Hotel Bookings Prediction Project - Final Report

HarvardX: PH125.9x; Data Science Capstone

Marwa J. Nafakh
8-24-2020

Report Outline

Introduction 2
- I. Dataset Description 2
- II. Goal of the Project 2

Methods and Analysis 3
- I. Data cleaning 4
- II. Data Exploration and Visualization 5
  - 1) Exploring the structure of the hotel_data dataset 5
  - 2) Understanding Booking Cancellation Behavior 11
- III. Insights Gained 20
- IV. Create Data Partitions for training and validation purposes 22
- V. Modelling Approach 23
  - 1) Logistic Regression Model 23
  - 2) Classification Tree Model 24
  - 3) Random Forest Model 25

Results 27

Conclusion 28
- I. Summary of the report 28
- II. Limitation and Future Work 28

Introduction

Being able to accurately predict future hotel booking cancellation has a great impact on business management and revenue generation. Therefore, applying the science of data to build models for prediction is highly demanded by business owners and managers, and has direct and tangible impact on running the business efficiently and effectively.

In this project, several machine learning algorithms were developed based on testing and validating the models on the 'hotel bookings' dataset. Three different data models: logistic regression, classification tree, and random forest were selected to predict future booking cancellation based on the characteristics of the collected bookings data.

I. Dataset Description

The Hotel Booking dataset is publicly available data that has been collected over a three years period. The data has 119390 observations and 32 columns which include booking characteristics ranging from guests' meal selection to required parking space, number of accompanying children and many other factors affecting the prediction of cancellation likelihood/probability.

Data is available through the link: https://www.kaggle.com/jessemostipak/hotel-booking-demand/

For the purpose of this course project, the Hotel Booking dataset was explored, analyzed, and modeled. It has been subdivided into smaller datasets to ease the process of analysis and modelling.

1. The hotel_train dataset (training subset) has been created for the purpose of finding the optimal prediction algorithm representing 90% of the Hotel Booking dataset. Hotel_train dataset consisted of 107451 observations and 7 factors.
2. The hotel_valid dataset (testing/validation subset) representing 10% of the Hotel Booking dataset, used for validation purposes only acted upon by the generated data models to find the optimal model with the highest accuracy value.

II. Goal of the Project

This project aims at building a prediction algorithm based on cancelled hotel reservations to be able to predict future cancellation taking into consideration seven different factors affecting the prediction algorithm. Validation of the selected machine learning algorithm is ensured through the validation dataset. The evaluation criterion of the generated models is the accuracy metrics.

Methods and Analysis

Hotel Booking dataset was downloaded through Kaggle data repository. This project and all its related files are available at:

https://github.com/MarwaJN/CYO-Project.git

The data set was extensively explored and analyzed and then divided into a test and train subsets to generate the machine learning algorithms. Hotel_valid (test set) dataset was used for model validation purposes assuming that this dataset is unknown to measure the accuracy and select the best model based on the highest accuracy value. Through cross validation method of the predicted cancellations against the actual cancellations in the validation test set.

The following steps were performed on the hotel_data dataset leading to proper definition of relationships between predictors and hence gaining an insight for developing an effective algorithm with the highest accuracy as the evaluation criteria of the models.

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2")
if(!require(gridExtra)) install.packages("gridExtra")
if(!require(dplyr)) install.packages("dplyr")
if(!require(rpart)) install.packages("rpart")
if(!require(rpart.plot)) install.packages("rpart.plot")
if(!require(rattle)) install.packages("rattle")
if(!require(randomForest)) install.packages("randomForest")
if(!require(corrplot)) install.packages("corrplot")
if(!require("e1071")) install.packages("e1071")
if(!require("class")) install.packages("class")
```

Workspace of the project was linked to a github repository to ensure proper version control and ease of access to the source file.

```
hotel_data<-read.csv("hotel_bookings.csv")
str(hotel_data)
```

```
## 'data.frame':    119390 obs. of  32 variables:
##  $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
##  $ is_canceled                   : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
##  $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ arrival_date_month            : chr  "July" "July" "July" "July" ...
##  $ arrival_date_week_number      : int  27 27 27 27 27 27 27 27 27 27 ...
##  $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ stays_in_weekend_nights       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
##  $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
##  $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ meal                          : chr  "BB" "BB" "BB" "BB" ...
##  $ country                       : chr  "PRT" "PRT" "GBR" "GBR" ...
##  $ market_segment                : chr  "Direct" "Direct" "Direct" "Corporate" ...
##  $ distribution_channel          : chr  "Direct" "Direct" "Direct" "Corporate" ...
##  $ is_repeated_guest             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ reserved_room_type            : chr  "C" "C" "A" "A" ...
##  $ assigned_room_type            : chr  "C" "C" "C" "A" ...
##  $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
##  $ deposit_type                  : chr  "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
##  $ agent                         : chr  "NULL" "NULL" "NULL" "304" ...
##  $ company                       : chr  "NULL" "NULL" "NULL" "NULL" ...
##  $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ customer_type                 : chr  "Transient" "Transient" "Transient" "Transient" ...
##  $ adr                           : num  0 0 75 75 98 ...
##  $ required_car_parking_spaces   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_of_special_requests     : int  0 0 0 0 1 1 0 1 1 0 ...
##  $ reservation_status            : chr  "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
##  $ reservation_status_date       : chr  "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...
```

In order to further understand the data two columns were added to calculate the total nights and total cost per stay per customer:

```
# Calculating total nights stayed at hotel for each customer in a new column
hotel_data <- hotel_data %>% mutate(total_nights = stays_in_weekend_nights + stays_in_week_nights)

# Calculating total total cost of stay for each customer in a new column
hotel_data <- hotel_data %>% mutate(total_cost = adr * total_nights)

# Check the added two columns
head(hotel_data)
```

I. Data cleaning

The hotel_data dataset has been checked for any missing value and returned TRUE, so missing values were identified, located, and replaced. Also, all character variables were converted to factors to ease the process of exploration and analysis. The data set was tidy since each row had one observation and columns stated the features for each observation.

```r
# Convert characters variables into factors for further analysis
hotel_data <- hotel_data %>%
  mutate(
    hotel = as.factor(hotel),
    meal = as.factor(meal),
    arrival_date_year = as.factor(arrival_date_year),
    arrival_date_month = as.factor(arrival_date_month),
    country = as.factor(country),
    market_segment = as.factor(market_segment),
    distribution_channel = as.factor(distribution_channel),
    reserved_room_type = as.factor(reserved_room_type),
    assigned_room_type = as.factor(assigned_room_type),
    deposit_type = as.factor(deposit_type),
    agent = as.factor(agent),
    company = as.factor(company),
    customer_type = as.factor(customer_type),
    reservation_status = as.factor(reservation_status)
  )
```

```r
# Check for any missing value in the hotel_data dataset
any(is.na(hotel_data))
```

```
## [1] TRUE
```

```r
# Find any missing values in the dataset and return the column name
list_NA <- colnames(hotel_data)[apply(hotel_data, 2, anyNA)]
list_NA
```

```
## [1] "children"
```

```r
# Replace the missing values in the Children Column in the hotel_data dataset with the babies column value

missing_list <- length(hotel_data$children)
for (i in 1:missing_list){
  if(is.na(hotel_data$children[i]))
    hotel_data$children[i] <- hotel_data$babies[i]
}
```

II. Data Exploration and Visualization

In order to better understand the hotel_data dataset, the relationships between the various variables and to gain a clear insight of how to develop an effective prediction model algorithm, various data exploration methods took place.

1) Exploring the structure of the hotel_data dataset

```r
dim(hotel_data)
```

```
## [1] 119390     34
```

```r
summary(hotel_data)
```

```
##          hotel        is_canceled       lead_time   arrival_date_year
##  City Hotel  :79330   Min.   :0.0000   Min.   :  0   2015:21996
##  Resort Hotel:40060   1st Qu.:0.0000   1st Qu.: 18   2016:56707
##                       Median :0.0000   Median : 69   2017:40687
##                       Mean   :0.3704   Mean   :104
##                       3rd Qu.:1.0000   3rd Qu.:160
##                       Max.   :1.0000   Max.   :737
##
##  arrival_date_month arrival_date_week_number arrival_date_day_of_month
##  August :13877      Min.   : 1.00            Min.   : 1.0
##  July   :12661      1st Qu.:16.00            1st Qu.: 8.0
##  May    :11791      Median :28.00            Median :16.0
##  October:11160      Mean   :27.17            Mean   :15.8
##  April  :11089      3rd Qu.:38.00            3rd Qu.:23.0
##  June   :10939      Max.   :53.00            Max.   :31.0
##  (Other):47873
##  stays_in_weekend_nights stays_in_week_nights     adults
##  Min.   : 0.0000         Min.   : 0.0         Min.   : 0.000
##  1st Qu.: 0.0000         1st Qu.: 1.0         1st Qu.: 2.000
##  Median : 1.0000         Median : 2.0         Median : 2.000
##  Mean   : 0.9276         Mean   : 2.5         Mean   : 1.856
##  3rd Qu.: 2.0000         3rd Qu.: 3.0         3rd Qu.: 2.000
##  Max.   :19.0000         Max.   :50.0         Max.   :55.000
##
##     children           babies                 meal          country
##  Min.   : 0.0000   Min.   : 0.000000   BB       :92310   PRT    :48590
##  1st Qu.: 0.0000   1st Qu.: 0.000000   FB       :  798   GBR    :12129
##  Median : 0.0000   Median : 0.000000   HB       :14463   FRA    :10415
##  Mean   : 0.1039   Mean   : 0.007949   SC       :10650   ESP    : 8568
##  3rd Qu.: 0.0000   3rd Qu.: 0.000000   Undefined: 1169   DEU    : 7287
##  Max.   :10.0000   Max.   :10.000000                     ITA    : 3766
##                                                          (Other):28635
```

```
##       market_segment  distribution_channel is_repeated_guest
##  Online TA    :56477   Corporate: 6677      Min.   :0.00000
##  Offline TA/TO:24219   Direct   :14645      1st Qu.:0.00000
##  Groups       :19811   GDS      :  193      Median :0.00000
##  Direct       :12606   TA/TO    :97870      Mean   :0.03191
##  Corporate    : 5295   Undefined:    5      3rd Qu.:0.00000
##  Complementary:  743                        Max.   :1.00000
##  (Other)      :  239
##  previous_cancellations previous_bookings_not_canceled reserved_room_type
##  Min.   : 0.00000       Min.   : 0.0000                A      :85994
##  1st Qu.: 0.00000       1st Qu.: 0.0000                D      :19201
##  Median : 0.00000       Median : 0.0000                E      : 6535
##  Mean   : 0.08712       Mean   : 0.1371                F      : 2897
##  3rd Qu.: 0.00000       3rd Qu.: 0.0000                G      : 2094
##  Max.   :26.00000       Max.   :72.0000                B      : 1118
##                                                        (Other): 1551
##  assigned_room_type booking_changes       deposit_type        agent
##  A      :74053      Min.   : 0.0000   No Deposit:104641   9      :31961
##  D      :25322      1st Qu.: 0.0000   Non Refund: 14587   NULL   :16340
##  E      : 7806      Median : 0.0000   Refundable:   162   240    :13922
##  F      : 3751      Mean   : 0.2211                       1      : 7191
##  G      : 2553      3rd Qu.: 0.0000                       14     : 3640
##  C      : 2375      Max.   :21.0000                       7      : 3539
##  (Other): 3530                                            (Other):42797
##     company       days_in_waiting_list         customer_type
##  NULL   :112593   Min.   :  0.000      Contract       : 4076
##  40     :   927   1st Qu.:  0.000      Group          :  577
##  223    :   784   Median :  0.000      Transient      :89613
##  67     :   267   Mean   :  2.321      Transient-Party:25124
##  45     :   250   3rd Qu.:  0.000
##  153    :   215   Max.   :391.000
##  (Other):  4354
##       adr          required_car_parking_spaces total_of_special_requests
```

```
##  Min.   :  -6.38   Min.   :0.00000            Min.   :0.0000
##  1st Qu.:  69.29   1st Qu.:0.00000            1st Qu.:0.0000
##  Median :  94.58   Median :0.00000            Median :0.0000
##  Mean   : 101.83   Mean   :0.06252            Mean   :0.5714
##  3rd Qu.: 126.00   3rd Qu.:0.00000            3rd Qu.:1.0000
##  Max.   :5400.00   Max.   :8.00000            Max.   :5.0000
##
##  reservation_status reservation_status_date  total_nights      total_cost
##  Canceled :43017    Length:119390           Min.   : 0.000   Min.   : -63.8
##  Check-Out:75166    Class :character        1st Qu.: 2.000   1st Qu.: 146.0
##  No-Show  : 1207    Mode  :character        Median : 3.000   Median : 267.0
##                                             Mean   : 3.428   Mean   : 357.8
##                                             3rd Qu.: 4.000   3rd Qu.: 446.2
##                                             Max.   :69.000   Max.   :7590.0
##
```

```
class(hotel_data)
```

```
## [1] "data.frame"
```

```
paste('There are ',nrow(hotel_data),'rows', 'and ',
      ncol(hotel_data), 'columns in the hotel data dataset')
```

```
## [1] "There are  119390 rows and  34 columns in the hotel data dataset"
```

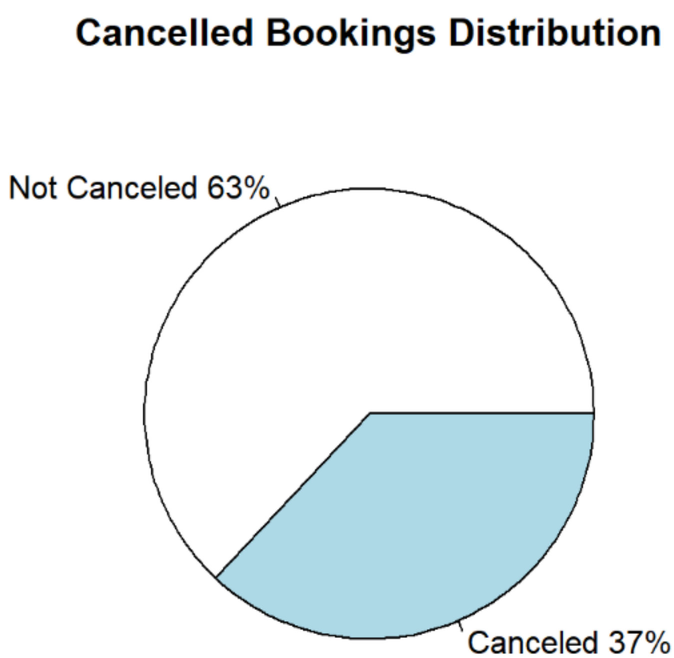
```
table(hotel_data$hotel)
```

```
##
##   City Hotel Resort Hotel
##        79330        40060
```

```
# It is noted that City Hotel had much more reservations than Resort Hotels
```

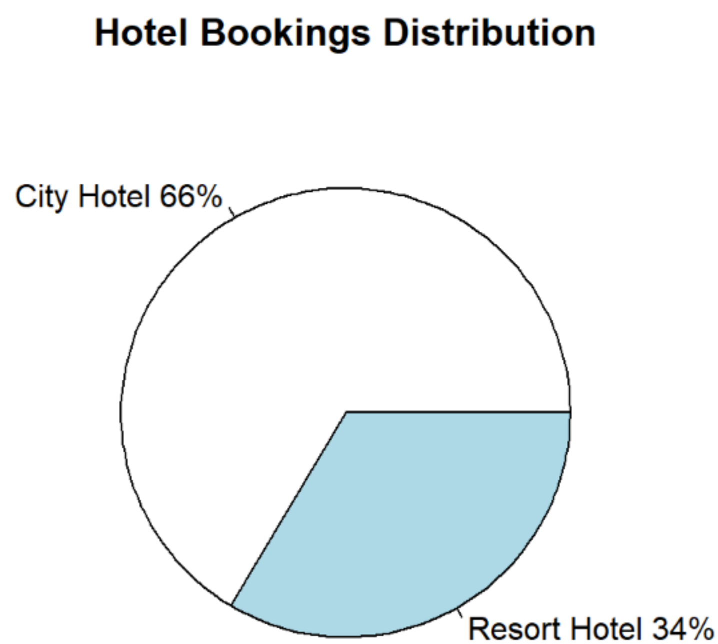
Also, data visuals were used to assist in understanding data:

```
hotel_pie <- table(hotel_data$is_canceled)
hotel_cancel <- c("Not Canceled", "Canceled")
percent <- round(hotel_pie/sum(hotel_pie)*100)
hotel_cancel <- paste(hotel_cancel,percent)
hotel_cancel <- paste(hotel_cancel,"%", sep="")
pie(hotel_pie, hotel_cancel, main = "Cancelled Bookings Distribution")
```

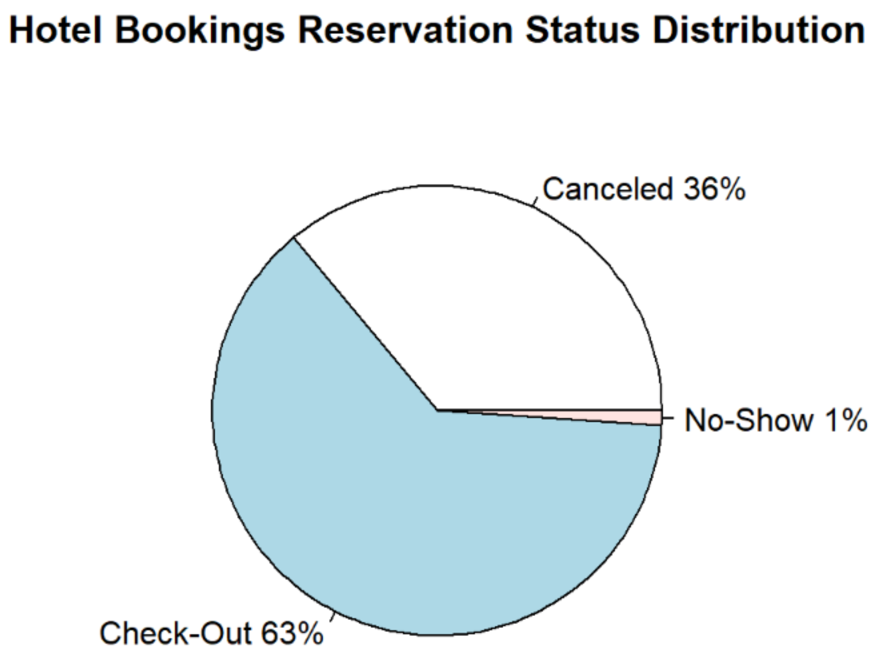


As shown 37% of the bookings were canceled.

```
hotel_pie <- table(hotel_data$hotel)
hotel_type <- names(hotel_pie)
percent <- round(hotel_pie/sum(hotel_pie)*100)
hotel_type <- paste(hotel_type,percent)
hotel_type <- paste(hotel_type,"%", sep="")
pie(hotel_pie, hotel_type, main = "Hotel Bookings Distribution")
```




```
hotel_pie <- table(hotel_data$reservation_status)
hotel_status <- names(hotel_pie)
percent <- round(hotel_pie/sum(hotel_pie)*100)
hotel_status <- paste(hotel_status,percent)
hotel_status <- paste(hotel_status,"%", sep="")
pie(hotel_pie, hotel_status, main = "Hotel Bookings Reservation Status Distribution")
```



It is noted that the status of the booking divided the booking into three different categories in which canceled bookings represented 36%.

Number of reservations for each city and resort hotels was listed: (where it showed Portugal had the highest number of bookings).

```
hotel_data %>% group_by(hotel,country)%>%
  summarize(No. = n())%>%
  arrange(desc(No.))
```

```
## # A tibble: 293 x 3
## # Groups:   hotel [2]
##    hotel        country   No.
##    <fct>        <fct>   <int>
##  1 City Hotel   PRT     30960
##  2 Resort Hotel PRT     17630
##  3 City Hotel   FRA      8804
##  4 Resort Hotel GBR      6814
##  5 City Hotel   DEU      6084
##  6 City Hotel   GBR      5315
##  7 City Hotel   ESP      4611
##  8 Resort Hotel ESP      3957
##  9 City Hotel   ITA      3307
## 10 Resort Hotel IRL      2166
## # ... with 283 more rows
```

Market segments were also analyzed showing that the highest number of reservations were booked through an agent for city hotels.

```
hotel_data %>% group_by(hotel, market_segment)%>%
  summarize(No. = n())%>%
  arrange(desc(No.))
```

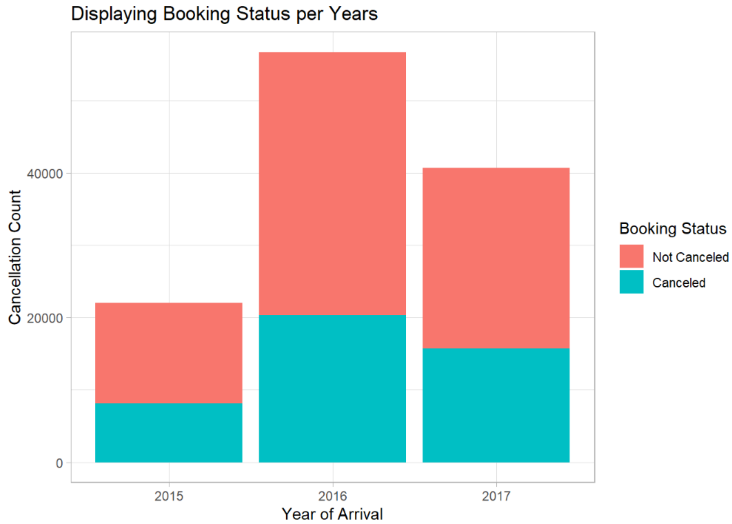
```
## # A tibble: 14 x 3
## # Groups:   hotel [2]
##    hotel        market_segment   No.
##    <fct>        <fct>          <int>
##  1 City Hotel   Online TA      38748
##  2 Resort Hotel Online TA      17729
##  3 City Hotel   Offline TA/TO  16747
##  4 City Hotel   Groups         13975
##  5 Resort Hotel Offline TA/TO   7472
##  6 Resort Hotel Direct          6513
##  7 City Hotel   Direct          6093
##  8 Resort Hotel Groups          5836
##  9 City Hotel   Corporate       2986
## 10 Resort Hotel Corporate       2309
## 11 City Hotel   Complementary    542
## 12 City Hotel   Aviation         237
## 13 Resort Hotel Complementary    201
## 14 City Hotel   Undefined          2
```

2) Understanding Booking Cancellation Behavior

After that, the various factors included in the dataset were further explored in respect to their relation with the target factor "is_canceled" giving an insight on how to select the variables having the strongest relation in order to set an accurate prediction model. The gained knowledge and understanding of the visuals in this section are documented in the insight section of this report.

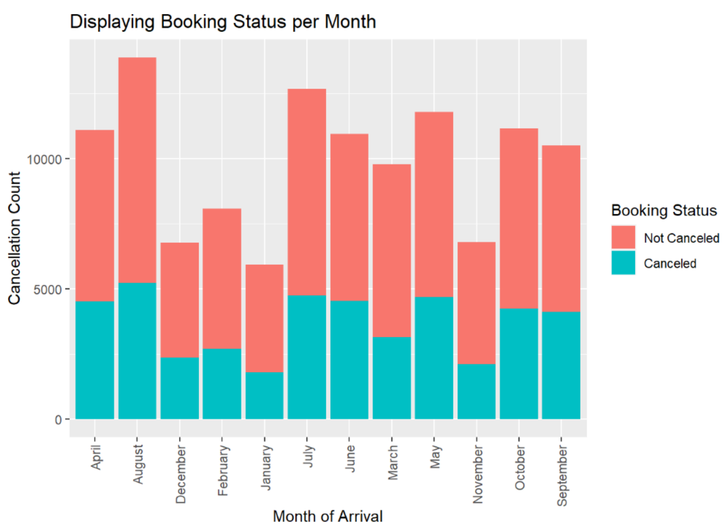
- Display booking status per year:

```
hotel_data %>% ggplot(aes(x=arrival_date_year, fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Years",
       x= "Year of Arrival",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme_light()
```



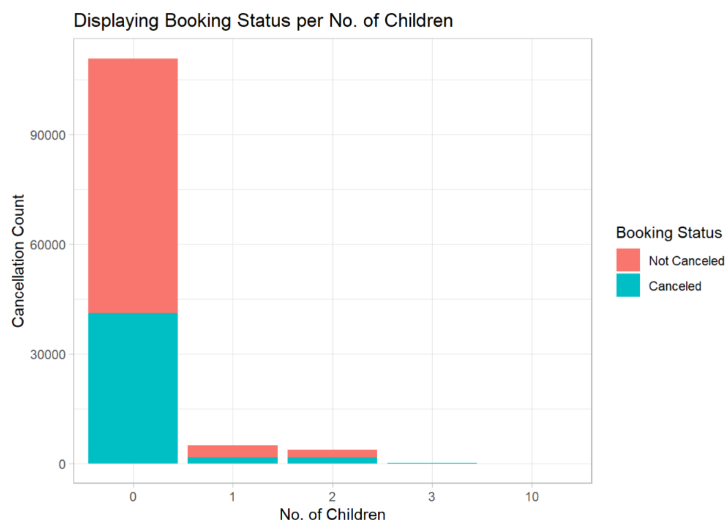
❖ Display booking status per month:

```
hotel_data %>% ggplot(aes(x=arrival_date_month, fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Month",
       x= "Month of Arrival",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



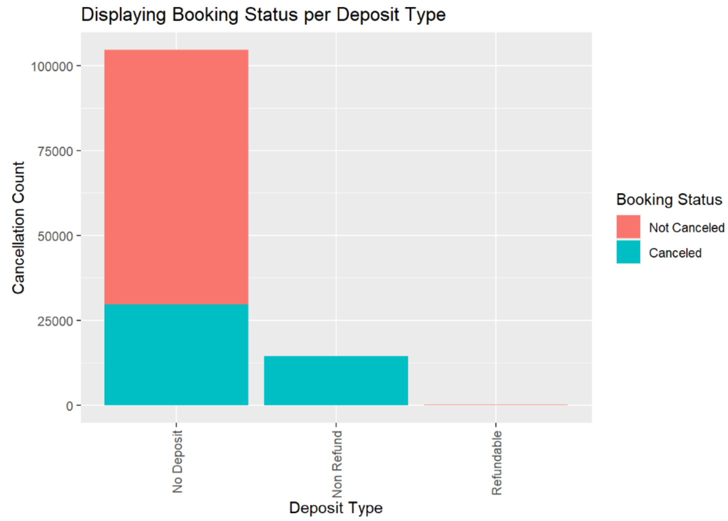
❖ Display booking status per No. of children:

```
hotel_data %>% ggplot(aes(x=as.factor(children), fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per No. of Children",
       x= "No. of Children",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme_light()
```



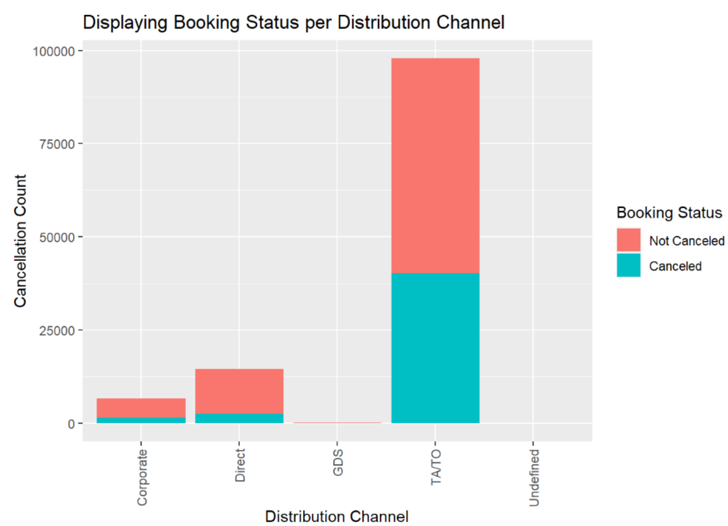
❖ Display booking status per deposit type:

```
hotel_data %>% ggplot(aes(x=deposit_type, fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Deposit Type",
       x= "Deposit Type",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



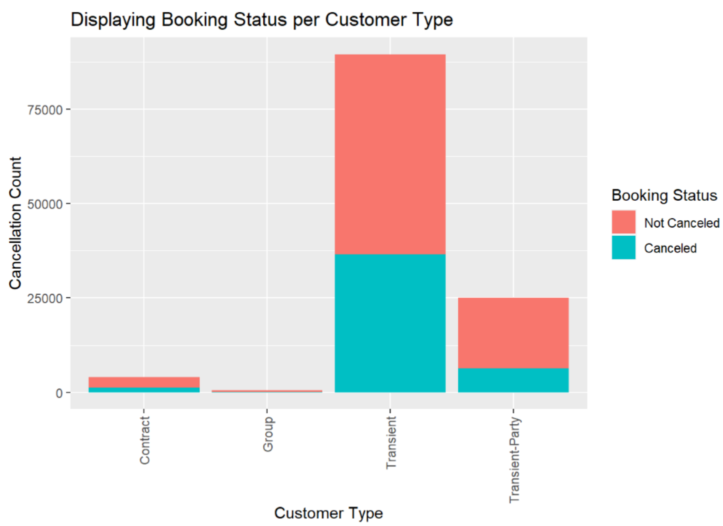
❖ Display booking status per distribution channel:

```
hotel_data %>% ggplot(aes(x=distribution_channel, fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Distribution Channel",
       x= "Distribution Channel",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



❖ Display booking status per customer type:

```
hotel_data %>% ggplot(aes(x=customer_type, fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Customer Type",
       x= "Customer Type",
       y= "Cancellation Count")+
  scale_fill_discrete(name= "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

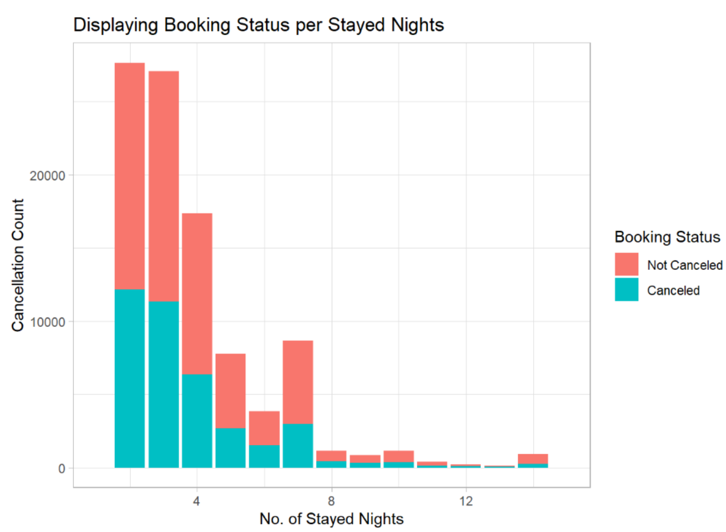


❖ Display booking status per repeated guests (0 non-repeated-guest, 1 repeated guest):

```
hotel_data %>% ggplot(aes(x=as.factor(is_repeated_guest), fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Repeated Guests",
       x= "Repeated Guests",
       y= "Cancellation Count")+
  scale_fill_discrete(name= "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme_light()
```

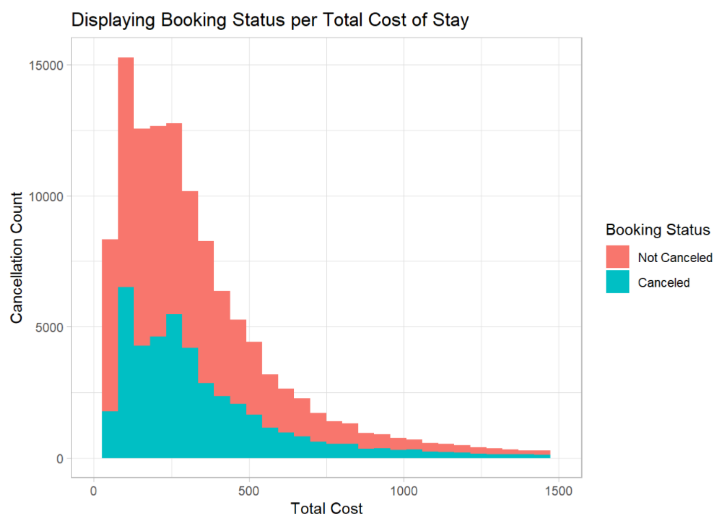
❖ Display booking status per stayed nights:

```
hotel_data %>% ggplot(aes(x=total_nights, fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Stayed Nights",
       x= "No. of Stayed Nights",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  xlim(1,15)+
  theme_light()
```



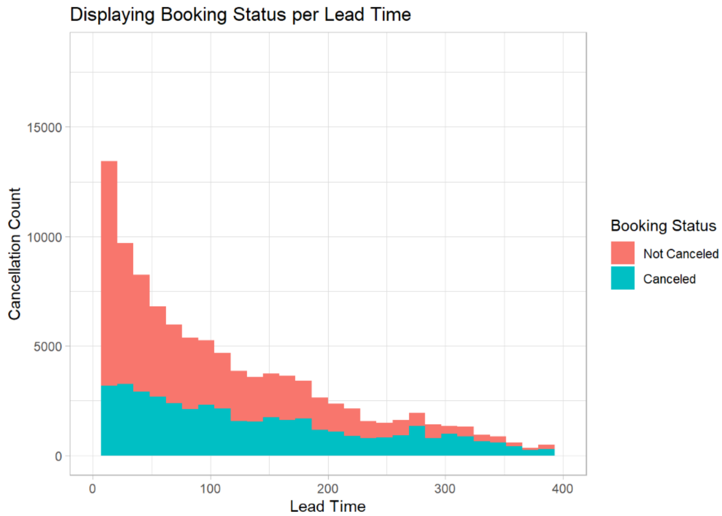
❖ Display booking status per total cost of stay:

```
hotel_data %>% ggplot(aes(x=total_cost, fill = factor(is_canceled)))+
  geom_histogram()+
  labs(title="Displaying Booking Status per Total Cost of Stay",
       x= "Total Cost",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  xlim(0,1500)+
  theme_light()
```



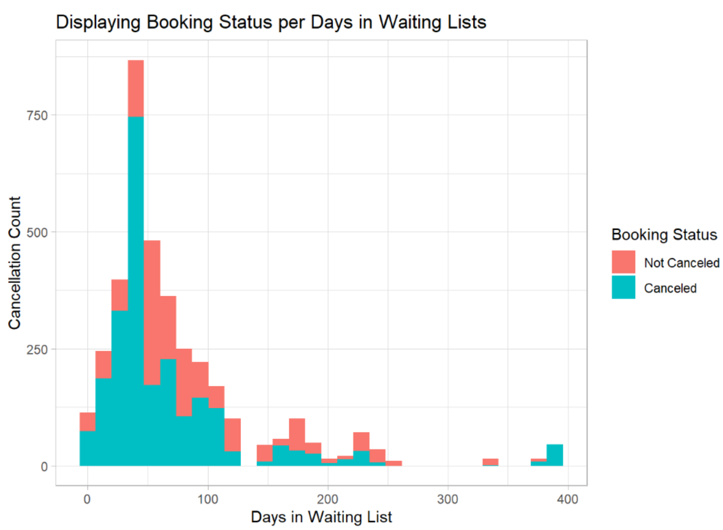
❖ Display booking status per lead time:

```
hotel_data %>% ggplot(aes(x=lead_time, fill = factor(is_canceled)))+
  geom_histogram()+
  labs(title="Displaying Booking Status per Lead Time",
       x= "Lead Time",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  xlim(0,400)+
  theme_light()
```



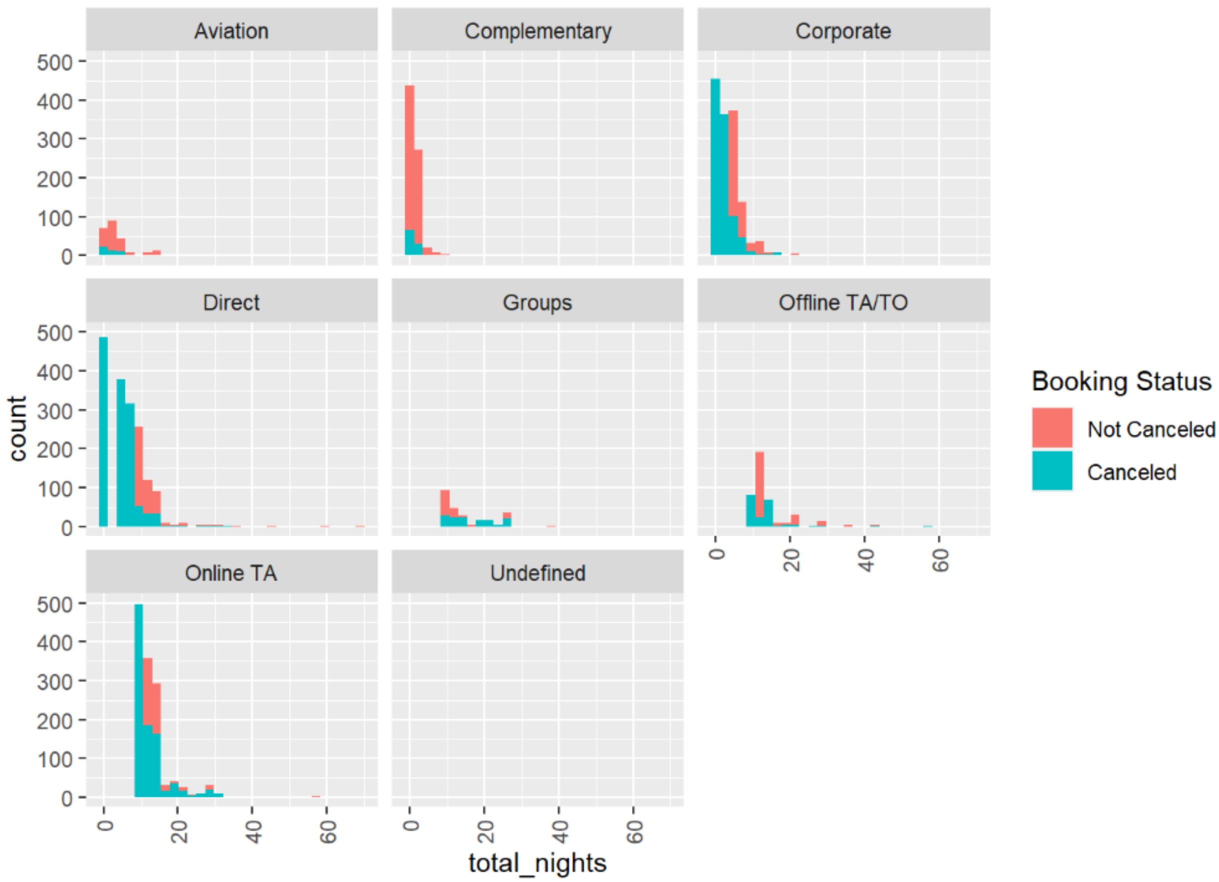
❖ Display booking status per days in waiting list:

```
hotel_data %>% filter(days_in_waiting_list>1) %>%
  ggplot(aes(x=days_in_waiting_list, fill = factor(is_canceled)))+
  geom_histogram()+
  labs(title="Displaying Booking Status per Days in Waiting Lists",
       x= "Days in Waiting List",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme_light()
```



❖ Display booking Status across Market Segments:

```
hotel_data %>% ggplot(aes(x=total_nights, fill=factor(is_canceled)))+
  geom_histogram()+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),
                      label = c("Not Canceled", "Canceled"))+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ylim(0,500)+
  facet_wrap(~market_segment)
```



III. Insights Gained

Based on the performed exploration strategies and the concluded data properties, some observations were interesting:

- It is noted that City Hotel had much more reservations than Resort Hotels.
- Online City Hotel bookings through agent had the highest record of cancellations.
- The greatest number of cancellations occurred in the 2016 (however collected data covered three years only; 2015, 2016, 2017).
- It was noted that number of children in the booking did not have a major impact on cancellation.
- Majority of cancellation transactions occurred for those with no deposit provided at the time of booking (since there will be no obligations).
- Bookings through Travel Agents and Tour Operators had the greatest number of cancellations compared to other distribution channels (this may be due to agreements set between those agents & operators and the service providers).
- Transient Customers had the highest number of cancellations.
- Majority of cancellations occurred for non-repeating guests.
- It was noted that as the number of nights of stay and the total cost were increasing the number of cancellations was decreasing.
- It was apparent that the shorter the time lead the higher number of cancellations was.
- It was noted that when total nights of stays were around 20, major of cancellations occurred in the Direct, Online and Corporate market segments.

Due to limited resources to perform adequate analysis taking into consideration all the factors available in this data set, it has been decided to study the correlation coefficient values between the target variable (is_canceled) and the rest of the variables (total 34 counting total_cost & total_nights) and pick those with the strongest relation for generating the models.

In order to start the modeling, process the factor variables has been converted to numeric variables in our training set.

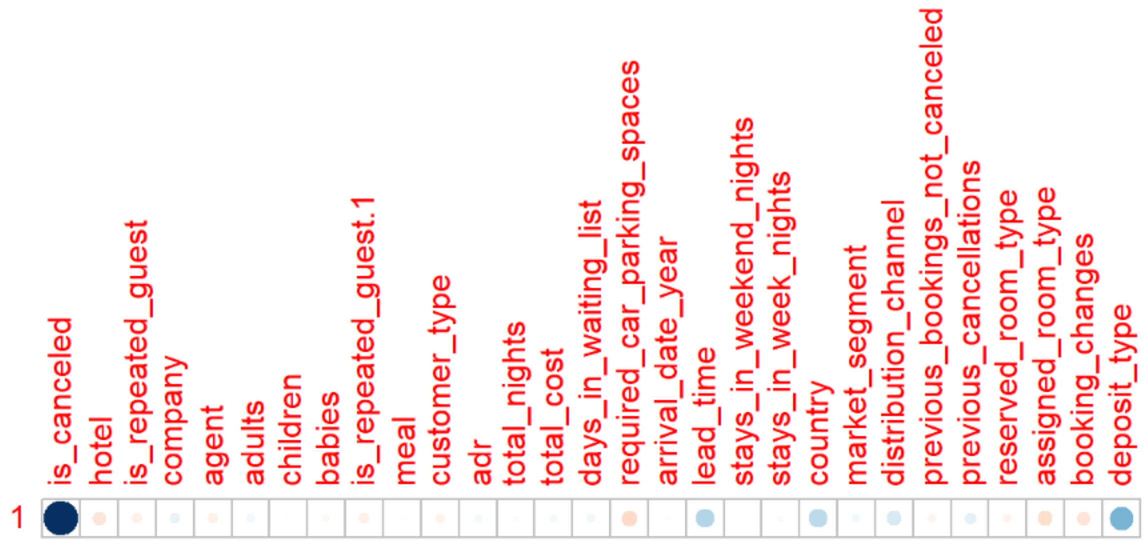
```
conv_numeric <- hotel_train %>% mutate_if(is.factor, as.numeric)
```

Calculate the correlation coefficient for the target variable "is_canceled":

```
correlations <- cor(conv_numeric$is_canceled,  conv_numeric[,c("is_canceled","hotel","is_repeated_guest", "compan
y",          "agent", "adults", "children", "babies", "is_repeated_guest",
"meal","customer_type", "adr", "total_nights", "total_cost",  "days_in_waiting_list", "required_car_parking_space
s", "arrival_date_year","lead_time", "stays_in_weekend_nights", "stays_in_week_nights", "country", "market_segmen
t","distribution_channel", "previous_bookings_not_canceled", "previous_cancellations",
"reserved_room_type", "assigned_room_type", "booking_changes", "deposit_type")])
```

Then plot the correlation coefficient:

```
corrplot(correlations, method="circle")
```



It is apparent from the plot that the following variables have strong relation to cancellation:

deposit_type	distribution_channel	previous_cancellations
country	company	lead_time
	required_car_parking	

Then the factors with the strong relation to the target variable were selected for further modeling and analysis from both training & testing datasets hotel_train, hotel_valid respectively.

```
hotel_train <- hotel_train[c("is_canceled", "country", "deposit_type", "distribution_channel", "company", "lead_t
ime", "required_car_parking_spaces", "previous_cancellations")]
colnames(hotel_train)
```

```
## [1] "is_canceled"                 "country"
## [3] "deposit_type"                "distribution_channel"
## [5] "company"                     "lead_time"
## [7] "required_car_parking_spaces" "previous_cancellations"
```

```
hotel_valid <- hotel_train[c("is_canceled", "country", "deposit_type", "distribution_channel", "company", "lead_t
ime", "required_car_parking_spaces", "previous_cancellations")]
colnames(hotel_valid)
```

```
## [1] "is_canceled"                 "country"
## [3] "deposit_type"                "distribution_channel"
## [5] "company"                     "lead_time"
## [7] "required_car_parking_spaces" "previous_cancellations"
```

IV. Create Data Partitions for training and validation purposes

Hotel_data was subdivided to hotel_train; representing 90% of the total data, and hotel_valid; representing 10% of the total data and serving as the validation dataset assuming that it is unknown dataset.

```
set.seed(1, sample.kind="Rounding")

test_index <- createDataPartition(y = hotel_data$is_canceled, times = 1, p = 0.1, list = FALSE)
hotel_train <- hotel_data[-test_index,]
dim(hotel_train)
```

```
## [1] 107451     34
```

```
temp <- hotel_data[test_index,]

# Validation data set is 10% of the hotel_data
hotel_valid <- temp
dim(hotel_valid)
```

```
## [1] 11939    34
```

```
# Clean memory
rm(temp, test_index)
```

V. Modelling Approach

First, the factor variables were converted to numeric values for modeling purposes:

```
hotel_train <- hotel_train %>% mutate_if(is.factor, as.numeric)
hotel_valid <- hotel_valid %>% mutate_if(is.factor, as.numeric)
```

Data modelling was based on three various modelling approaches:

1) Logistic Regression Model

For the binary outcome model the family 'binomial' was called in the glm function. Then, the predicted model was validated on the validation dataset hotel_valid through the predict function. The outcomes of the model were classified into values of 0 and 1 based on the prediction result greater or less than 0.5.

```
set.seed(1, sample.kind="Rounding")

# Generate glm model
glm_model <- glm(is_canceled~.,family="binomial", data = hotel_train)

# Predict the model on the validation dataset
pred_glm <- predict(glm_model, hotel_valid, type="response")
# Record the model prediction results in a binary form of 0 and 1
pred_glm_class <-ifelse(pred_glm>0.5,"1","0")

# Record the prediction against actual data in the validation dataset
glm_pred_table <- table(pred_glm_class, hotel_valid$is_canceled, dnn=c("predicted","actual"))
glm_pred_table
```

```
##          actual
## predicted     0     1
##         0 65522 23095
##         1  2029 16805
```

The accuracy of the model was derived form the prediction table where predicted value matched actual in the validation dataset, and then the sum was divided by the total validation dataset and multiplied by 100 to get the percentage value.

```
# Calculate model accuracy based on the prediction table "pred_table" where prediction met actual in the validati
on dataset hotel_valid
glm_accuracy <- ((glm_pred_table[1,1]+glm_pred_table[2,2])/nrow(hotel_valid))*100
```

Finally, the results and model description were stored in a dataframe for future comparison with other models.

```
model_results <- data.frame(Method_Name = "Logestic Regression Model", Accuracy = glm_accuracy)
model_results
```

```
##                 Method_Name Accuracy
## 1 Logestic Regression Model 76.61818
```

```
# Store and Update Model Results Table
model_results %>% knitr::kable()
```

Method_Name	Accuracy
Logestic Regression Model	76.61818

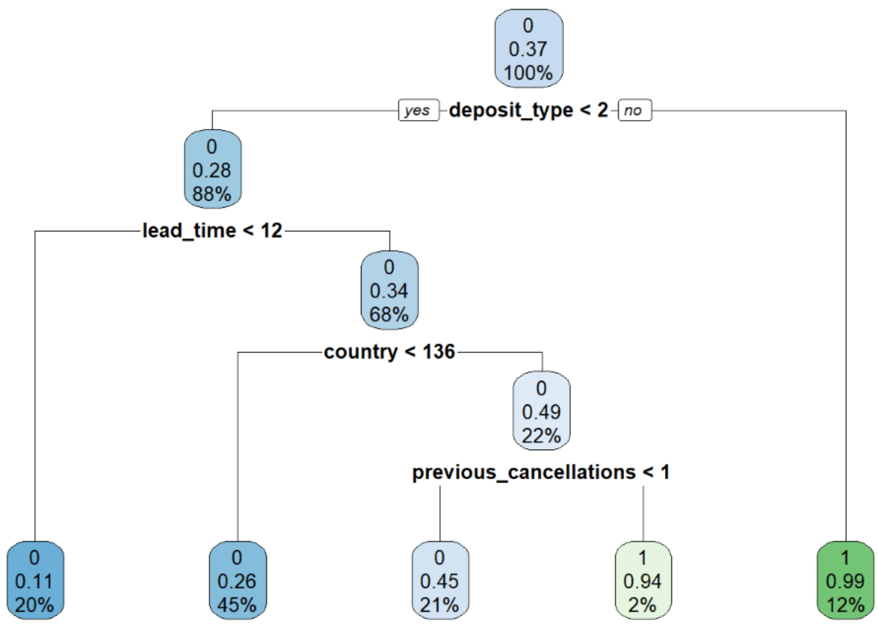
2) Classification Tree Model

Classification/decision tree model was used and plotted to generate the second model in this project. It is noted that deposit_type was the root node in this tree forming the first split, followed by the lead_time (time from reservation to actual appearance), and then the country, and finally the previous cancellations.

```
set.seed(1, sample.kind="Rounding")

# Generate the classification tree model
class_tree_model <- rpart(is_canceled~., data = hotel_train, method="class")

# Plot the classification tree
rpart.plot(class_tree_model)
```




```
# Predict the model on the validation dataset
pred_class_tree <- predict(class_tree_model, as.data.frame(hotel_valid), type = "class")

# Display prediction results
class_tree_pred_table <- table(pred_class_tree, hotel_valid$is_canceled, dnn = c("Predicted","Actual"))
class_tree_pred_table
```

```
##          Actual
## Predicted     0     1
##         0 67244 24958
##         1   307 14942
```

The accuracy of the model was derived form the prediction table where predicted value matched actual in the validation dataset, and then the sum was divided by the total validation dataset and multiplied by 100 to get the percentage value.

```r
# Calculate accuracy of the class tree model
class_tree_accuracy <- ((class_tree_pred_table[1,1]+class_tree_pred_table[2,2])/nrow(hotel_valid))*100

model_results <- bind_rows(model_results, data.frame(Method_Name = "Classification Tree Model", Accuracy = class_
tree_accuracy))
model_results
```

```
##                 Method_Name Accuracy
## 1 Logestic Regression Model 76.61818
## 2 Classification Tree Model 76.48696
```

```r
# Store and Update Model Results Table
model_results %>% knitr::kable()
```

Model results were then stored for future comparison with other models.

Method_Name	Accuracy
Logestic Regression Model	76.61818
Classification Tree Model	76.48696

3) Random Forest Model

The third and final model generated for this project was the random forest. By default the number of trees generated for this model is 10. In this project various number of trees/nodes were selected to see the best performing option. It has been noted that when the number of trees was 50 the model performed better.

A similar approach to previous models was used in predicting, classifying the outcome, calculating model accuracy and finally storing the results of the model.

```r
set.seed(1, sample.kind="Rounding")

# Generate random forest model
rf_model <- randomForest(is_canceled~., data = hotel_train, ntree= 50)

# Predict the model on the validation dataset
pred_rf <- predict(rf_model,hotel_valid,type="response")

# Record the model prediction results in a binary form of 0 and 1
pred_rf_class <-ifelse(pred_rf>0.5,"1","0")

# Record the prediction against actual data in the validation dataset
rf_pred_table <- table(pred_rf_class, hotel_valid$is_canceled, dnn = c("predicted","actual"))
rf_pred_table
```

```
##          actual
## predicted     0     1
##         0 64634 19834
##         1  2917 20066
```

The accuracy of the model was derived from the prediction table where predicted value matched actual in the validation dataset, and then the sum was divided by the total validation dataset and multiplied by 100 to get the percentage value.

```
# Calculate accuracy of the Random Forest Model
rf_accuracy <- ((rf_pred_table[1,1]+rf_pred_table[2,2])/nrow(hotel_valid))*100
```

```
model_results <- bind_rows(model_results, data.frame(Method_Name = "Random Forest Model", Accuracy = rf_accurac
y))
model_results
```

```
##                  Method_Name Accuracy
## 1 Logestic Regression Model 76.61818
## 2 Classification Tree Model 76.48696
## 3       Random Forest Model 78.81546
```

```
# Store and Update Model Results Table
model_results %>% knitr::kable()
```

Method_Name	Accuracy
Logestic Regression Model	76.61818
Classification Tree Model	76.48696
Random Forest Model	78.82663

Discussion of the three models and their performance is further detailed in the Result section of this report.

Results

After conducting comprehensive exploration and analysis of the data, different models were generated taking into consideration 7 different factors with strong positive and negative relations to the target variable is_cancel.

The evaluation criteria of all three generated data models considered the accuracy of the model based on the predicted cancellations matching the actual cancellation in the validation dataset hotel_valid. As the outcome of the models is binary (0 and 1) the accuracy was simply calculated from the prediction table for each of the generated models as follows:

(Predicted[0]Actual[0] + Predicted[1]Actual[1] / total obs. of validation dataset) * 100

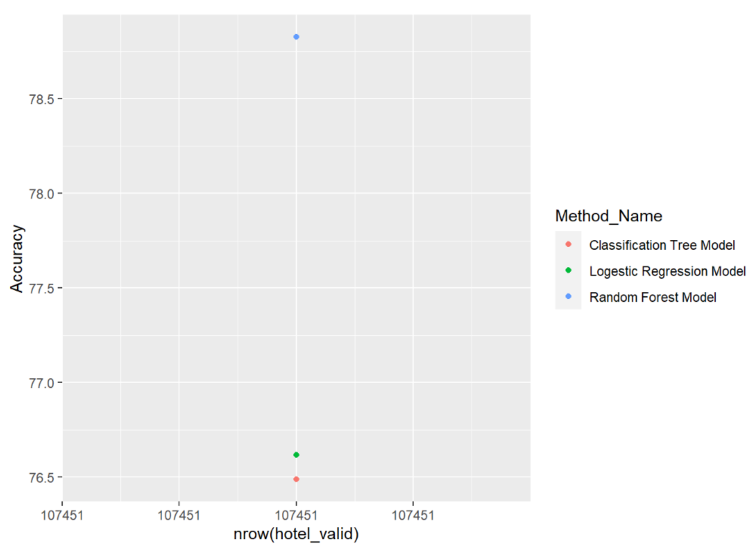
	Actual	
Predicted	0	1
0		
1		

The accuracies of the three generated models are summarized in the table below:

Method_Name	Accuracy
Logestic Regression Model	76.61818
Classification Tree Model	76.48696
Random Forest Model	78.82663

And plotted in the below visual:

```
model_results %>% ggplot(aes(nrow(hotel_valid),Accuracy, color=Method_Name))+geom_point()
```



As shown, the best model was the Random Forest Model with an accuracy score of approximately 79%. The selected number of trees for this model was 50.

Conclusion

I. Summary of the report

In conclusion, based on the available resources the best machine algorithm for predicting future booking cancellations for this project took into consideration seven different affecting the cancellation of hotel bookings. Those factors were selected based on the correlation coefficient value associated with the target logical variable in the dataset 'is_canceled'. It has been concluded that the Random Forest Model would give the most accurate prediction for future booking cancellations.

II. Limitation and Future Work

This algorithm may be further enhanced to achieve better results. More complex algorithms can be generated and evaluated through better processing power and analyzing more factors in dataset may also lead to better results. Due to limited available computational processing power and the nature of the dataset (short period of data records) only three models were tested and validated on the available dataset.