any form of commitment to the hotel, he/she does not feel obliged to cancel the reservation, which further pressures the revenue management system. Moreover, both the "canceled" and the "No-Show" classes are classified in the label variable "is_canceled" as canceled.

A chi-square test was performed to be able to determine the existence of a correlation between this feature variable and the label variable. The following Hypotheses were tested:

H_0 **:** there exists no correlation

H_1: there exists a correlation

The p-value chosen for this test was 0.05. The Chi-squared test yielded a value that is very close to zero, which is 0.0. Since this value is <0.05, the null hypothesis was rejected and there exists a correlation between this feature variable and the label.

Data Cleaning and Preprocessing

The data set was cleaned in a way that makes it more understandable and clear. Inconsistent values, missing data and categorical variables were dealt with appropriately.

1. Preprocessing the Feature Variable Hotel

The feature hotel is a categorical variable with two classes (resort and city hotels). It seemed appropriate to encode it using one-hot encoding and transforming it to a binary variable (0=resort and 1=city).

2. Preprocessing the Feature Variable Country

As mentioned before, the feature country had 488 missing values and 177 unique values. The number of missing values is insignificant since the data set encompasses exactly 119,390 observations. The most appropriate course of action is to fill in the missing values with the mode of the feature. The mode was the country Portugal ("PERT"). The missing values were replaced with the value ("PERT"). In this data set, most guests are Portuguese. The hotels in this data set are located in Portugal, so we can infer that most guests at these hotels are local customers.

In order to solve the problem of the unique values, one-hot encoding was used. Two new features were created: the first one “Portugal” is a binary variable indicating whether the guest is from Portugal (1) or not (0) and the second “International” indicating whether the guest is from another country (1) or not (0). The column country was subsequently dropped. It was found that local Portuguese customers have the highest cancellation rate (56%). The reason behind this high cancellation rate is that the no-deposit type. This type of deposit is the highest among Portuguese customers. The no-deposit type has high cancellation rates.

3. Preprocessing the Feature Variable Lead Time

The numerical variable lead time was normalized using Minimum and Maximum scaling.

4. Preprocessing the Feature Variable Arrival Month

The feature is a categorical variable with 12 classes (the 12 months of the years). In order to facilitate analysis, the months were transformed into numerical variables with respect to the correct order of months in a year (e.g: January is 1 and June is 6).

5. Preprocessing the Features Variables Arrival Date Week Number and Day of Month

The aforementioned variables have insignificant correlation coefficients with the label. In addition, the number of unique values of each feature is significant. The arrival year and month are much more significant (in terms of correlation coefficient and Chi-Square test) and provide more insights on the customer’s arrival date. These variables will therefore be dropped.

6. Preprocessing the Features Variable Stays in Weeknights and Weekend Nights

As previously mentioned the two features both have an insignificant correlation coefficient with the label . The heat map of the variables in this data set underscored a collinearity problem between the two features. An appropriate course of action would be eliminating them both. Instead of deleting both of these variables at the risk of losing significant data, a new feature called “total_stay” will be created combining the aforementioned variables. The feature engineering section of the report explains the significance of this new variable in detail. The arrival date, week number and day of month were therefore dropped.

7. Preprocessing the Feature Variable Adult

All observations with no adults (=0) were removed because it is impossible for a child/baby (a minor) to book a reservation at the hotel.

8. Preprocessing the Feature Variable Children

The variable children had four missing values which is an insignificant number given the size of the data. The observations were removed.

9. Preprocessing the Feature Variable Meal Type

The meal type is a categorical variable with four classes. It seemed appropriate to encode them and separate them into four different columns. Four new binary columns were subsequently created and the main column meal type was dropped.

10. Preprocessing the Feature Variable Market Segment

The feature market segment is a nominal variable with 6 categories. The six categories were encoded using the one-hot encoding technique. Six new binary variables representing the different categories of this feature were added to the data set. The original variable was subsequently dropped.

11. Preprocessing the Feature Variable Distribution Channel

Distribution channel is a categorical variable with 5 categories. Just like most of the nominal variables in this data set, one-hot encoding was applied to discretize the different classes of this feature. Five new binary features each representing a category of this variable were added to the data set. The original variable was subsequently dropped.

12. Preprocessing the Feature Variable Reserved Room Type and Assigned Room Type.

Both the reserved room type and the assigned room type variables are categorical variables with 10 and 12 categories, respectively. Due to the various classes in each variable, one-hot encoding would result in sparsity. Therefore, label encoding was applied to discretize the different classes of the feature. For the assigned room type variable the following encoding was applied: {'C':0, 'A':1, 'D':2,

'E':3, 'G':4, 'F':5, 'I':6, 'B':7, 'H':8, 'P':9, 'L'"10, 'K':11} and for the reserved room type variable the following encoding was applied:{'C':0, 'A':1, 'D':2, 'E':3, 'G':4, 'F':5, 'H':6, 'L':7, 'P':8, 'B':9}.

13. Preprocessing the Feature Variable Deposit Type

The feature deposit type is a categorical variable with 3 categories. The three categories were encoded using the one-hot encoding technique. Three new binary variables representing the different categories of this feature were added to the data set. The original variable was subsequently dropped.

14. Preprocessing the Feature Variable Agent

This feature has more than 14% missing data. While trying to find a suitable technique for dealing with these missing values, it becomes evident that the numeric values are considered categorical as they represent the different IDs of the agents sitting on the front desks that are responsible for reservation verification. Moreover, the number of unique values in this column is huge and it would not make sense to replace the missing values with either the mean or the most frequent agent ID. Accordingly, this variable was dropped from the dataset.

15. Preprocessing the Feature Variable Company

Since more than 94% of the data in this column is missing, it won't be of any benefit in our prediction as it does not capture any information that is generalizable. Accordingly, this feature variable was dropped.

16. Preprocessing the Feature Variable Customer Type

The customer type variable is a categorical variable with 4 categories. Just like most of the nominal variables in this data set, one-hot encoding was applied to differentiate the different classes of this feature and ease future computations. Four new binary features each representing a category of this variable were added to the data set. The original variable was subsequently dropped.

17. Preprocessing the Feature Variable ADR

During the analysis process and while plotting the distribution of this variable, it became noticeable that there is a value of the average daily rate that is 5400, which is represented as a major outlier in the boxplot plot. The only explanation for this outlier could be a typo. Accordingly 5400 was changed to 540.

18. Preprocessing the Feature Variable Reservation Status

As mentioned above, both the "canceled" and the "No-Show" classes are classified in the label variable "is_canceled" as canceled. Accordingly, label encoding was used to map both "Canceled" and "No-Show" to the same label. (Check-Out=0, Canceled & No-Show=1). As a result of this label encoding, this feature variable became an exact copy of the label variable. Hence, this feature is redundant and was therefore dropped from the dataset.

Feature Engineering

The data was leveraged and new variables were created with the goal of simplifying the data and enhancing model accuracy.

The feature “kids”

Major booking websites such as Booking and Trivago do not differentiate between children and babies. Anyone under the age of 18 falls under the category of “kids”. This is why a new feature called “kids” was created. It is basically the sum of the number of children and babies for each observation. The columns babies and children were subsequently dropped. In addition, previous analysis has shown that both variables were not of significant importance to the prediction of hotel cancellation.

The feature “total_number_of_stay”

As previously mentioned, the feature variables stays in weekends and weekdays were removed due to collinearity problems and low correlation with the label. However, instead of eliminating both features at the risk of losing significant data, a new feature called “total_stay” will be created combining the aforementioned variables. This new feature is basically the sum of the number of stays during the weekends and weekdays. This new feature will allow us to simplify the data even further through creating a more meaningful feature .

The feature "total_guests"

Previous analysis has shown that the variables adults and kids do not play a significant role in determining hotel cancellation prediction. This is why adding a new feature combining the three aforementioned variables seems to be an appropriate solution to the problem. This new feature is the sum of the number of adults and kids under one booking. This new meaningful feature will enhance overall model accuracy by providing new insights.

Data Post-Cleaning

The new cleaned data set encompasses 118,339 observations and 37 columns.

Variables
hotel
is_canceled
lead_time
arrival_data_year
arrival_data_month
total_stays
adults
kids
HB
FB
BB
SC
Portugal
International
online TA
offline TA
groups
direct
corporate
complementary
aviation
TA
TO
GDS
previous_cancellation
previous_booking_not_ canceled
booking_changes
days_in_waiting_list
adr
required_car_parking_ spaces
total_of_special_requests
reserved_room_type
assigned_room_type
no_deposit
refundable
non_refundable
transient
transient-party
group
contract
total_guests
reserved_room_type

Hotel Booking Cancelation Prediction using Machine Learning Models

Pilot Study

Farida Simaika
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
faridasimaika@aucegypt.edu

Katia Gabriel
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
katiag@aucegypt.edu

Experimental Analysis of the Machine Learning Models

The aim of this project is to predict the hotel booking cancellations. As a first step to decide which machine learning model is applicable for implementation, the nature of the label must be observed. Since the dataset used for developing the respective machine learning model is a dataset with a binary label, it becomes a binary classification problem. Accordingly, the following models have been chosen to predict the hotel booking cancellations: Logistic Regression, K-Nearest Neighbours, Naive Bayes Classifier, Multiple Layer Perceptron Learning (Artificial Neural Network), Decision Trees (ID3, C 4.5), and Random Forests. All models will be developed, compared and the top 3 performing models will be picked for further hypertuning of the parameters.

All models were developed with the help of scikit-learn machine learning package in python.

Before the implementation of any model, the dataset was split into training and testing sets. 80% of the dataset is used for training, while the other 20% is used for testing.

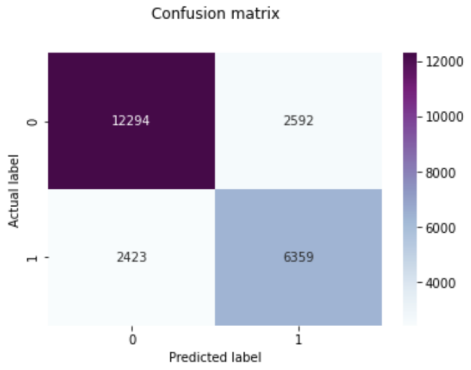
1. Logistic Regression

The Logistic regression model is suitable for the data at hand since it is generally used to predict a dependent categorical label based on the various independent feature variables. Since our label is binary, a binary logistic regression model has been trained to assess the likelihood of a cancellation occurring by a certain customer. Before fitting the model, the feature variables were standardized using a standard-scaler, as they are in different scales, to decrease the variability in the dataset. Moreover, during learning, a 5-fold-cross-validation was performed to mitigate the problem of overfitting and to ensure the model's generalization ability on unseen data. The following results were obtained:

```
Average cross validation score: 0.785
Test accuracy: 0.788
F1 score: 0.717
Classification Report :
              precision    recall  f1-score   support

           0       0.84      0.83      0.83     14886
           1       0.71      0.72      0.72      8782

    accuracy                           0.79     23668
   macro avg       0.77      0.77      0.77     23668
weighted avg       0.79      0.79      0.79     23668
```



The results show that the model has a 78.5% accuracy score. The portion of actual cancellations that were identified correctly is the most important metric in evaluating the model. Accordingly, The True Negative Rate (TNR) was calculated and has a value of 0.7240947392393532 ≈ 72%, which is also considerably high. While looking at the confusion matrix, it becomes evident that the model was able to correctly predict 82.5% of the label '0' and 72.4% of the label '1'. The misclassification error of this model is around 21%. The best possible set of weights W that were estimated by the model, which maximize the

likelihood of the labels of all the observations given feature variables are shown below:

```
intercept W_0 : [-0.54592446]
coefficients W_i: [[ 2.57640827e-01  3.16288950e+00  4.77301614e-01  1.54154987e-01
   3.79033334e+00 -8.43796481e-01  1.95911791e+01 -7.09941225e+00
  -5.59651054e+00 -1.67039918e+00  3.50970908e+00 -2.32041543e+01
  -2.79149013e+00  3.27256008e-01  4.19039265e+00  3.85712028e+00
  -2.29105923e-02 -3.84472956e-01 -1.57045056e-01 -7.87838660e-01
  -2.07573635e-02  6.28838056e-01 -4.72994532e-01 -1.20340682e-01
  -1.79611434e-01  7.22522140e-01 -8.87388095e-01  8.83077724e-01
  -2.64863407e-01 -2.13604573e-01 -5.69475699e-01 -1.74608430e+00
   2.53126189e+00 -9.50043551e-01  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00]]
```

Since the weights in the logistic regression equation measure the impact a unit change in the feature corresponding to the weight affects the likelihood that the label will be true or not, the list of weights depicted below confirm the following. The last 20 feature variables have a weight value of 0, which indicates that the last feature variables do not have an impact on whether a customer decides to cancel their reservation or not and are therefore redundant variables. The last 20 variables resemble the one-hot encoding columns for the variables "assigned_room_type" and "reserved_room_type", which means that these factors have no effect on the cancellation prediction. The feature variables corresponding to the weights with the highest impact are: "lead_time", "adults", "previous_bookings_not_cancelled", "booking_changes", "days_in_waiting", "adr", "special_requests", "total_number_of_stays", "total_guests", "total_bookings", and "BB".

The logistic regression model's results are very understandable and can be applied on the dataset. The set of estimated weight can intuitively show the ultimate combination of feature variables that best describe the label. However, in our dataset some features are often correlated with each other and thus the multicollinearity problem rises which needs to be dealt with and the model might need further tuning, accordingly . This might explain the accuracy score and give room for further tuning of the model.

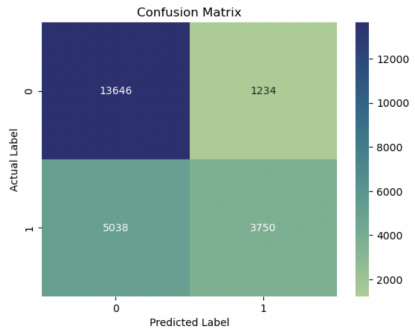
2. Naive Bayes

One of the strengths of the Naive Bayes classifier is that it works well with binary labels as it provides a probability score for each class of the label. This can be useful in hotel cancellation predictions where a high probability will most likely indicate a possible cancellation. This is why a Naive Bayes classifier was trained in order to calculate the probability of a cancellation given the input features. The classifier calculated the probability of each class of the binary label given the input features and predicted the class with the highest probability. Different performance measures were calculated in order to assess the performance of this classifier. The following results were obtained:

```
Average cross validation score: 0.688
Test accuracy: 0.686
F1 score: 0.666
Recall score: 0.839
Precision score: 0.552
Classification Report :
              precision    recall  f1-score   support

           0       0.86      0.59      0.70     14841
           1       0.55      0.84      0.67      8827

    accuracy                           0.69     23668
   macro avg       0.71      0.72      0.68     23668
weighted avg       0.75      0.69      0.69     23668
```



The implementation of the Naive Bayes classifier yielded an accuracy score of 69% which is relatively low compared to other machine learning models. The accuracy score measures the proportion of correctly classified samples out of the total number of samples in the data. The Naive Bayes Classifier correctly classified 69% of the samples which is relatively low compared to other machine learning models. This low accuracy score might be due to the fact that the Naive Bayes classifier assumes conditional independence of each feature given the label. However, in real-world scenarios and in our dataset some features are often correlated with each other and thus the assumption of independence does not hold true. This might explain the low accuracy score of this model. The specificity (True Negative Rate) was calculated to be approximately 84.39%. The Naive Bayes classifier was able to correctly predict 84.39% of canceled hotel reservations. The above output also indicates that the precision for label "0" (not canceled bookings) is 0.86. Approximately 86% of the

reservations predicted not to be canceled were actually not canceled. Similarly, the precision for label “1” (canceled reservations) is 0.55, which means that 55% of the predicted canceled reservations were truly canceled.
It has been shown that the Naive Bayes classifier works best for predicting reservations that are not canceled (label “0”). The algorithm has a low accuracy score and thus does not meet the project’s goals.

3. Decision Trees

Decision trees are a powerful machine learning algorithm that can be used for a wide range of classification tasks. Each leaf node in a decision tree represents a class label or a predicted value, whereas each internal node represents a decision based on a feature. The tree is constructed by iteratively dividing the data into subgroups according to the optimal feature that maximizes the Information Gain or Gini Index. The most crucial features that affect the prediction may be automatically identified by decision trees which enhances the model's effectiveness and overall accuracy.

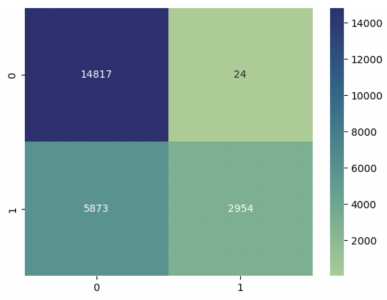
ID3 Algorithm:

ID3 is a decision tree algorithm used for classification tasks. At each internal node, the features that maximize the information gain and minimize the entropy are used to build the tree. Each leaf node indicates a class label. The ID3 algorithm was implemented on the data set and the results were as follow:

```
Average cross validation score: 0.749
Test accuracy: 0.751
F1 score: 0.500
Classification Report :
              precision    recall  f1-score   support

           0       0.86      0.91      0.88     14841
           1       0.83      0.75      0.79      8827

    accuracy                           0.85     23668
   macro avg       0.85      0.83      0.84     23668
weighted avg       0.85      0.85      0.85     23668
```



The ID3 classifier has an accuracy score of approximately 85% meaning that the model is able to predict the correct class label for 85% of the testing data. The classification report shows that the model has high precision and recall for class label “0” (not canceled reservation) and lower precision and recall for class label “1” (canceled reservations), which indicates that the model is better at predicting class label “0” than class label “1”. Using the confusion matrix, the proportion of correctly identified cancellations (the specificity- True Negative Rate) was calculated to be approximately 33% which is relatively low compared to other machine learning algorithms.
Despite the decent accuracy score, the ID3 algorithm does not fulfill the project’s objective which is correctly predicting cancellations (label “1”). The ID3 algorithm is better at predicting reservations that are not canceled. Another problem with the ID3 algorithm is its greedy approach, it selects the features that maximizes the information gain and thus might result in suboptimal splits that reduce the model's overall accuracy.

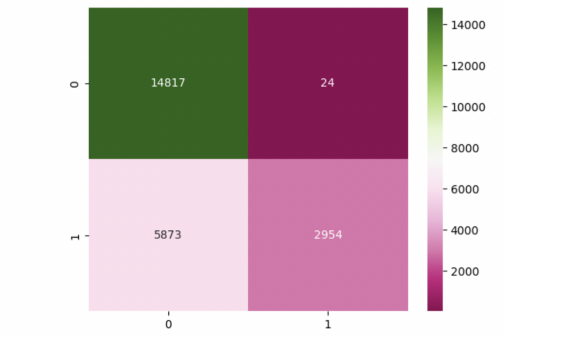
CART Algorithm:

The CART decision tree is an extension of the ID3 algorithm. It is also an improved version of the ID3 decision tree. The CART algorithm can create a more generalized model than ID3 that includes continuous variables. CART constructs binary trees using the feature and threshold that yield the largest information gain and smallest Gini Index at each node. The CART algorithm was implemented on the data set and the results were as follow:

```
Average cross validation score: 0.749
Test accuracy: 0.751
F1 score: 0.500
Classification Report :
              precision    recall  f1-score   support

           0       0.86      0.91      0.88     14841
           1       0.83      0.75      0.79      8827

    accuracy                           0.85     23668
   macro avg       0.85      0.83      0.84     23668
weighted avg       0.85      0.85      0.85     23668
```



The CART classifier has an accuracy score of approximately 75% meaning that the model is able to predict the correct class label for 75% of the testing data. The classification report shows that the model has high precision and recall for class label "0" (not canceled reservation) and lower precision and recall for class label "1" (canceled reservations), which indicates that the model is better at predicting class label "0" than class label "1". The proportion of actual cancellations that were classified correctly was calculated. Accordingly, The True Negative Rate (TNR) has a value of 33% which is also considerably low.

The CART algorithm has a decent accuracy score however it still does not fulfill the project's objective which is accurately predicting the canceled reservations. This might be due to the fact that the data at hand is unbalanced. The CART algorithm might produce biased trees if the data set is unbalanced. In addition, the CART algorithm is prone to overfitting and instability, meaning that small changes in the data can lead to significant changes in the tree structure.

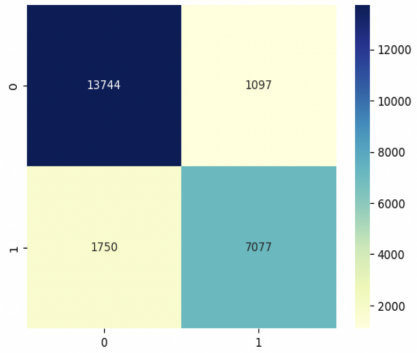
Random Forest

Random Forest decision trees combine the output of multiple decision trees to improve predictive performance. Different decision trees are trained using randomly selected subsets of the training data. A collection of distinct and unrelated decision trees are produced as a result and are subsequently combined to give a final prediction.

```
Accuracy Score of Random Forest is : 0.8797110021970593
F1 score: 0.833
Classification Report :
              precision    recall  f1-score   support

           0       0.89      0.93      0.91     14841
           1       0.87      0.80      0.83      8827

    accuracy                           0.88     23668
   macro avg       0.88      0.86      0.87     23668
weighted avg       0.88      0.88      0.88     23668
```



The Random Forest algorithm was applied on the data and achieved an accuracy score of 88%. The Random Forest algorithm was able to successfully predict the label of almost 88% of the data. The specificity score (the True Negative Rate) was calculated using the confusion matrix to be around 81% which is relatively high considering that the portion of actual cancellations that were identified correctly is the most important metric in evaluating the model. The misclassification error is around 12%. The classification report also shows that the model has a slightly higher precision for label "0" (not canceled reservations) than label "1". The model is slightly better at predicting the label "0" (canceled reservations) than the label "1".

In general, the Random Forest model is one of the best performing models given the nature of our data. One of the advantages of Random Forest is that it is resistant to overfitting due to the combination of multiple decision trees. It fulfills the project's goal which is accurately predicting the canceled reservations. However, the model can be computationally expensive and difficult to interpret compared to other decision trees such as ID3 and CART algorithm.

5. Artificial Neural Network
(Multiple Layer Perceptron)

Artificial Neural Networks are based on the idea of a perceptron, in which they take the feature variables with random weights assigned to them as its input, apply an activation function and try to determine the output. In the learning process, the weights keep updating themselves until the optimal set of weights that minimizes the error is achieved. Various activation functions are used in the ANN to determine the best performing model of them all.

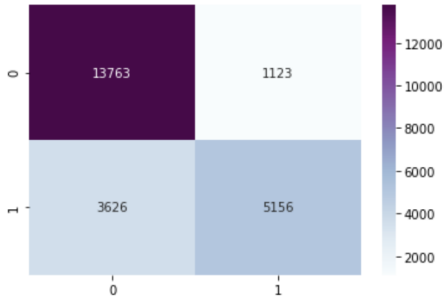
Identity Function as Activation Function

The following results were obtained for a multiple layer perceptron trained with the identity function acting as the activation function for the ANN. The following model consists of an input layer, 20 neurons in one hidden layer and an output layer. The weights are learned and updated using a stochastic gradient descent. The following results were obtained:

```
Train score: 0.796
Test accuracy: 0.799
F1 score: 0.685
Classification Report :
              precision    recall  f1-score   support

           0       0.79      0.92      0.85     14886
           1       0.82      0.59      0.68      8782

    accuracy                           0.80     23668
   macro avg       0.81      0.76      0.77     23668
weighted avg       0.80      0.80      0.79     23668
```



The results show that the overall accuracy of this model is 79.9% . The specificity score is around 60% which is relatively low considering that the portion of actual cancellations that were identified correctly is the most important metric in evaluating the model. The misclassification error is around 20%. This model has high precision and low recall for class label label "1" (canceled reservations), which indicates that whenever the model flags an observation as "canceled" it is a credible decision, however the model is better at predicting class label "0" than class label "1".

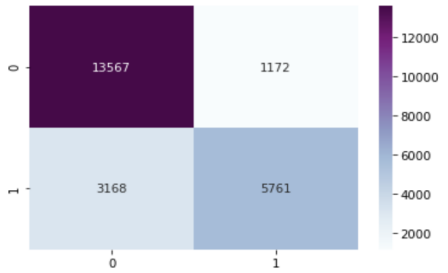
Sigmoid Function as Activation Function

The following results were obtained for a multiple layer perceptron trained with the sigmoid function acting as the activation function for the ANN. The following model consists of an input layer, 47 neurons in the first hidden layer and 20 neurons in the second hidden layer and an output layer. The weights are learned and updated using a stochastic gradient descent. The following results were obtained:

```
Train score: 0.821
Test accuracy: 0.817
F1 score: 0.726
Classification Report :
              precision    recall  f1-score   support

           0       0.81      0.92      0.86     14739
           1       0.83      0.65      0.73      8929

    accuracy                           0.82     23668
   macro avg       0.82      0.78      0.79     23668
weighted avg       0.82      0.82      0.81     23668
```



The results show that the overall accuracy for this model is around 82%, which is considerably high. The specificity score was calculated and turned out to be 0.6421771754955762 ≈ 64%. Meaning that the model can predict 64% of the cancellations correctly. The misclassification error is around 18%. The classification report shows that the model has a higher precision score of 83% for class label "1" (canceled reservations) in comparison to the class label "0" (not canceled reservations), which indicates that whenever this model predicts an observation to be a potential cancellation instance it is actually correct.

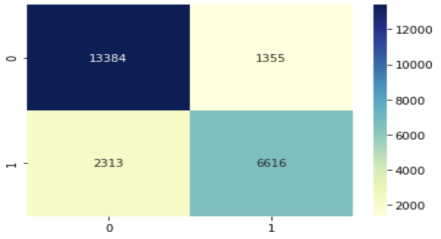
Hyperbolic Tangent as Activation Function

The following results were obtained for a multiple layer perceptron trained with the hyperbolic tangent function acting as the activation function for the ANN. The following model consists of an input layer, 20 neurons in the one hidden layer and an output layer. The weights are learned and updated using a stochastic gradient descent. The following results were obtained:

```
Train score: 0.852
Test accuracy: 0.845
F1 score: 0.783
Classification Report :
              precision    recall  f1-score   support

           0       0.85      0.91      0.88     14739
           1       0.83      0.74      0.78      8929

    accuracy                           0.85     23668
   macro avg       0.84      0.82      0.83     23668
weighted avg       0.84      0.85      0.84     23668
```



The results show that the overall accuracy for this model is around 85%, which is considerably high. The specificity score was calculated and turned out to be 0.7283010415500056 ≈ 73%, meaning that the model can predict 73% of the cancellations correctly, which is considerably high. The misclassification error is around 15%. The classification report shows that this model has a higher precision score and recall score than the rest of the ANN models so far.

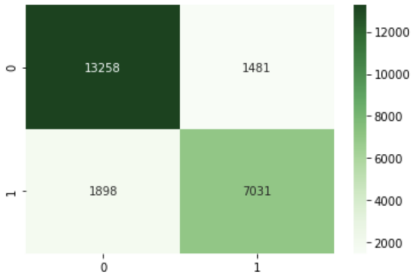
Rectified Linear Unit Function as Activation Function

The following results were obtained for a multiple layer perceptron trained with the rectified linear unit function acting as the activation function for the ANN. The following model consists of an input layer, 100 neurons in the first hidden layer, 50 neurons in the second hidden layer, 10 neurons in the third hidden layer and an output layer. The weights are learned and updated using a stochastic gradient-based optimizer "adam". The following results were obtained:

```
Train score: 0.891
Test accuracy: 0.857
F1 score: 0.806
Classification Report :
              precision    recall  f1-score   support

           0       0.87      0.90      0.89     14739
           1       0.83      0.79      0.81      8929

    accuracy                           0.86     23668
   macro avg       0.85      0.84      0.85     23668
weighted avg       0.86      0.86      0.86     23668
```



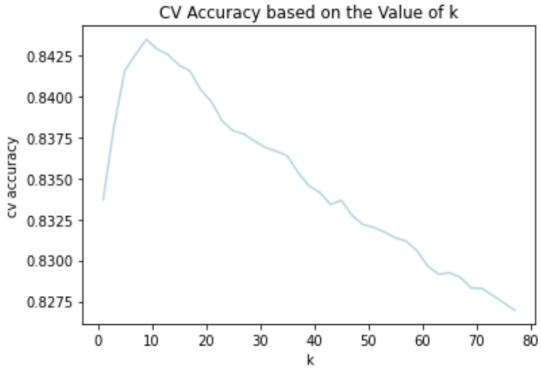
The results show that the overall accuracy for this model is around 89%, which is considerably high. The specificity score was calculated and turned out to be 0.7874342031582484 ≈ 79%, meaning that the model can predict 79% of the cancellations correctly, which is considerably high. While looking at the confusion matrix, it becomes evident that the model was able to correctly predict 89% of the label '0' (not canceled reservations) and 78% of the label '1'(canceled reservations).The misclassification error is around 14%.

In general it is evident that the artificial neural network has performed relatively well on the dataset at hand, however, these models are computationally expensive and are hard to visualize and explain. The choice of parameters is hard and there is no straightforward way of choosing it, e.g. how many hidden layers to choose for the network, how many neurons for each hidden layer, etc. Moreover, the amount of weights that have been generated for each arc is a lot and is therefore very hard to visualize, compare, and associate with the feature variables to determine which features are most important in predicting potential hotel cancellations. By looking at all ANN models with the various activation functions, it is evident that the best performing model out of all of them is the ANN with the rectified linear unit function as its activation function. However, this model will be disregarded due to the fact that ANN models are computationally complex and memory intensive. In addition, ANN models lack transparency

(difficult to interpret) and are very sensitive to initialization. To function well, ANNs need large datasets. This is due to the fact that they encompass numerous parameters that must be optimized during training. The ANN model might not be able to uncover the underlying correlations and patterns if the dataset is too small.

6.K-Nearest Neighbors (KNN)

This machine learning model is based on instance based supervised learning. The distance between each observation is calculated. Based on that the k nearest neighbors are identified to classify any new instance based on the majority of votes of the surrounding neighbors identified. The distance function used in developing the model is the euclidean distance. Before fitting the model, the feature variables were standardized using a standard-scaler, as they are in different scales, which would, if not scaled, affect the euclidean distance measure and the results will be biased towards observations that have by nature smaller values. Moreover, during learning, a 5-fold-cross-validation was performed to mitigate the problem of overfitting and to ensure the model's generalization ability on unseen data. In order to choose the best k, the model was trained on a range of k-values from 1 to $\sqrt{n}$, excluding the even numbers and taking the odd numbers only. Since the range of numbers is huge and is both computationally and timely expensive, a subset consisting of the odd numbers between 1-77 were chosen and iterated over. For every k value, the accuracy score, the classification report, the confusion matrix, and the AUC were computed and lastly the ROC curve was plotted. The results obtained were summarized in the following graph:

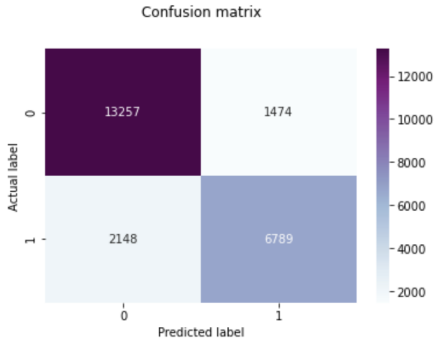


The graph above depicts the 5-fold-cross validation accuracy score as k increases. The graph has low accuracy at k=1 and the accuracies keep on increasing to reach their maximum at k=9 and then it decreases as k increases. It is evident that the k value with the highest cv-accuracy score is k=9. The model was trained on k=9 and the following results were obtained:

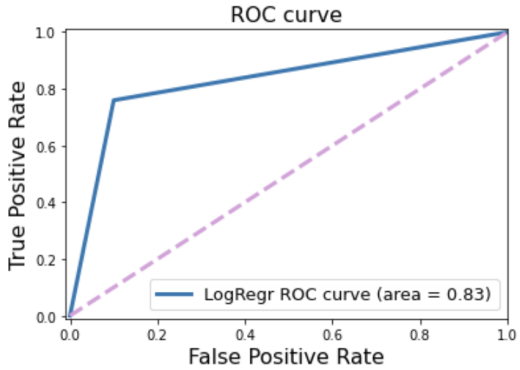
```
For k =  9
accuracy score for training =  0.84347773423471004
accuracy score for testing =  0.8469663680919385
error rate in prediction =  0.15303363190806152
              precision    recall  f1-score   support

           0       0.86      0.90      0.88     14731
           1       0.82      0.76      0.79      8937

    accuracy                           0.85     23668
   macro avg       0.84      0.83      0.83     23668
weighted avg       0.85      0.85      0.85     23668
```



The results show that the model accuracy with k=9 is around 85% . The specificity score was calculated and turned out to be 0.759651 ≈ 75%, meaning that the model can predict 75% of the cancellations correctly, which is considerably high. While looking at the confusion matrix, it becomes evident that the model was able to correctly predict 90% out of the label '0' (not canceled reservations) and 76% out of the label '1'(canceled reservations).The misclassification error is around 15%.



The ROC curve depicts the change in TPR against the FPR for various values of k. Since the curve is shifted towards the top left corner, this indicates that the model's performance is good and the area under the curve quantifies the graph as its value is 0.83 which underlines the fact that the model is performing good.

The 9-NN model is very intuitive and is easy to understand, evaluate and validate. However, the process of choosing the best performing parameter, the choice of k, that maximizes the accuracy without falling in the trap of overfitting, is hard, subjective and time consuming, especially that the data at hand is large. However, the 9-NN model suits the data at hand and is one of the best performing models so far.

Model Selection

The aim of the project is to accurately predict canceled hotel reservations. Many factors were considered upon selecting the most appropriate model for the dataset at hand. A thorough experimentation with a variety of machine learning models was conducted to determine the best fit for our data. It has been shown that the KNN, Logistic Regression and Random Forest models are the most suitable for our objective of accurately predicting cancellations. The three aforementioned models have decent accuracy scores and considerably high predictive power.

K-Nearest Neighbors (KNN) is a simple and intuitive algorithm that works well for smaller datasets. However, its performance decreases as the dataset size increases due to the difficulty of choosing parameters that maximize its predictive power. In addition, the KNN algorithm is memory intensive making it suitable only for small to medium sized datasets. The cost of classifying a new instance using KNN can be very high since the algorithm iterates through all training instances. It is evident that the disadvantages of using the KNN classifier outweigh its advantages. This is why the model will be disregarded.

Logistic regression is particularly well-suited for this dataset because it can model the likelihood of the reservation being canceled based on the values of the input variables. However, in our dataset some features are often correlated with each other. It might be challenging to determine the precise impact of each input variable on the target variable when there is collinearity. The coefficient estimates might be biased and imprecise. In order to deal with the multicollinearity problem, the model might need further tuning. However, the logistic regression model will not be disregarded as its benefits outweigh its disadvantages. Logistic Regression is computationally efficient while being simple and easily interpretable. One of the advantages of the Logistic Regression classifier is its ability to handle noise and identify the most important features.

Finally, the Random Forest algorithm is one of the best performing models and is the most well-suited for our data. One of the advantages of Random Forest is that it is resistant to overfitting due to the combination of multiple decision trees. It fulfills the project's goal which is accurately predicting the canceled reservations. It has a high predictive power and is able to capture complex interactions between the features. However, the model has some drawbacks: it can be computationally expensive and difficult to interpret compared to other models.

In conclusion, after thorough analysis and experimentation, it is clear that the Random Forest Algorithm is the most suitable model for hotel cancellation prediction.

Hotel Booking Cancelation Prediction using Machine Learning Models

Model Design

Farida Simaika
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
faridasimaika@aucegypt.edu

Katia Gabriel
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
katiag@aucegypt.edu

Model Design and Fine-Tuning

The aim of this project is to accurately predict hotel booking cancellations. A thorough experimentation with a variety of machine learning models was previously conducted. It has been shown that the Logistic Regression and Random Forest models were the best fit for our goal of accurately predicting canceled reservations.

Both models presented decent recall scores. As previously shown, the recall is the most important performance measure given our objective of correctly predicting cancellations. In other words, the portion of hotel bookings that are correctly identified as belonging to the class of interest (the positive class with label "1" indicating a cancellation) is the most important metric in evaluating a model.

Both the Logistic Regression and Random Forest classifiers were trained using the default hyperparameters as a baseline. It was previously shown that both models were slightly better at predicting hotel reservations that were not canceled (class with label "0") .

Classification Report of Logistic Regression model:

```
Average cross validation score: 0.785
Test accuracy: 0.788
F1 score: 0.717
Classification Report :
              precision    recall  f1-score   support

           0       0.84      0.83      0.83     14886
           1       0.71      0.72      0.72      8782

    accuracy                           0.79     23668
   macro avg       0.77      0.77      0.77     23668
weighted avg       0.79      0.79      0.79     23668
```

Classification report of the Random Forest model:

```
Accuracy Score of Random Forest is : 0.8797110021970593
F1 score: 0.833
Classification Report :
              precision    recall  f1-score   support

           0       0.89      0.93      0.91     14841
           1       0.87      0.80      0.83      8827

    accuracy                           0.88     23668
   macro avg       0.88      0.86      0.87     23668
weighted avg       0.88      0.88      0.88     23668
```

Using default hyperparameters, the Logistic Regression and Random Forest classifier have a decent accuracy score of 79% and 87% respectively. However this accuracy score is misleading because the dataset is imbalanced. There are 74,250 instances with label "0" (not canceled) and 44,088 instances with label "1" (canceled reservations). It is evident that both classifiers are correctly predicting the majority class (class with label "0"). The majority class is predominant making the accuracy automatically high.

The recall of both classifiers for the class of interest (the positive class indicating canceled reservations) is slightly lower than the recall of the other class that represents the bookings that were not canceled. In the hotel business, an inaccurate estimate of the number of guests might result in an overstaffing or understaffing of personnel and insufficient supplies, which can result in huge financial losses for the hotel. It is now evident that both classifiers and the dataset require further fine-tuning and pre-processing in order to improve their performance.

Logistic Regression Model Fine-Tuning

One way to tackle the imbalanced label distribution of the dataset at hand is to adjust the class weights. We could use the inverse of the label distribution.

The dataset has a label distribution of 63:37. For the majority class (label "0"), a weight of 37 will be used and the minority class (label "1") will be assigned a weight of 63. The penalty of wrong prediction of minority class (label "1") would be 63 times more severe than wrong classification of majority class (label "0"). Using these class weights, we will increase the weight (and thus the importance) of the class of interest with label "1" (canceled hotel reservations). This way we could compensate for the imbalance between the classes. By modifying the weights of the classes, we would expect a drop in the accuracy score and an increase in the recall of the class of interest (label "1").

Using the default hyperparameters, some of the weights of the Logistic Regression equation had values of 0. The weights in the logistic regression equation measure the impact of a unit change in the feature corresponding to the weight. The weights affect the likelihood of a label being true or not.

```
intercept W_0 : [-0.54592446]
coefficients W_i: [[ 2.57640827e-01  3.16288950e+00  4.77301614e-01  1.54154987e-01
   3.79033334e+00 -8.43796481e-01  1.95911791e+01 -7.09941225e+00
  -5.59651054e+00 -1.67039918e+00  3.50970908e+00 -2.32041543e+01
  -2.79149013e+00  3.27256008e-01  4.19039265e+00  3.85712028e+00
  -2.29105923e-02 -3.84472956e-01 -1.57045056e-01 -7.87838660e-01
  -2.07573635e-02  6.28838056e-01 -4.72994532e-01 -1.20340682e-01
  -1.79611434e-01  7.22522140e-01 -8.87388095e-01  8.83077724e-01
  -2.64863407e-01 -2.13604573e-01 -5.69475699e-01 -1.74608430e+00
   2.53126189e+00 -9.50043551e-01  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00]]
```

As shown above, the last twenty feature variables have weight values of 0. These features do not affect whether a booking will be canceled or not. These features are the one-hot encoded variables for each class of the variables "assigned_room_type" and "reserved_room_type". We can further preprocess the dataset and drop these variables to reduce the dimensionality of the dataset. The Logistic Regression classifier will be trained without these feature variables to see if there are any improvements regarding the performance metrics.

In order to meet the project's objectives, we will need to further tune the hyperparameters of the Logistic Regression Classifier. An appropriate and logical choice of the hyperparameters will improve the performance of the classifier.

Using the library *sklearn*, the main hyperparameters that require tuning are the solver, the penalty and the log loss function.

The solver is the algorithm that will be used in the optimization problem. Each solver tries to find the parameter weights that minimize a cost function. According to the *sklearn* documentation, the solver "lbfgs"(Limited-memory-Broyden–Fletcher–Goldfarb–Shanno) seems to be the most appropriate for the dataset at hand. The solver is not memory-intensive and is time-efficient (since it uses an approximation of the Hessian Matrix) when implemented on large datasets. The "newton_cg" solver is also a suitable optimization algorithm that uses a quadratic function minimization. However, it is computationally expensive (so not applicable on large datasets) as it uses an exact Hessian matrix so it computes the second derivative.

The penalty parameter aims to decrease the model generalization error and limits overfitting. Some penalties do not work with some solvers. The L2 penalty term is the only one that works with the "newton_cg" and "lbfgs" solvers. The L2 regularization limits overfitting by decreasing the weights. This means that the less significant features in predicting a cancellation would have smaller weights (close to 0 but non-zero). The sum of squares of all the feature weights is added to the loss function when implementing L2 (Ridge) regularization.

Finally, the other hyperparameters that optimize the model's predictions will be found using the Grid Search function. These parameters are "max_iter" (maximum number of iterations taken for the solver to converge) and C (inverse of regularization strength). The Grid Search function will be implemented using 5-fold cross-validation.

Finally, the log loss function (loss function of Logistic Regression) will allow us to heavily penalize the predictions that deviate from their true class.

Fine-tuning the hyperparameters of the logistic regression classifier will allow us to find the optimal combination that will minimize the loss function. Upon implementation of all the aforementioned changes, the recall function of the class of interest should increase. An overall improvement of the different performance measures of the model (precision, recall, accuracy and f1) should also be seen.

Random Forest Model Fine-Tuning

Our aim is to find the optimal set of hyperparameters that will improve the performance of the Random

Forest model. The Random Forest classifier has many parameters that require refining in order to maximize the classifier's performance and minimize loss.

The Random Forest Classifier will be trained on the data using a for loop that will iterate over a range of numbers to ensure that the optimal hyperparameters are found. The main hyperparameters that require tuning are the tree's depth, the number of leaf nodes, the number of decision trees, the number of features to consider when looking for the best split, the criterion that evaluates the quality of a split and the minimum number of samples required to split an internal node.

The maximum depth of the tree is the longest path between the root node and the leaf node. This hyperparameter needs to be optimized in order to prevent the tree from learning all the training instances and thus overfitting. A for loop will iterate over a range of numbers and at each iteration the training and accuracy scores will be printed. This will allow us to detect overfitting (when the validation accuracy decreases and training accuracy starts increasing).

The number of leaf nodes (maximum nodes of each tree) will also be refined. The number of leaf nodes helps in controlling the classifier's complexity. According to the *sklearn* documentation, the classifier splits the node with the lowest Gini Index. A for loop iterating through different values of the number of leaf nodes will be implemented. The value that will yield to decent performance measures and validation scores will be selected.

The number of decision trees also requires optimization. The number should be kept minimal in order to reduce dimensionality and complexity of the classifier. The training and validation scores will be calculated for each value of the number of decision trees. We will try to select the lower number of trees that provide the best results.

The number of features to be considered for each tree needs to be optimized in order not to end up with the same trees. This will allow the generalization of the Random Forest classifier. According to the *sklearn* documentation, we will either be using the square root or the log of the total number of features at each split. The number of features that yields decent training and validation scores without overfitting will be selected.

The Random Forest classifier will be trained using different criterions (Gini Index, Log loss and Entropy). The criterion is a function that measures the quality of a split. The selection of the criterion will depend on the performance of the classifier and its training and accuracy scores.

Finally, tuning the minimum number of samples required to split an internal node is essential to avoid overfitting and having a model that does not generalize well. This is why a testing of the different hyperparameters will be conducted. The values that yield the best performance measures without overfitting will be chosen.

Bootstrapping will be enabled so that the classifier does not use the whole data set to build each tree.

Finally, the other hyperparameters that optimize the model's predictions will be found using the Grid Search function. The Grid Search function will be implemented using 5-fold cross-validation.

The ideal combination of hyperparameters will eventually be found via fine-tuning. The recall of the class of interest (canceled reservations label "1") should improve when all the aforementioned adjustments have been made. The model's various performance metrics (precision, recall, accuracy, and f1) should all show improvement.

Model Selection:

Many factors were considered upon selecting the most appropriate model for the dataset at hand. A thorough experimentation with both the Logistic Regression and Random forest classifiers was conducted to determine the best fit for our data. The two aforementioned models have a considerably high predictive power and decent recall scores for the class of interest (label "1": canceled reservations). The Logistic Regression model was selected despite the fact that the Random Forest classifier had a slightly higher recall score for the class of interest.

One of the advantages that the Logistic Regression Classifier has over the Random Forest model is that it is computationally efficient when implemented on large datasets. Unlike the Random Forest algorithm, the Logistic Regression classifier is simple and easily interpretable. The Logistic Regression classifier is easily able to handle noise and identify the most important features. The coefficients of the Logistic Regression model can be used to understand the impact of each feature on the probability of

cancellation. Additionally, logistic regression can handle both categorical and continuous variables, making it a versatile model for hotel cancellation prediction.

On the other hand, the Random forest classifier is computationally expensive when dealing with large datasets. Training a Random Forest model and deploying it on an application can be especially hard with a high number of trees. In addition, the Random Forest classifier is more difficult to understand and interpret than Logistic Regression. The implementation of a Random Forest model requires the fine-tuning of a significant number of hyperparameters (number of trees, depth, minimum number of samples required) in order to reach an optimal combination. There is not always a clear rationale behind the fine-tuning process; there may not be a clear understanding of how each hyperparameter affects the model's performance. The process requires trial and error and is therefore computationally expensive.

In conclusion, after thorough analysis and experimentation, it is clear that the Logistic Regression Classifier is the most suitable model for hotel cancellation prediction.

Utility Application Design

After hypertuning the parameters and reaching the model that fits our data best and has the best scores, it is essential to be able to provide this machine learning model to customers and users to make use of it. Accordingly, a web-based application will be developed to provide users with a user-friendly interface, where they can input their datasets/ instances to be predicted and apply the machine learning algorithms to generate the respective predictions. The application will be designed in a way that will guide the users through the model and provide the prediction of new instances. The application will be designed using the python package called *Streamlit*. Streamlit turns the python code (backend) which has the machine learning algorithm into an application that can be accessed through a generated url (frontend). The python code will be used to generate the design of the application's interface as well as run the machine learning algorithm to display the needed information to the user. To ensure that the model does not assume that the model is on the same machine, the source code and the application will be deployed on the *Streamlit Cloud* and the client will communicate the feature with it through an API and get the prediction as a response. The application will provide a Logistic Regression Classifier. The objective of this application is to predict instances using a Logistic Regression Classifier. Accordingly, the architecture of the application will be designed as follows:

1. User Input

At first the user will be asked to input the information related to the instance that needs prediction. All the feature variables will be displayed and the user will be obligated to fill in the data for each feature variable for one single instance.

2. Data Preprocessing

After inputting the data for the instance needed for prediction, the feature variables will undergo data preprocessing. All categorical variables will go through a one-hot encoding which is crucial to ensure the same model learning and performance. In the case of missing data, if a numerical variable was expected to be inputted, the value of this feature variable will be replaced by the mean of this variable in the training dataset. However, if a categorical variable was expected to be inputted, an all-zeros vector will be generated using one-hot encoding to indicate the unknown value. Moreover, the whole observation will be scaled to ensure that no variable having relatively larger values dominates the results and creates biasness. The scaling will be done by standardizing the range of the feature variables, by subtracting the sample mean and dividing by the standard deviation of the variables in the training set. Afterwards, the instance will be displayed to the user after applying the respective data preprocessing. The data is now ready to be inputted into the model.

3. Machine Learning Model and Its Parameters

As a next step, the application will display the machine learning model that has been chosen, which is the Logistic Regression Classifier. Moreover the user will be guided through the parameters that have been hypertuned and chosen, yielding the best results. All the chosen values for the parameters will be displayed and explained clearly. The parameters that are crucial for the Logistic Regression, as discussed above, are the solver, the penalty and the log loss

function. The weights for each variable that has been obtained during training will be displayed to give the user an idea of which features are crucial and play a dominant role in predicting whether a hotel booking will be canceled or not. Furthermore, the confusion matrix, the classification report, and the ROC curve will be also displayed, to inform the user of the model's performance and ensure that the model was properly trained.

4. Hotel Cancellation Prediction

After identifying the model and displaying its parameters, the instance will be then inputted to the model for prediction. The application will output only one of two numbers either 0 or 1. The results will then be explained to the user, identifying that the "0" means that this instance will not cancel the booking and "1" means that this instance will cancel the booking. According to the output, human intervention is needed to decide what to do with this instance in the hotel revenue management system.

5. Application Reset and User Input

By predicting the instance inputted by the user, the application terminates, restarts and asks the user to input new data for a new instance to be predicted.

Hotel Booking Cancelation Prediction using Machine Learning Models

Model and Utility Application Implementation

Farida Simaika
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
faridasimaika@aucegypt.edu

Katia Gabriel
Department of Mathematics and Actuarial Science
The American University in Cairo
Cairo, Egypt
katiag@aucegypt.edu

Logistic Regression Model Fine Tuning

The aim of this project is to accurately predict hotel booking cancellations. In the hotel business, an inaccurate prediction of the number of customers might result in an overstaffing or understaffing of personnel and inadequate supplies. This, in turn, may lead to significant financial losses for the hotel. The developed machine learning model will become a tool that will empower hotel managers in optimizing overbooking restrictions and in better forecasting an accurate net revenue. Accordingly, the hotel industry will be able to capture important customer behavioral patterns that might be the root cause behind hotel booking cancellations.

A thorough experimentation with a variety of machine learning models was previously conducted. It has been shown that the Logistic Regression classifier is the best fit for our goal of accurately predicting canceled reservations. In terms of accurately predicting canceled reservations, the model demonstrated satisfactory recall scores for the class of interest (canceled booking reservations). The recall is the most crucial performance metric given the project's objective.

As previously mentioned, the dataset has an imbalance label distribution. There are 74,250 instances with label “0” (not canceled) and 44,088 instances with label “1” which is the class of interest (canceled reservations). A cost-sensitive learning approach was implemented in order to tackle the problem of imbalanced label distribution. The objective function of the model was changed and optimized to penalize the misclassification of canceled bookings higher than the non-canceled hotel reservations. In other words, a higher cost was assigned to misclassifying a canceled hotel reservation as non-canceled as opposed to misclassifying a non-canceled booking as canceled. In order to carry out this cost-sensitive approach, weighted logistic regression was implemented. A weight was assigned to each observation based on its class. A higher weight was assigned to the class of interest (canceled bookings). The higher weight will allow the model to learn from canceled bookings despite their relative rarity in the dataset. The optimal weights that maximize the model’s performance were found using the GridSearchCV function. A for loop iterating over a wide range of weights was implemented. It was also specified that the recall (correctly predicting cancellations) was the “scoring” parameter to be optimized. Using 5-fold cross validation, the optimal weights that maximize the model’s recall for the class of interest were found.

Hyperparameters Tuning

A combination of GridSearchCV and cross-validation was implemented in order to find the hyperparameters that maximize the model’s predictive power for the class of interest (canceled bookings). A dictionary with a wide range of values for each parameter was defined in order to find the optimal combination of hyperparameters.

Inverse Regularization Strength Parameter (C)

The regularization parameter is a technique used to prevent the machine learning model from overfitting and therefore improving its generalization. A value of 0.1 was found to be the optimal parameter. This value indicates that the model will be moderately regularized during training to prevent overfitting.

Number Of Iterations

The number of iterations hyperparameter controls how many times the model goes through the entire

dataset during the training process. The number of iterations should be high enough to ensure model convergence to the optimal solution without overfitting. The optimal number of iterations was found to be around 500.

Solver

The solver is used to optimize the loss function of the model during training. The scikit-learn library provides several solvers to choose from, including “liblinear”, “lbfgs”, “sag”, and “saga”. Using the function GridSearchCV, the “liblinear” best fits our data. This solver can handle both L1 and L2 penalties. It uses a coordinate descent algorithm that iteratively updates the coefficients by solving a one-dimensional optimization problem for each feature.

Penalty

The penalty is a regularization term added to the loss function of the model that helps prevent overfitting by shrinking the coefficients towards zero. The penalty parameter aims to decrease the model generalization error. The GridSearchCV iterated over the L1 and L2 penalties. The L1 penalty was found to be the optimal penalty given the nature of our dataset.

Final Model Evaluation

```
This best score is : 0.7943636978198412
Classification Report :
              precision    recall  f1-score   support

           0       0.84      0.83      0.83     14741
           1       0.72      0.74      0.73      8927

    accuracy                           0.79     23668
   macro avg       0.78      0.78      0.78     23668
weighted avg       0.80      0.79      0.79     23668
```

The performance metrics of the hypertuned model for the class of interest (label 1: canceled reservations) have significantly improved. The results show that the fine tuned model has almost an 80% accuracy score. The portion of actual cancellations that were identified correctly is the most important metric in evaluating the model. The recall of the class of interest is considerably high at almost 75%. The cost sensitive learning approach has significantly improved the metrics of the class of interest despite its relative rarity in the dataset.

Utility Application Implementation

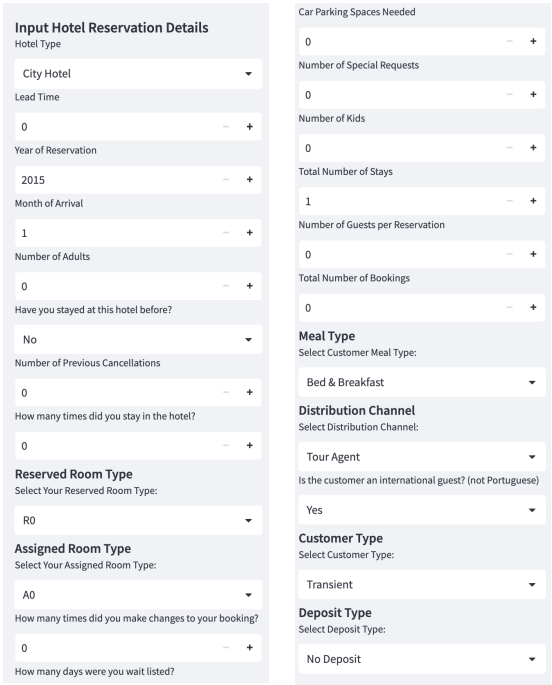
Finally, the machine learning model that best fits the nature of the data and achieves this project's objective has been implemented and is ready to be used by hotel managers. However, not all managers are acquainted with machine learning model algorithms and are able to understand the implementation of various machine learning techniques. Therefore, it is of huge importance to be able to provide users with a user-friendly interface, and mitigate the gap between the management and the complexity of the implementation of the model so that they can make use of such a powerful tool. Accordingly, a web application has been developed that guides the users through the model and provides them with the prediction of new imputed instances. In the following course of this paper the implementation of the web application and the steps to be taken by the user will be thoroughly discussed:

Application Implementation

The application is designed using the python package called *Streamlit*. Streamlit turns the python code (backend) which has the machine learning algorithm into an application that can be accessed through a generated url (frontend). The python code will be used to generate the design of the application’s interface as well as run the machine learning algorithm to display the needed information to the user. The application will provide a Logistic Regression Classifier. The objective of this application is to predict instances using a Logistic Regression Classifier.

User Experience: Data Entry

As a first step the user will be provided with a web link through which the application can be accessed. After opening the application, the user will be asked to input 26 data entries related to the instance that needs prediction. All the feature variables will be displayed and the user will be asked to fill in the data for each feature variable for one single instance to be predicted as can be seen below:



As the user is inputting the data in the required fields, all feature variables will undergo data preprocessing whenever it is needed. All categorical variables will go through a one-hot encoding which is crucial to ensure the same model learning and performance.

User Experience: Running the Model

After the user is done with inputting the needed information, the user will be urged to click the "run" button, which will trigger the model to use the instance given to predict its label.

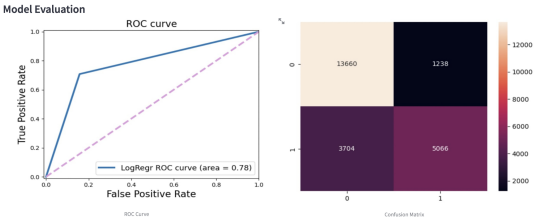
Upon clicking the "run" button, the application will display the machine learning model that has been chosen, which is the Logistic Regression Classifier. Moreover the user will be guided through the parameters that have been hypertuned and chosen, yielding the best results. All the chosen values for the parameters will be displayed and explained clearly. The parameters that are crucial for the Logistic Regression, as discussed above, are the regularization strength, the solver, the number of iterations and the

penalty.

User Experience: Model Output

After identifying the model and displaying its parameters, the instance will be then inputted to the model for prediction. The application outputs a percentage which stands for the probability of cancellation. Low percentage means that this booking represented by the instance inputted by the user will most likely be fulfilled and not canceled. However, a higher percentage means that this booking will most probably be canceled. According to the output, human intervention is needed to decide what to do with this instance in the hotel revenue management system. An example of the output of the model is displayed below:

Furthermore, the confusion matrix, the classification report, and the ROC curve will be also displayed, to inform the user of the model's performance and ensure that the model was properly trained.



User Experience: Application Reset

By refreshing the browser's page, the application will reset and the user will be able to input the data of the next instance.

Conclusion

After the completion of this project, it became evident that machine learning models are very powerful and can help a lot of industries in taking better informed decisions based on the predictions provided. Forecasting the future can drive businesses to better prepare for the future, adjust their strategies, quantify their risks and eventually maximize their profits. Especially in the hotel industry which relies

solely on future events, machine learning techniques are to be used by the management to achieve a competitive advantage but also to maximize revenue and minimize booking cancellations.