Hotel-booking-demand.Rmd

Rajeev

27/08/2020

Introduction

Being able to accurately predict future hotel booking cancellation has a great impact on the 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, a machine learning algorithm was developed based on testing three different data models: logistic regression, classification tree, and random forest to predict future booking cancellation based on the characteristics of the collected bookings data.

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.

Methodology

After exploring the dataset, three different models were adopted based on 7 different features of the dataset selected based on the correlation coefficient with the target variable is_canceled.

1. Logistic Regression Model
2. Classification Tree Model
3. Random Forest Model

Then, cross validation was applied to determine the best model with the highest accuracy value on the validation dataset.

```r
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")
```

```r
if(!require(scales)) install.packages("scales")
if(!require(readr)) install.packages("readr")
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")
```

Download and install necessary packages

To access the source file hotel_bookings.csv from the github repository

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

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

```
## '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" ...
```

```r
# 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)
```

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

```
##          hotel is_canceled lead_time arrival_date_year arrival_date_month
## 1 Resort Hotel           0       342              2015               July
## 2 Resort Hotel           0       737              2015               July
## 3 Resort Hotel           0         7              2015               July
## 4 Resort Hotel           0        13              2015               July
## 5 Resort Hotel           0        14              2015               July
## 6 Resort Hotel           0        14              2015               July
##   arrival_date_week_number arrival_date_day_of_month stays_in_weekend_nights
## 1                       27                         1                       0
## 2                       27                         1                       0
## 3                       27                         1                       0
## 4                       27                         1                       0
## 5                       27                         1                       0
## 6                       27                         1                       0
##   stays_in_week_nights adults children babies meal country market_segment
## 1                    0      2        0      0   BB     PRT         Direct
## 2                    0      2        0      0   BB     PRT         Direct
## 3                    1      1        0      0   BB     GBR         Direct
## 4                    1      1        0      0   BB     GBR      Corporate
## 5                    2      2        0      0   BB     GBR      Online TA
## 6                    2      2        0      0   BB     GBR      Online TA
##   distribution_channel is_repeated_guest previous_cancellations
## 1               Direct                 0                      0
## 2               Direct                 0                      0
## 3               Direct                 0                      0
## 4            Corporate                 0                      0
## 5                TA/TO                 0                      0
## 6                TA/TO                 0                      0
##   previous_bookings_not_canceled reserved_room_type assigned_room_type
## 1                              0                  C                  C
## 2                              0                  C                  C
## 3                              0                  A                  C
## 4                              0                  A                  A
## 5                              0                  A                  A
## 6                              0                  A                  A
```

```
##   booking_changes deposit_type agent company days_in_waiting_list customer_type
## 1               3   No Deposit  NULL    NULL                    0     Transient
## 2               4   No Deposit  NULL    NULL                    0     Transient
## 3               0   No Deposit  NULL    NULL                    0     Transient
## 4               0   No Deposit   304    NULL                    0     Transient
## 5               0   No Deposit   240    NULL                    0     Transient
## 6               0   No Deposit   240    NULL                    0     Transient
##   adr required_car_parking_spaces total_of_special_requests reservation_status
## 1   0                           0                         0          Check-Out
## 2   0                           0                         0          Check-Out
## 3  75                           0                         0          Check-Out
## 4  75                           0                         0          Check-Out
## 5  98                           0                         1          Check-Out
## 6  98                           0                         1          Check-Out
##   reservation_status_date total_nights total_cost
## 1              2015-07-01            0          0
## 2              2015-07-01            0          0
## 3              2015-07-02            1         75
## 4              2015-07-02            1         75
## 5              2015-07-03            2        196
## 6              2015-07-03            2        196
```

```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))
```

Clean the dataset to prepare for exploration and further analysis and replace any missing values

```
## [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 va

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]
}
```

1) Exploring the structure of the hotel_data dataset

```r
dim(hotel_data)
```

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

```r
str(hotel_data)
```

```
## 'data.frame':    119390 obs. of  34 variables:
##  $ hotel                         : Factor w/ 2 levels "City Hotel","Resort Hotel": 2 2 2 2 2 2 2 2 2
##  $ 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             : Factor w/ 3 levels "2015","2016",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ arrival_date_month            : Factor w/ 12 levels "April","August",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ 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                          : Factor w/ 5 levels "BB","FB","HB",..: 1 1 1 1 1 1 1 2 1 3 ...
##  $ country                       : Factor w/ 178 levels "ABW","AGO","AIA",..: 137 137 60 60 60 60 13
##  $ market_segment                : Factor w/ 8 levels "Aviation","Complementary",..: 4 4 4 3 7 7 4 4
##  $ distribution_channel          : Factor w/ 5 levels "Corporate","Direct",..: 2 2 2 1 4 4 2 2 4 4 .
##  $ 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            : Factor w/ 10 levels "A","B","C","D",..: 3 3 1 1 1 1 3 3 1 4 ...
##  $ assigned_room_type            : Factor w/ 12 levels "A","B","C","D",..: 3 3 3 1 1 1 3 3 1 4 ...
##  $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
##  $ deposit_type                  : Factor w/ 3 levels "No Deposit","Non Refund",..: 1 1 1 1 1 1 1 1
##  $ agent                         : Factor w/ 334 levels "1","10","103",..: 334 334 334 157 103 103 33
##  $ company                       : Factor w/ 353 levels "10","100","101",..: 353 353 353 353 353 353
##  $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ customer_type                 : Factor w/ 4 levels "Contract","Group",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```
##  $ 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             : Factor w/ 3 levels "Canceled","Check-Out",..: 2 2 2 2 2 2 2 2 1 1
##  $ reservation_status_date        : chr  "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...
##  $ total_nights                   : int  0 0 1 1 2 2 2 2 3 3 ...
##  $ total_cost                     : num  0 0 75 75 196 ...
```

```
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
##
```

```r
class(hotel_data)
```

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

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

Calculating and displaying the number of rows and columns in the hotel_data dataset

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

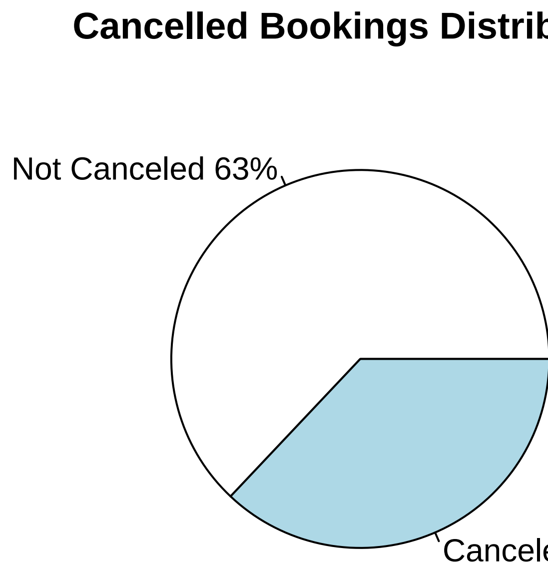
```r
table(hotel_data$hotel)
```

Displaying a table of the two available options of reservations

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

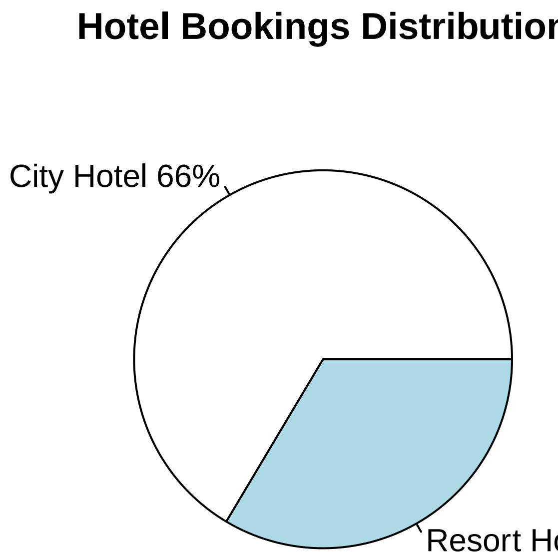
```r
# It is noted that City Hotel had much more reservations than Resort Hotels
```

```r
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")
```



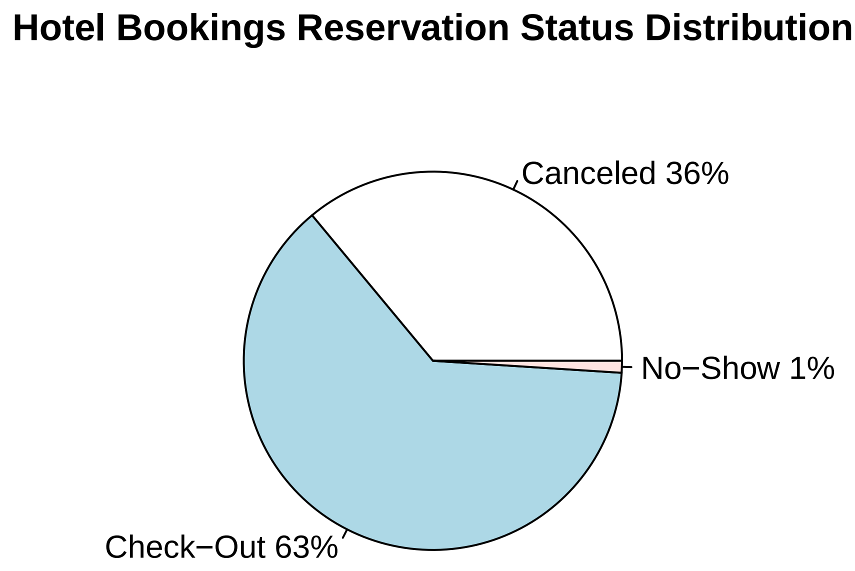
Display pie_chart of the canceled bookings

```
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")
```



Display pie_chart of the hotels variable
Display pie_chart of the Reservation Status of the Booking

```
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")
```



Display country with highest number of reservations for both city and resort

```
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
```

```
# Portugal has the highest number of hotel bookings
```

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

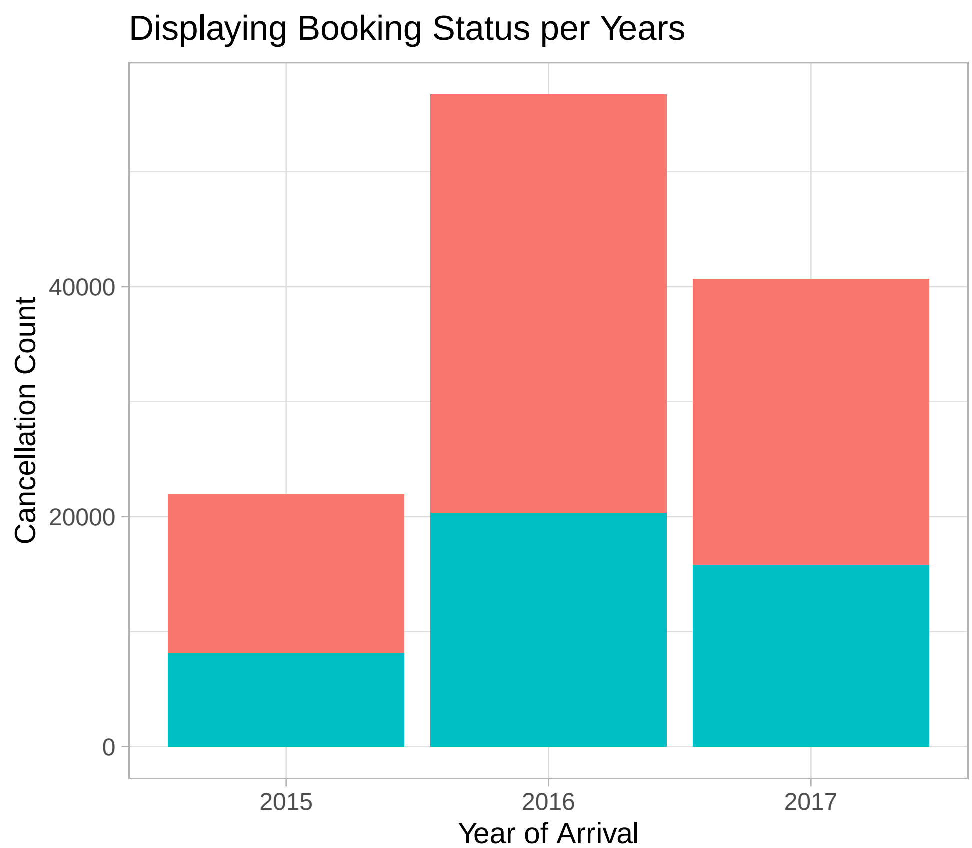
Display market segment with the highest number of bookings for both city and resort hotels

```
## # 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
```

```r
# Online City Hotel bookings through agent had the highest record
```

2) Understanding Cancellation Behavior in the hotel_data dataset

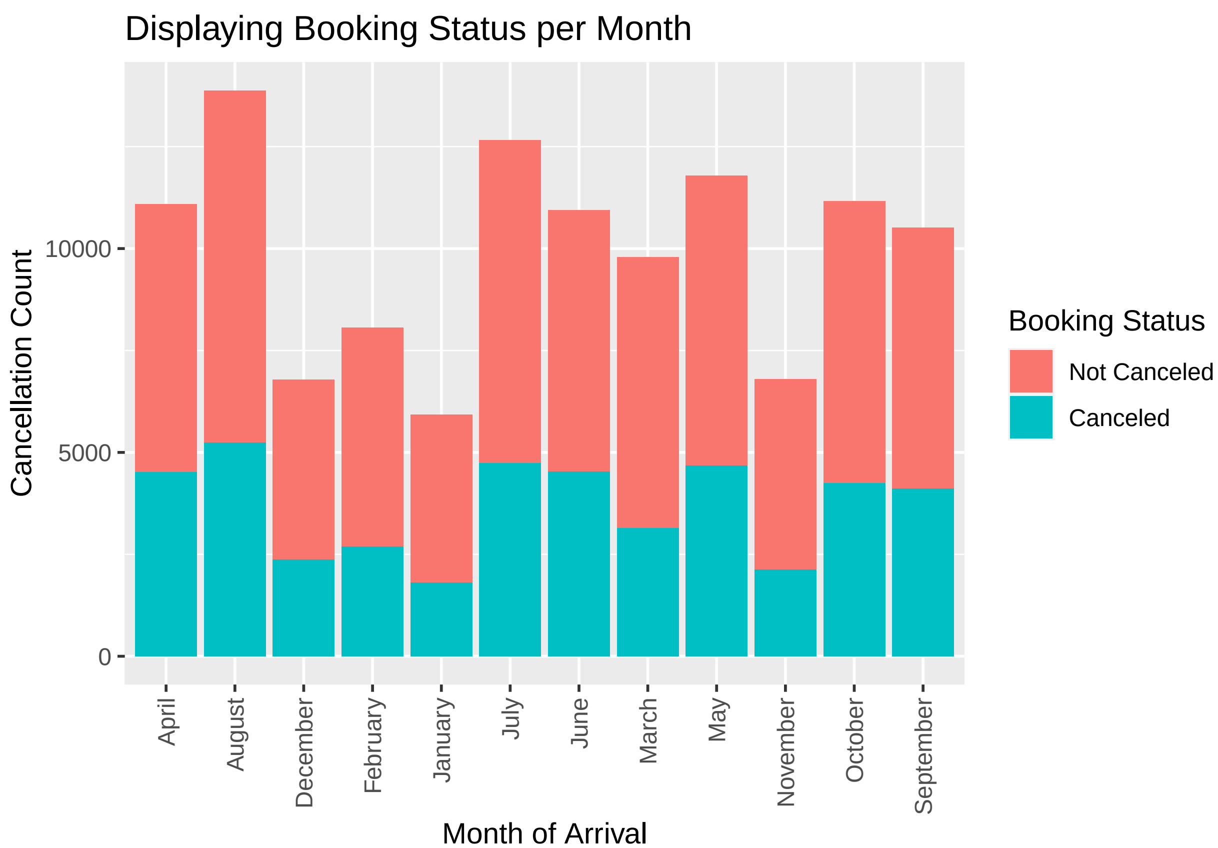
```r
hotel_data %>% ggplot(aes(x=arrival_date_year, fill = factor(is_canceled)))+
  geom_bar()+
  labs(title="Displaying Booking Status per Years",
       x= "Year of Arrival",
       y= "Cancellation Count")+
  scale_fill_discrete(name = "Booking Status",
                      breaks = c("0", "1"),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



Display booking status per year

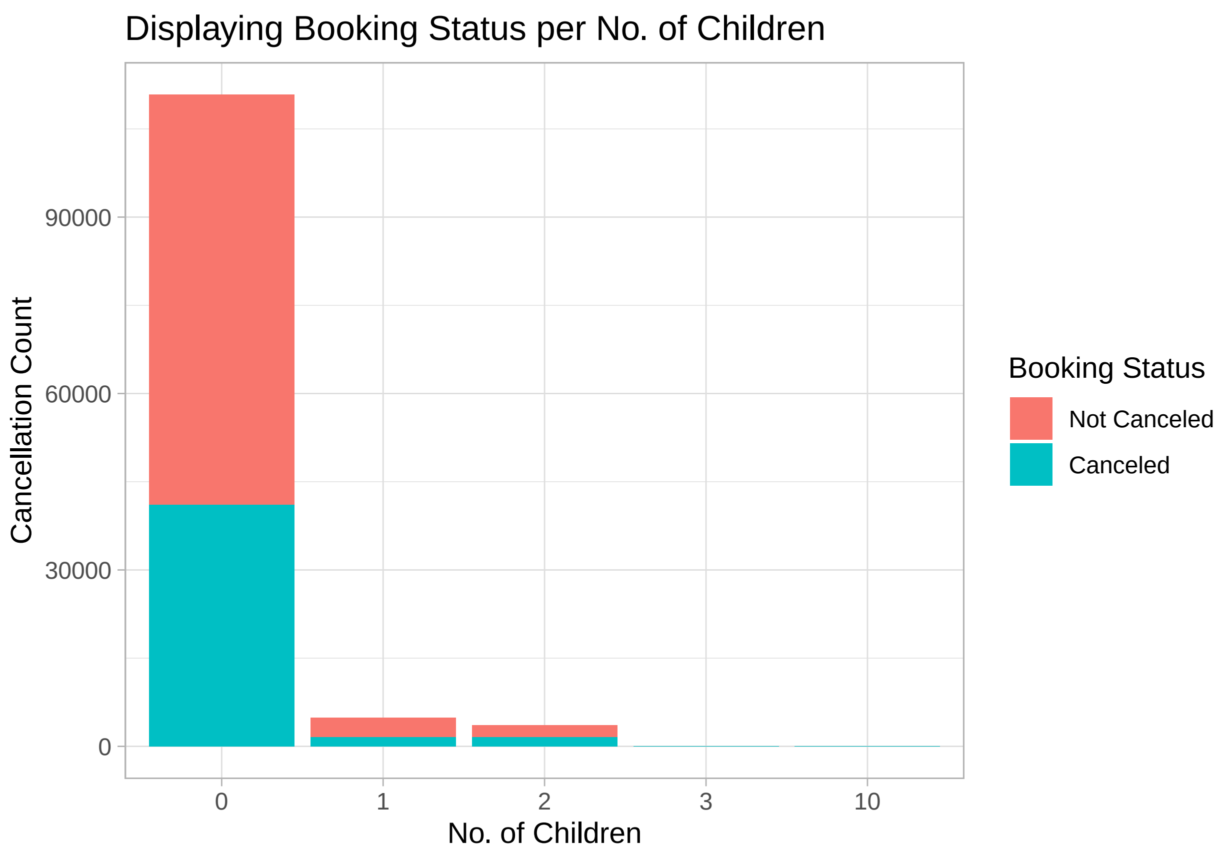
Display booking status per month

```
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))
```



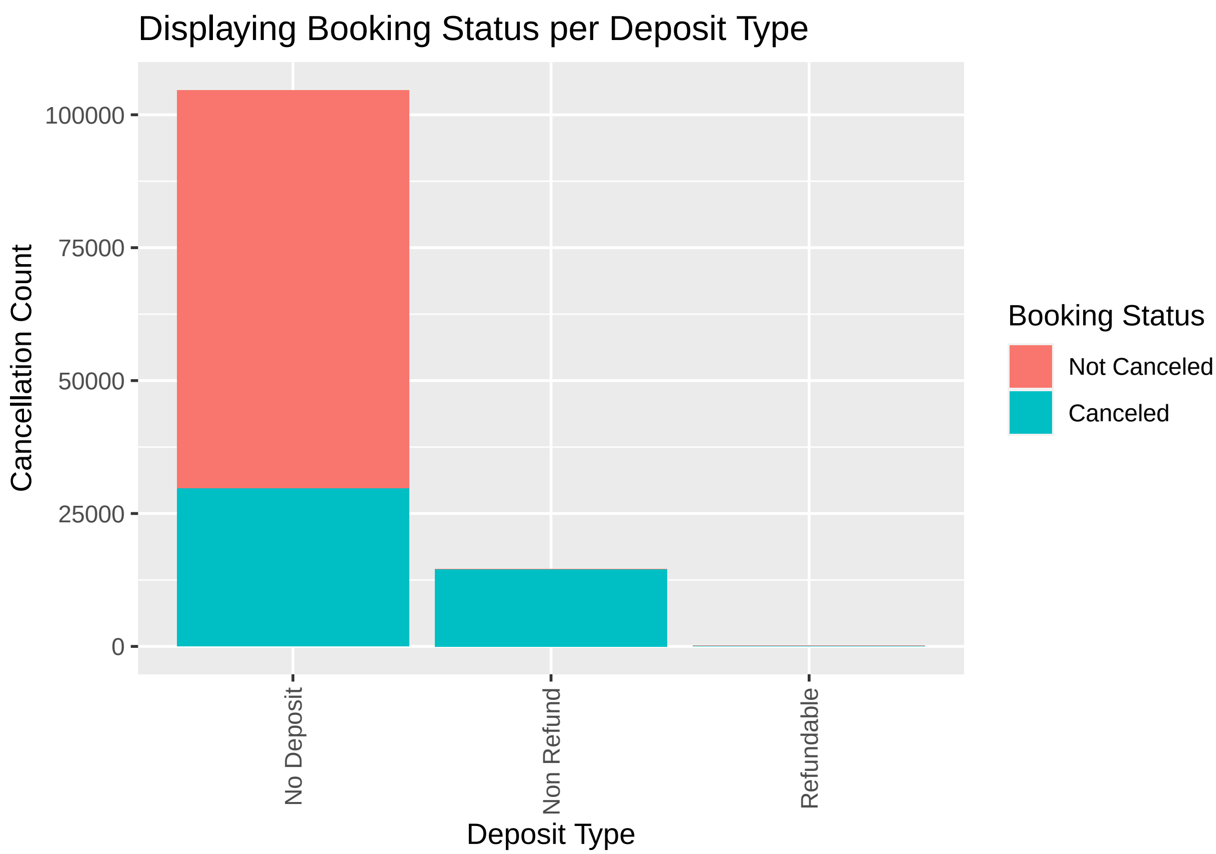
Display booking status per No. of children

```
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()
```



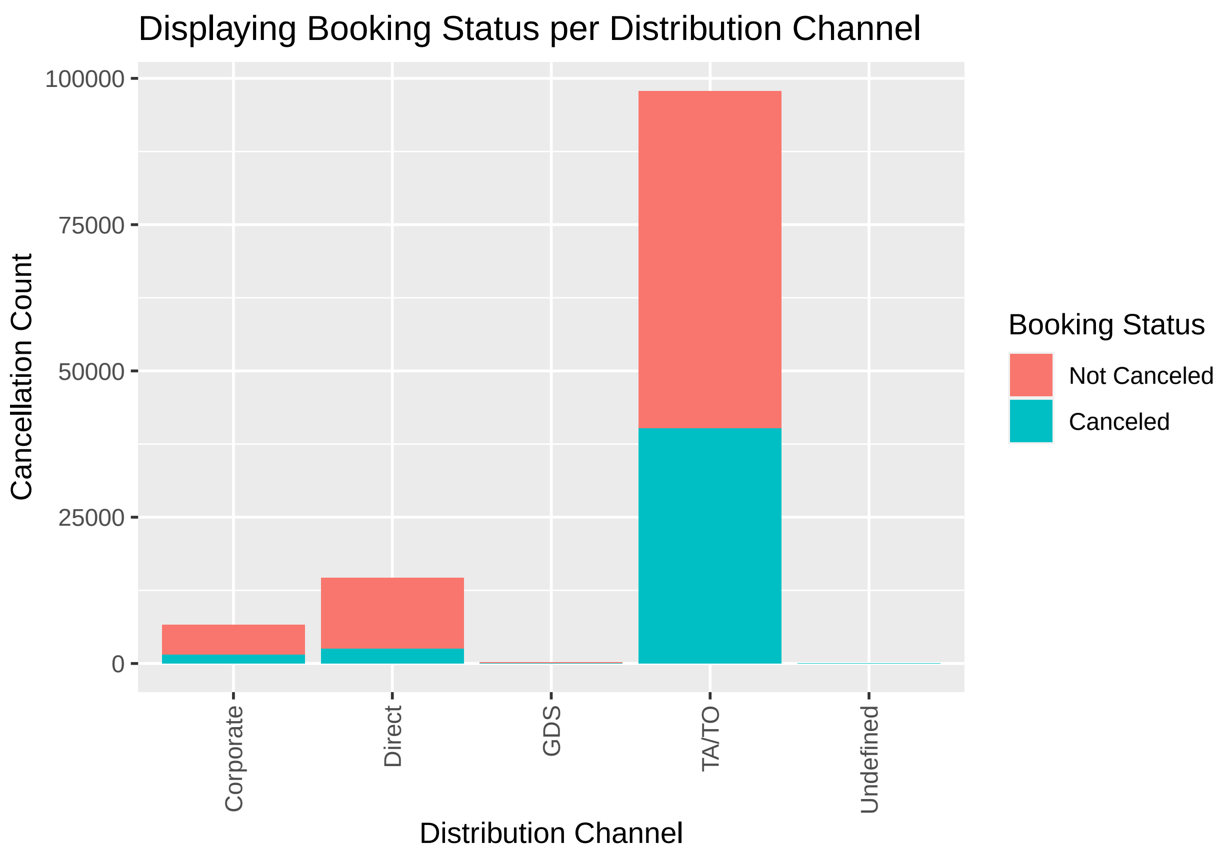
Display booking status per deposit type

```
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))
```



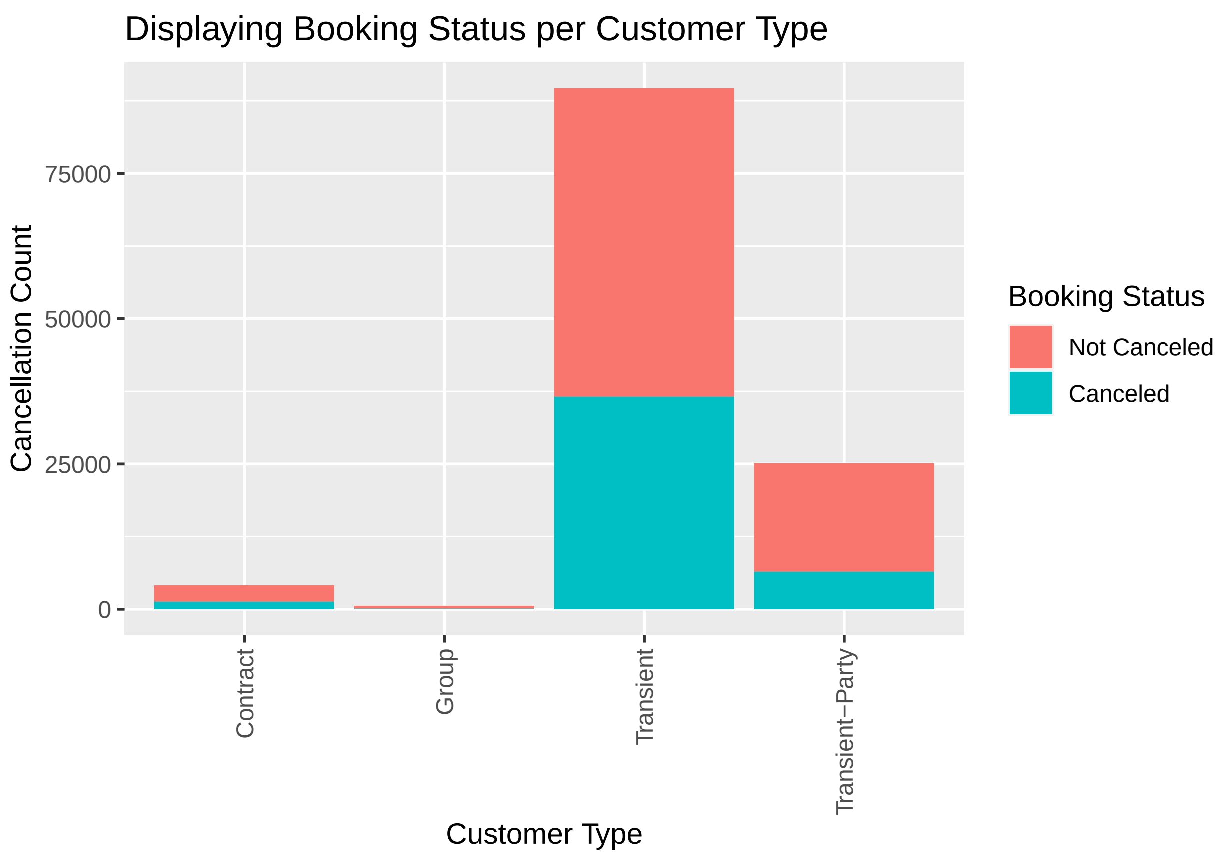
Display booking status per distribution channel

```
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))
```

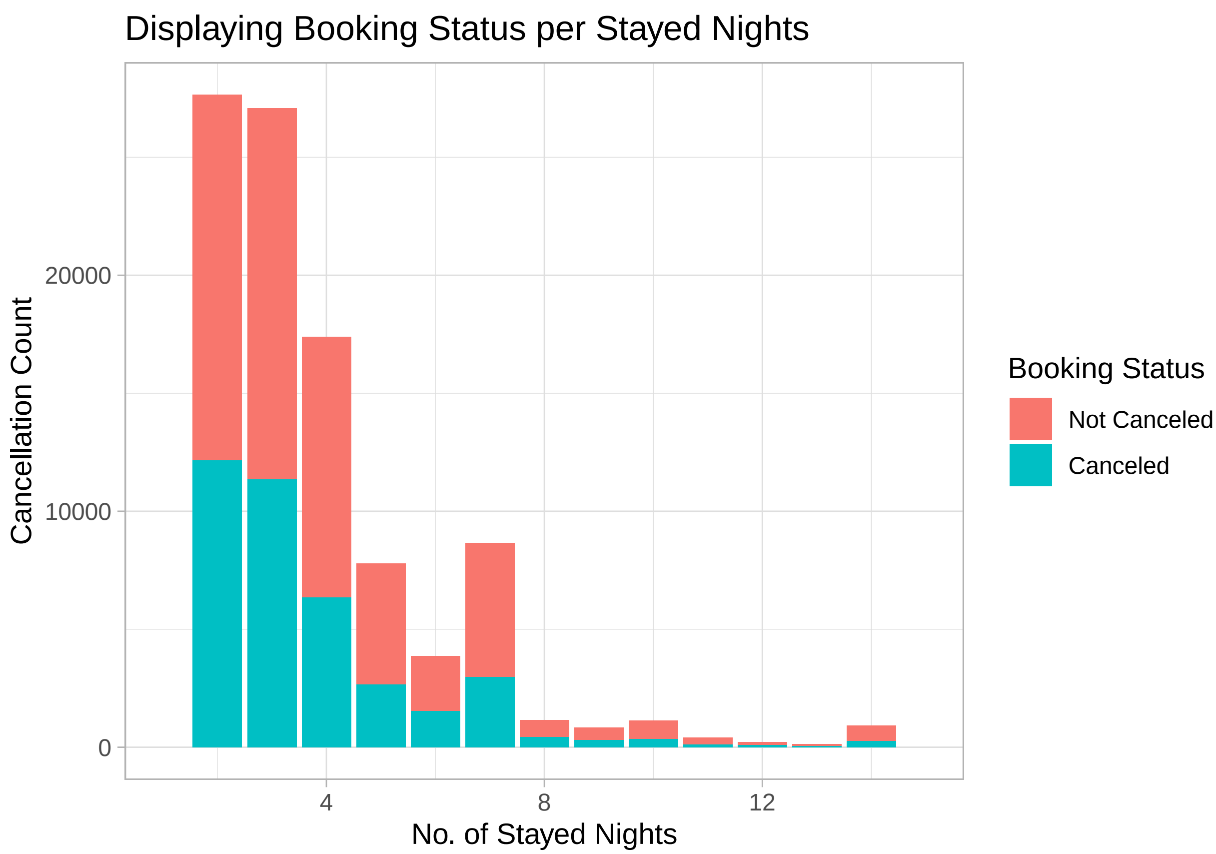


Display booking status per repeated guests

```
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()
```

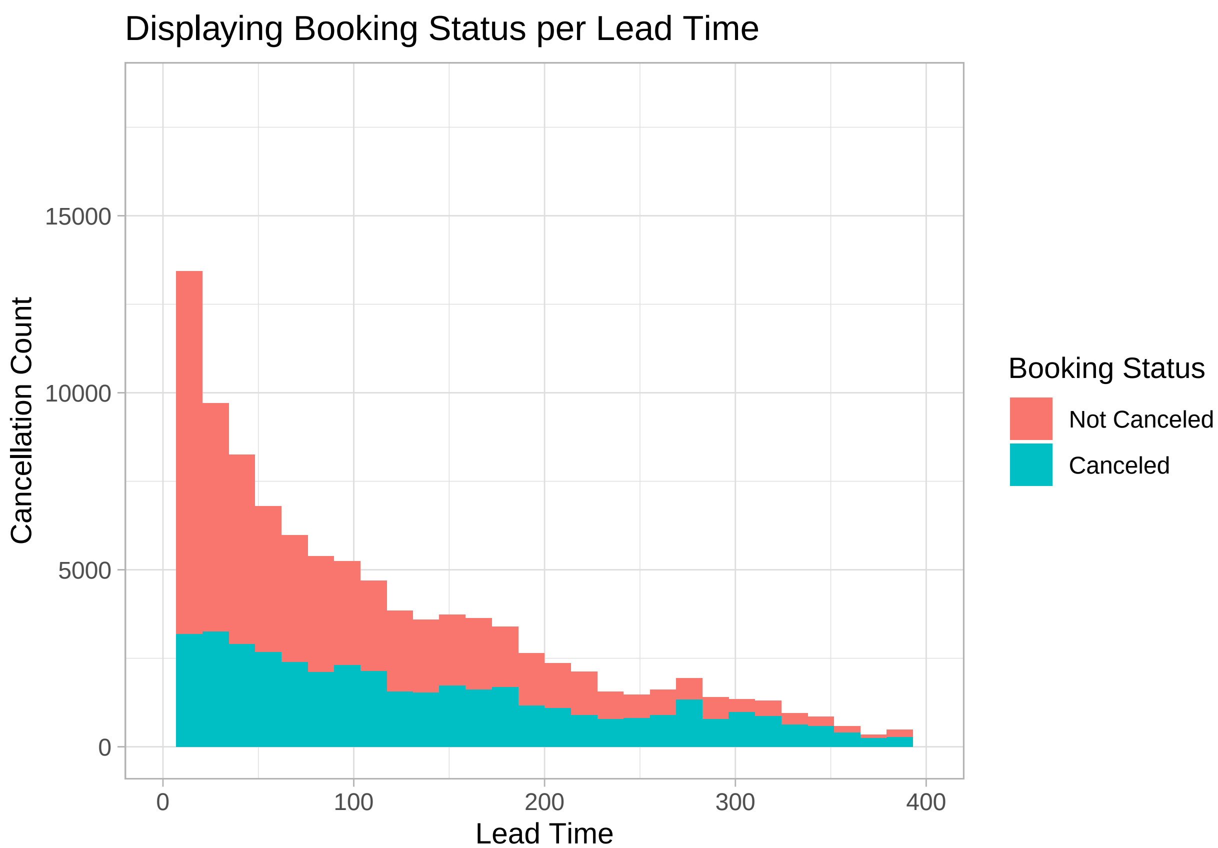


Display booking status per total cost of stay

```
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()
```

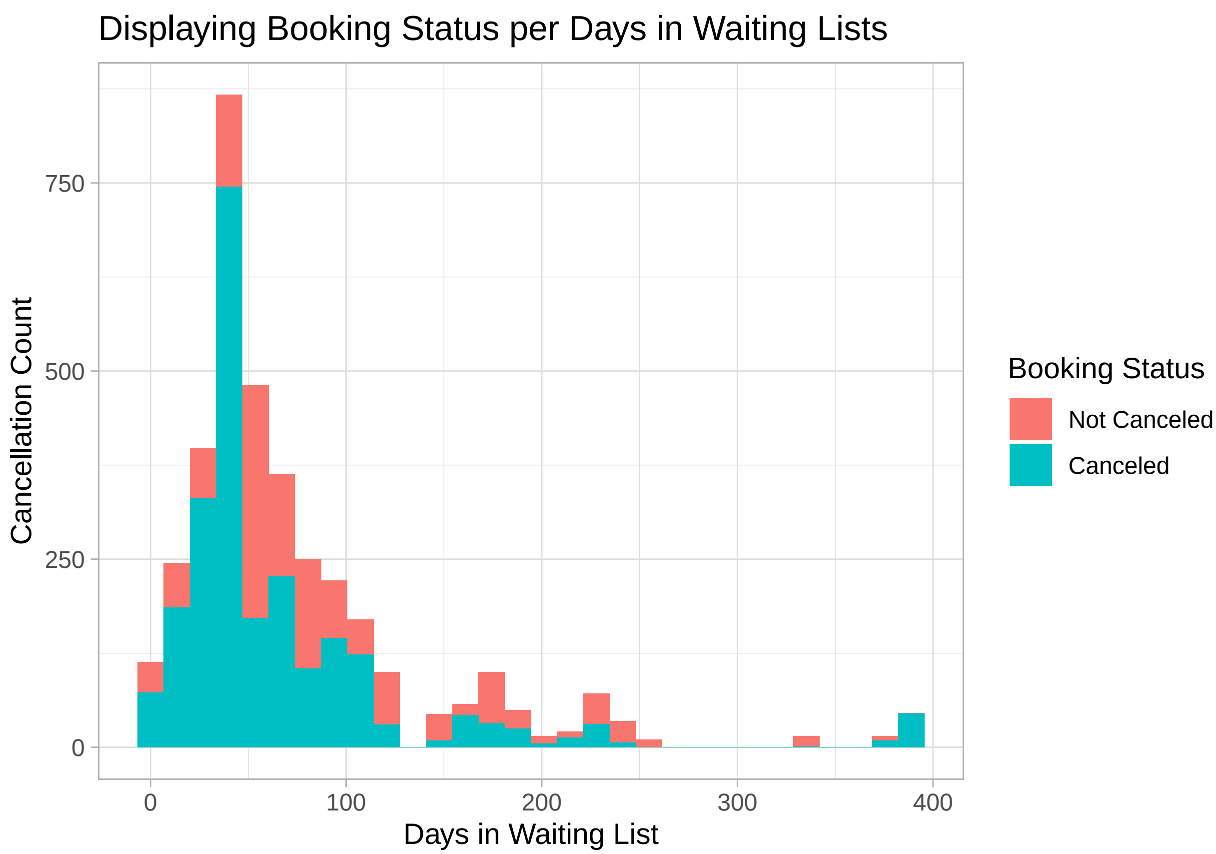
Display booking status per lead time

```
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()
```



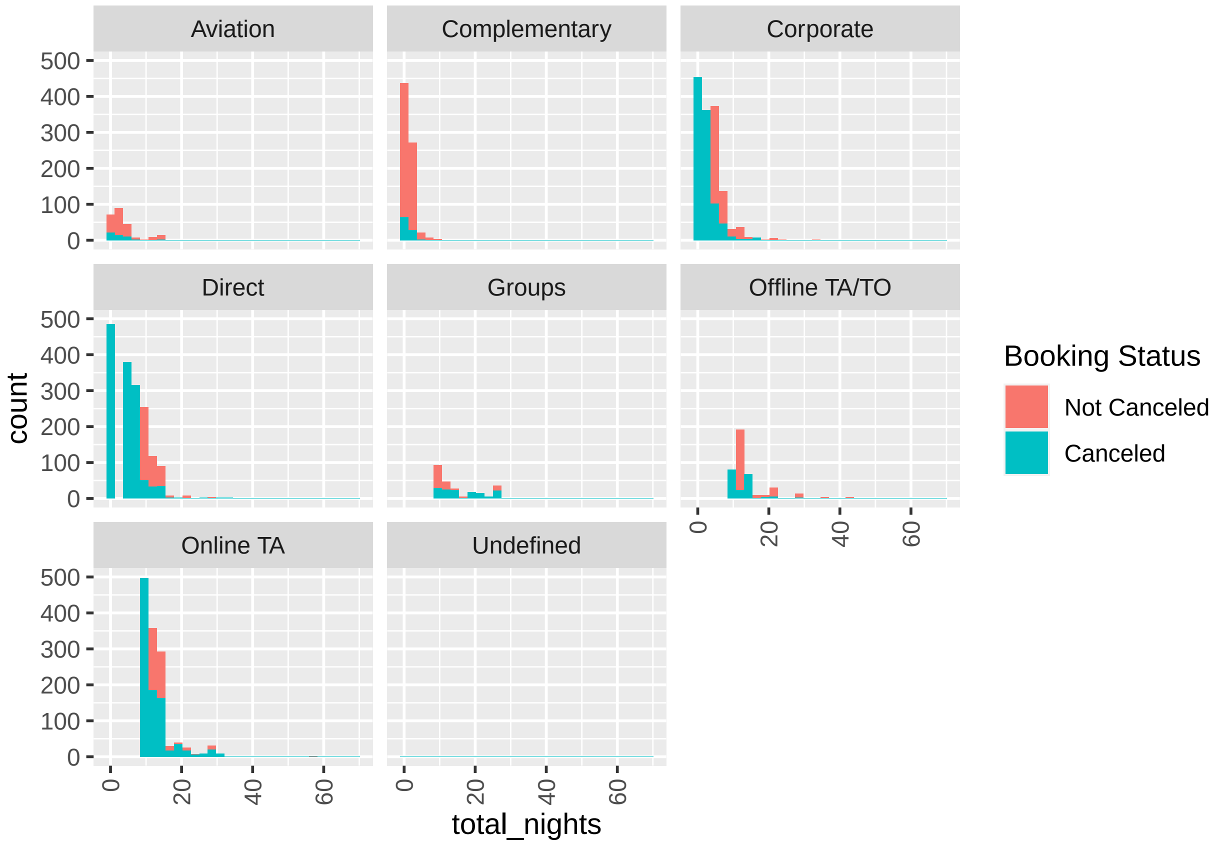
Display booking status per days in waiting list

```
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)
```



3) Create Data Partitions for training and validation purposes

```
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)
```

4) Data Analysis & Modelling

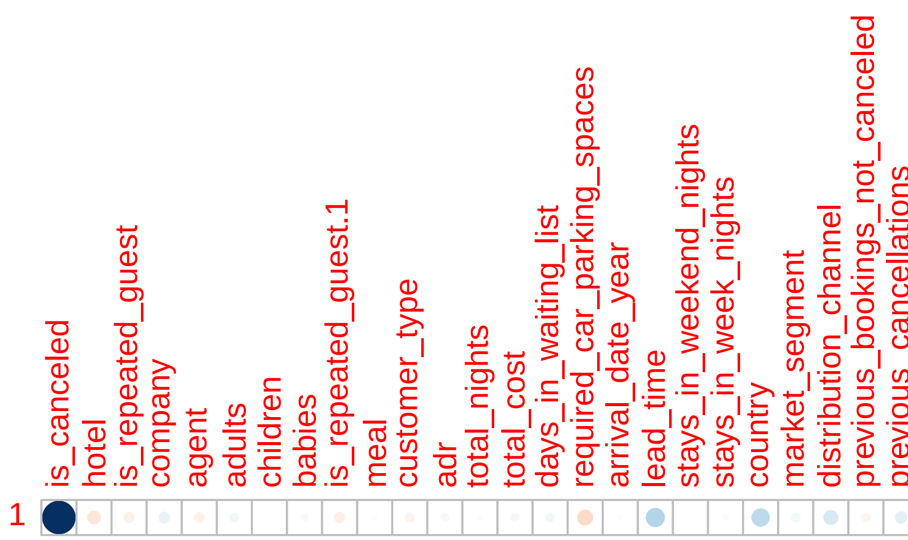
```
conv_numeric <- hotel_train %>% mutate_if(is.factor, as.numeric)
```

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

```
correlations <- cor(conv_numeric$is_canceled,  conv_numeric[,c("is_canceled","hotel","is_repeated_guest
"meal","customer_type", "adr", "total_nights", "total_cost",  "days_in_waiting_list", "required_car_parl
"reserved_room_type", "assigned_room_type", "booking_changes", "deposit_type")])
```

Calculate the correlation coefficient for the target variable "is_canceled"

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



Then plot the correlation coefficient
It is apparent from the plot that the following variables have strong relation to cancellation #### deposit_type, country, distribution_channel, company, lead_time, previous_cancellations, required_car_parking

```
hotel_train <- hotel_train[c("is_canceled", "country", "deposit_type", "distribution_channel", "company
colnames(hotel_train)
```

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

```
## [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
colnames(hotel_valid)
```

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

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

Convert factors to numeric values for modeling purposes

A) glm Model

```
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
```

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

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

Store Model Results

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

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

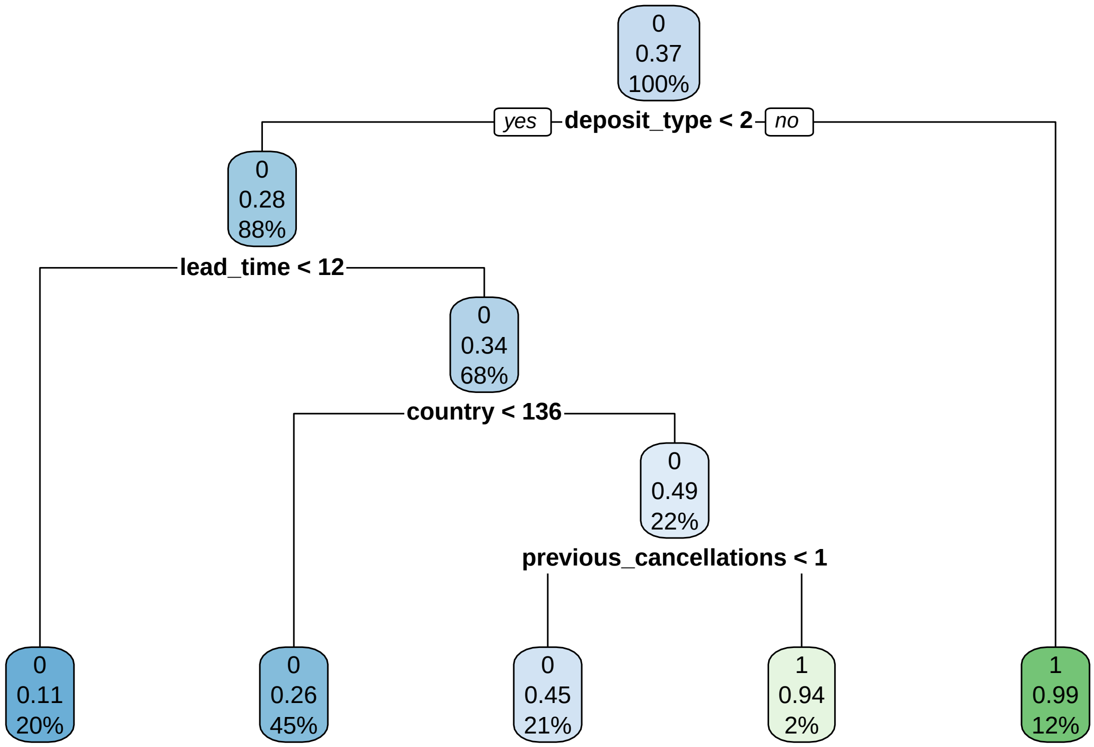
Method_Name	Accuracy
Logestic Regression Model	76.61818

B) Classification Tree Model

```r
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)
```




```r
# 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
```

```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
```

```r
model_results <- bind_rows(model_results, data.frame(Method_Name = "Classification Tree Model", Accurac
model_results
```

Store 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()
```

Method_Name	Accuracy
Logestic Regression Model	76.61818
Classification Tree Model	76.48696

C) Random Forest 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
```

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

```r
model_results <- bind_rows(model_results, data.frame(Method_Name = "Random Forest Model", Accuracy = rf_
model_results
```

Store Model Results

```
##                 Method_Name Accuracy
## 1 Logestic Regression Model 76.61818
## 2 Classification Tree Model 76.48696
## 3       Random Forest Model 78.82663
```

```r
# 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

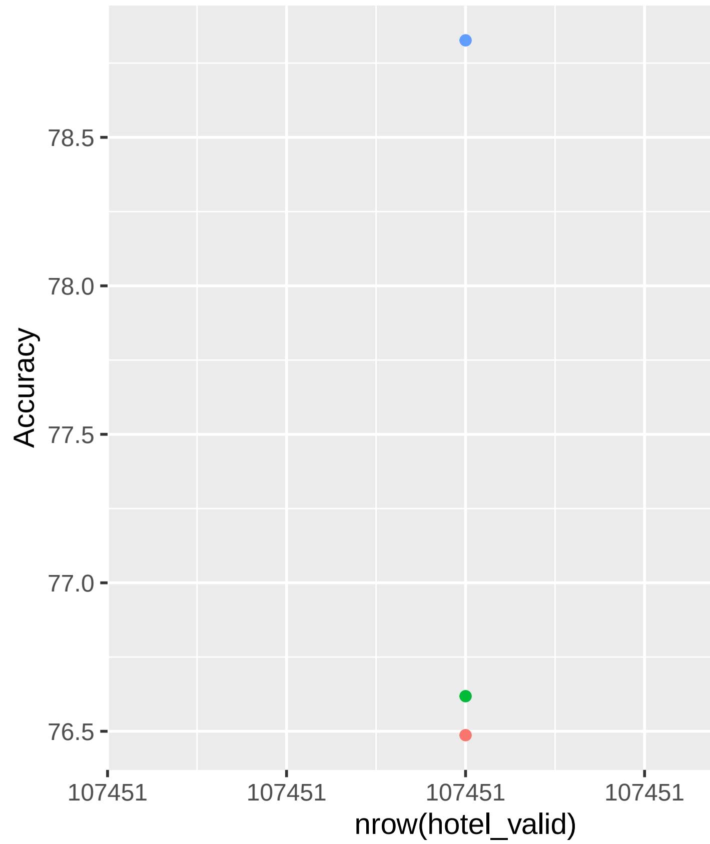
Results

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

```
model_results %>% knitr::kable()
```

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

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



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

Conclusion

In conclusion, based on the available resources the best machine algorithm for predicting future booking cancellations for this project took into consideration seven different factors 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 â€˜is_canceledâ€™. It has been concluded that the Random Forest Model would give the most accurate prediction for future booking cancellations.

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.