Code

```
df<-read.csv("hotel_bookings.csv",header = T)

View(df)

summary(df)

# Features

# hotel: Resort Hotel or City Hotel

# is_canceled: Value indicating if the booking was canceled (1) or not (0)

# lead_time: Number of days between the booking date to the arrival date

# arrival_date_year: Year of arrival

# arrival_date_month: Month of arrival

# arrival_date_week_number: Week number according to year of arrival

# arrival_date_day_of_month: Day of arrival

# stays_in_weekend_nights: Number of weekend nights booked (Saturday or Sunday)

# stays_in_week_nights: Number of week nights booked (Monday to Friday)

# adults: Number of adults

# children: Number of children

# babies: Number of babies

# meal: Type of meal booked

# country: Country of origin

# market_segment: Market segment designation, typically influences the price sensitivity

# distribution_channel: Booking distribution channel, refers to how the booking was made

# is_repeated_guest: Value indication if the booking was from a repeated guest (1) or not (0)

# previous_cancellations: Number of previous cancellations prior to current booking

# previous_bookings_not_canceled: Number of previous booking not canceled prior to current
booking

# reserved_room_type: Code of room type reserved

# assigned_room_type: Code for the type of room assigned to the booking

# booking_changes: Number of changes made to the booking since entering the hotel management
system

# deposit_type: Type of deposit made for the reservation

# agent: ID of the travel agency that made the booking
```

```
# company: ID of the company/organization that made the booking or is responsible for payment
# days_in_waiting_list: Number of days booking was in the waiting list until it was confirmed
# customer_type: Type of booking
# adr: Average Daily Rate (the sum of transactions divided by the number of nights stayed)
# required_car_parking_spaces: Number of car parking spaces requested
# total_of_special_requests: Number of special requests made by the customer
# reservation_status: Last reservation status (Canceled, Check-Out, No-Show)
# reservation_status_date: Date at which the last status was set

#Spliting into train test split
smp_size<-floor(0.75*(nrow(df)))
set.seed(123)
train_ind<-sample(seq_len(nrow(df)),size=smp_size)

train_df<-df[train_ind,]
test_df<-df[-train_ind,]



#Printing no. of rows in each train,test,df
nrow(train_df)
nrow(test_df)
nrow(df)

#Copying dataset for dummy purpose
df2=train_df

colSums(is.na(df))

sum(is.na(df$children))
sum(df$country=="NULL")
sum(df$agent=="NULL")
```

```
sum(df$company=="NULL")

df2=df
sum(is.na(df2$children))
sum(df2$country=="NULL")
sum(df2$agent=="NULL")
sum(df2$company=="NULL")


x<-df2$children
sort(table(x))


df2$children
sort(table(x))

#10 children in outlier
which(grepl(10,df2$children))

grep("children",colnames(df2))

df2[329,11]

dim(df2)
#Removing 10 from dataframe
df2=df2[-c(329),]
dim(df2)

df2[329,11]

##########################
```

```
x<-df2$children
sort(table(x))

#mode
names(table(x))[table(x)==max(table(x))]

#Replacing na with mode
df2$children[is.na(df2$children)]<-names(table(x))[table(x)==max(table(x))]
#checking if any na's
sum(is.na(df2$children))

#checking for country column
x_country<-df2$country
sort(table(x_country))
names(table(x_country))[table(x_country)==max(table(x_country))]

df2$country[df2$country=="NULL"]<-
names(table(x_country))[table(x_country)==max(table(x_country))]

#missing agent can be made 0 since bookings are made private
df2$agent[df2$agent=="NULL"]<-"0"
sum(df2$agent=="NULL")

#similarly for company
df2$company[df2$company=="NULL"]<-"0"
sum(df2$company=="NULL")



str(df2)
class(df2$children)="integer"
```

```
typeof(df2$children)
typeof(df2$babies)
#dataset for correlation matrix
a<-c()
k<-1
for(i in 1:length(colnames(df2))){
  if(typeof(df2[[i]])=="integer"  || typeof(df2[[i]])=="double"){
    a[k]=i
    k=k+1
  }
}
a
temp<-df2[a]
View(temp)
#Total guest
temp["guest_stayed"]=temp[["adults"]]+temp[["children"]]+temp[["babies"]]
View(temp)

#create col with total nights booked
temp["nights_stayed"]=temp[["stays_in_week_nights"]]+temp[["stays_in_weekend_nights"]]

#checking correlation matrix again
data<-cor(temp)
t<-data[,"is_canceled"]
t[order(t,decreasing = TRUE)]

# The strongest positive correlations (0.1 or more) are:
#
#   lead_time
#   previous_cancellations
#
```

```
# The strongest negative correlations (-0.1 or less) are:
#
#   total_of_special_requests
#   required_car_parking_spaces
#   booking_changes

# install.packages("kdensity")
# library("kdensity")


hist(temp$lead_time,xlab = "Lead time days",col = "blue")
?hist



# install.packages("plyr")
# library("plyr")
# count(lead_time_1)
lead_time_1=temp[temp[,"lead_time"]<100,]
nrow(lead_time_1)
lead_time_2=temp[temp[,"lead_time"]<365,]
nrow(lead_time_2)
print(nrow(lead_time_2)-nrow(lead_time_1))
lead_time_3=temp[temp[,"lead_time"]>=365,]
nrow(lead_time_3)


#Cancelation increased if lead time increased
lead_cancel_1=table(lead_time_1$is_canceled)
total<-lead_cancel_1[1]+lead_cancel_1[2]
per<-lead_cancel_1[2]/total
```

```
cat("Percentage of cancelled booking between: 0 to 99:",per)
lead_cancel_2=table(lead_time_2$is_canceled)
total<-lead_cancel_2[1]+lead_cancel_2[2]
per<-lead_cancel_2[2]/total
cat("Percentage of cancelled booking between:100 to 364:",per)
lead_cancel_3=table(lead_time_3$is_canceled)
total<-lead_cancel_3[1]+lead_cancel_3[2]
per<-lead_cancel_3[2]/total
cat("Percentage of cancelled booking between: 365 or more",per)



#Previous cancellations rates
x=table(df2$previous_cancellations)

cat("Never canceled: ",mean(df2$previous_cancellations==0)*100)
cat("Cancelled once: ",mean(df2$previous_cancellations==1)*100)
cat("Cancelled more than 10 times: ",mean(df2$previous_cancellations>=10)*100)

#Booking space canceled on no parking space
x=table(df2$required_car_parking_spaces)
no_park_cancel=df2[df2["is_canceled"]==1,]
no_park_cancel=no_park_cancel[no_park_cancel["required_car_parking_spaces"]==0,]
park_cancel=df2[df2["is_canceled"]==0,]
park_cancel=park_cancel[park_cancel["required_car_parking_spaces"]==0,]
Percentage_BookingCanceled_NoParkingSapce=(nrow(no_park_cancel))/(nrow(no_park_cancel)+nr
ow(park_cancel))*100
cat("Percentage of booking space canceled on no parking
space:",Percentage_BookingCanceled_NoParkingSapce)



#Data Cleaning
df3=train_df
```

```
library(dplyr)

df3["guest_stayed"]=df3[["adults"]]+df3[["children"]]+df3[["babies"]]

View(df3)

df3=df3[df3$guest_stayed>0,]

table(df3$guest_stayed)

#dropping adults,children,babies column

df3[,c("adults","children","babies")]<-list(NULL)

View(df3)


#Defining X_train,y_train

X_train=df3[-c(2)]

View(X_train)

y_train=df3["is_canceled"]


# Removing the Following Columns:

#  Numerical Attributes:

#  arrival_date_year: This category references towards certain years. This could be problematic for instances during years that do not appear in the training data, or perhaps have bias towards certain years specifically due to the unequal amounts of observations in the training data.

#  arrival_date_day_of_month: The column arrival date week of month generalizes this.

#  booking_changes: Could change over time, potentially causing data leakage.

#  days_in_waiting_list: Could constantly change over time. Additionally, there are many instances. This could prevent the model from generalizing.

#  agent & company: Represented by an ID. These columns are uninformative since they contain a substantial amount of various numerical values without having an actual numerical meaning. Since other columns (such as market segment) indicate the type of reservation, these columns won't be needed.

# Categorical Attributes:

#  country: There are many categories, most with few instances. In order to make a model that generalizes, it is better to dismiss this category.

#  assigned_room_type: Similar to reserved_room_type and seems like the reserved room is a more suitable choice.

#  reservation_status: Major data leakage! The categories are Check-Out, Canceled and No-Show. This is exactly what we are trying to predict.
```

```
#     reservation_status_date: This is the date when the reservation status was last changed, and therefore is irrelevant.


num_features=c("lead_time","stays_in_weekend_nights","stays_in_week_nights", "adults", "children", "babies",
          "is_repeated_guest", "previous_cancellations","previous_bookings_not_canceled","adr",
          "required_car_parking_spaces", "total_of_special_requests")
cat_features=c("hotel", "meal", "market_segment",
          "distribution_channel", "reserved_room_type","deposit_type","customer_type")


#Removing features
X_train[,c("arrival_date_year","arrival_date_day_of_month","booking_changes","days_in_waiting_list","agent","company",
       "country","assigned_room_type","reservation_status","reservation_status_date")]<-list(NULL)
View(X_train)



#SC and undefined are same in meal,hence replacing Undefined with meal
library(stringi)
table(X_train$meal)
X_train["meal"]=stri_replace(X_train$meal,"SC",regex = "Undefined")
table(X_train$meal)
#checking na value in hotel column
sum(is.na(X_train$hotel))
X_train[is.na(X_train$hotel),]="City Hotel"
sum(is.na(X_train$hotel))
sum(is.na(X_train))

#install.packages("mltools")
library(mltools)
library(data.table)
#install.packages("caret")
```

```
#library(caret)
# library("reshape2")

#Removing months and arriving week
X_train=X_train[-c(3,4)]
View(X_train)

#Memory limit reached
X_dum=X_train[1:10000,]
X_dum=cbind(ID=1:nrow(X_dum),X_dum)

#One hot encoding
dmy <- dummyVars(X_dum[cat_features], data = X_dum)
trsf <- data.frame(predict(dmy, newdata = X_dum))
View(trsf)
colnames(trsf)

X_dum[,c("hotel", "meal", "market_segment",
        "distribution_channel", "reserved_room_type", "deposit_type", "customer_type")]<-list(NULL)
X_dum=cbind(X_dum,trsf)
X_dum

#Normalize
for(i in 1:length(colnames(X_dum))){
  if(typeof(X_dum[[i]])!="integer"){
    class(X_dum[[i]])="integer"
  }
  #print(typeof(X_dum[[i]]))
}
X_dum=X_dum[-c(1)]
View(X_dum)
```

```
normalize=function(x){
  return((x- min(x)) /(max(x)-min(x)))
}
norm_data=as.data.frame(apply(X_dum,2,normalize))
View(norm_data)
y=y_train
y1=y[1:7000,]
View(y1)
#y_test_dum=y_test_dum["is_canceled"]
norm_data_train=norm_data[1:7000,]
norm_data_test=norm_data[7001:10000,]
View(norm_data_train)
#Contains knn function
library(class)
knn_pred_5 <- knn(train=norm_data_train,test=norm_data_test,cl=y1, k=5)
View(knn_pred_5)
#for confusion matrix
library(caret)
confusionMatrix(table(knn_pred_5,y[7001:10000,1]))
View(knn_pred_5)


#logistic regression
logistic_data_train=norm_data_train
logistic_data_train=cbind(logistic_data_train,y1)
View(logistic_data_train)
logistic_data_test=norm_data_test
target=y[7001:10000,1]
logistic_data_test=cbind(logistic_data_test,target)
View(logistic_data_test)
y_t=logistic_data_test["target"]
```

```
logistic<-glm(y1~lead_time+previous_cancellations,logistic_data_train,family = "binomial")
summary(logistic)
res<-predict(logistic,logistic_data_test,type="response")
View(res)
confmatrix1<-table(Actual_value=logistic_data_test$target,Predicted_value=res>0.2)
confmatrix2<-table(Actual_value=logistic_data_test$target,Predicted_value=res>0.3)
confmatrix3<-table(Actual_value=logistic_data_test$target,Predicted_value=res>0.4)
confmatrix1
confmatrix2
confmatrix3
```

Output

```
                Max.   :1.00000   Max.   :26.00000

 previous_bookings_not_canceled reserved_room_type assigned_room_type
 Min.   : 0.0000                Length:119390      Length:119390
 1st Qu.: 0.0000                Class :character   Class :character
 Median : 0.0000                Mode  :character   Mode  :character
 Mean   : 0.1371
 3rd Qu.: 0.0000
 Max.   :72.0000

 booking_changes   deposit_type         agent           company
 Min.   : 0.0000   Length:119390      Length:119390      Length:119390
 1st Qu.: 0.0000   Class :character   Class :character   Class :character
 Median : 0.0000   Mode  :character   Mode  :character   Mode  :character
```

```
Mean   : 0.2211
3rd Qu.: 0.0000
Max.   :21.0000

days_in_waiting_list customer_type         adr
Min.   :  0.000     Length:119390     Min.   :  -6.38
1st Qu.:  0.000     Class :character  1st Qu.:  69.29
Median :  0.000     Mode  :character  Median :  94.58
Mean   :  2.321                       Mean   : 101.83
3rd Qu.:  0.000                       3rd Qu.: 126.00
Max.   :391.000                       Max.   :5400.00

required_car_parking_spaces total_of_special_requests reservation_status
Min.   :0.00000             Min.   :0.0000            Length:119390
1st Qu.:0.00000             1st Qu.:0.0000            Class :character
Median :0.00000             Median :0.0000            Mode  :character
Mean   :0.06252             Mean   :0.5714
3rd Qu.:0.00000             3rd Qu.:1.0000
Max.   :8.00000             Max.   :5.0000

reservation_status_date
Length:119390
Class :character
Mode  :character




> # Features
> #
> # hotel: Resort Hotel or City Hotel
```

> # is_canceled: Value indicating if the booking was canceled (1) or not (0)

> # lead_time: Number of days between the booking date to the arrival date

> # arrival_date_year: Year of arrival

> # arrival_date_month: Month of arrival

> # arrival_date_week_number: Week number according to year of arrival

> # arrival_date_day_of_month: Day of arrival

> # stays_in_weekend_nights: Number of weekend nights booked (Saturday or Sunday)

> # stays_in_week_nights: Number of week nights booked (Monday to Friday)

> # adults: Number of adults

> # children: Number of children

> # babies: Number of babies

> # meal: Type of meal booked

> # country: Country of origin

> # market_segment: Market segment designation, typically influences the price sensitivity

> # distribution_channel: Booking distribution channel, refers to how the booking was made

> # is_repeated_guest: Value indication if the booking was from a repeated guest (1) or not (0)

> # previous_cancellations: Number of previous cancellations prior to current booking

> # previous_bookings_not_canceled: Number of previous booking not canceled prior to current booking

> # reserved_room_type: Code of room type reserved

> # assigned_room_type: Code for the type of room assigned to the booking

> # booking_changes: Number of changes made to the booking since entering the hotel management system

> # deposit_type: Type of deposit made for the reservation

> # agent: ID of the travel agency that made the booking

> # company: ID of the company/organization that made the booking or is responsible for payment

> # days_in_waiting_list: Number of days booking was in the waiting list until it was confirmed

> # customer_type: Type of booking

> # adr: Average Daily Rate (the sum of transactions divided by the number of nights stayed)

> # required_car_parking_spaces: Number of car parking spaces requested

> # total_of_special_requests: Number of special requests made by the customer

> # reservation_status: Last reservation status (Canceled, Check-Out, No-Show)

```
> # reservation_status_date: Date at which the last status was set
>
> #Spliting into train test split
> smp_size<-floor(0.75*(nrow(df)))
> set.seed(123)
> train_ind<-sample(seq_len(nrow(df)),size=smp_size)
>
> train_df<-df[train_ind,]
> test_df<-df[-train_ind,]
>
>
> #Printing no. of rows in each train,test,df
> nrow(train_df)
[1] 89542
> nrow(test_df)
[1] 29848
> nrow(df)
[1] 119390
>
> #Copying dataset for dummy purpose
> df2=train_df
>
> colSums(is.na(df))
                    hotel                is_canceled
                        0                          0
                lead_time          arrival_date_year
                        0                          0
       arrival_date_month     arrival_date_week_number
                        0                          0
  arrival_date_day_of_month      stays_in_weekend_nights
                        0                          0
```

```
      stays_in_week_nights                    adults
                         0                         0
                  children                    babies
                         4                         0
                      meal                   country
                         0                         0
             market_segment         distribution_channel
                         0                         0
           is_repeated_guest       previous_cancellations
                         0                         0
previous_bookings_not_canceled           reserved_room_type
                         0                         0
          assigned_room_type             booking_changes
                         0                         0
               deposit_type                      agent
                         0                         0
                    company          days_in_waiting_list
                         0                         0
              customer_type                       adr
                         0                         0
  required_car_parking_spaces     total_of_special_requests
                         0                         0
         reservation_status       reservation_status_date
                         0                         0
>
> sum(is.na(df$children))
[1] 4
> sum(df$country=="NULL")
[1] 488
> sum(df$agent=="NULL")
[1] 16340
```

```
> sum(df$company=="NULL")
[1] 112593
>
> df2=df
> sum(is.na(df2$children))
[1] 4
> sum(df2$country=="NULL")
[1] 488
> sum(df2$agent=="NULL")
[1] 16340
> sum(df2$company=="NULL")
[1] 112593
>
>
> x<-df2$children
> sort(table(x))
x
   10     3     2     1     0
    1    76  3652  4861 110796
>
>
> df2$children
  [1]  0 0 0  0 0 0 0 0  0 0 0 0 0 1  0 0 0 0 0 0  0 0 0 0 0  0
 [27]  0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 2  0 0 0 0 0 0
 [53]  0 0 0 2  0 0 0 0 0 0 0 0 0 2 0  0 0 0 0 0 0 0 0 0 0 0 0
 [79]  0 0 0 0  0 0 0 0 0 1  0 0 0 0 0 0  0 0 0 0 0  1 0 0 0 2 0
[105]  0 2 0 0 0  0 0 0 0 0 0  0 0 0 0 1 2  0 0 0 0 0 0  0 0 0 0 0
[131]  0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 2 0 2 0 0 0 2 0
[157]  0 0 0 0 2  0 0 0 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 0
[183]  0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 2 0 0 0 0 0 0
[209]  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 2 0 2 0
```

```
[235]  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
[261]  0  0  0  0  0  2  0  0  0  0  0  1  1  1  0  0  0  0  1  0  1  0  0  0  0  0
[287]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  2  0  1  0  0
[313]  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  2  0  0  0  0  0  0
[339]  0  0  0  2  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[365]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
[391]  0  0  0  0  0  1  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0
[417]  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  2  2  0
[443]  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[469]  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0  2  0  2  0  0  0  0  0  0  0
[495]  0  0  1  1  0  0  0  0  0  0  0  1  0  0  0  0  1  2  0  0  0  0  0  0  2  0
[521]  0  1  0  2  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[547]  0  0  0  0  2  0  0  0  0  0  0  1  2  0  0  0  2  0  0  0  0  2  0  0  0  0
[573]  0  0  0  0  0  2  0  0  2  0  0  0  1  2  0  0  0  1  0  1  0  0  1  0  0  0
[599]  0  0  0  0  2  0  0  0  0  0  0  1  0  0  0  0  1  2  0  0  2  0  0  0  0  0
[625]  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0
[651]  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
[677]  0  0  0  1  0  0  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[703]  0  0  1  0  0  0  0  0  0  1  0  0  1  0  2  0  0  0  0  0  0  2  0  2  0  0
[729]  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
[755]  0  0  0  0  0  0  1  1  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[781]  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  2  1  0
[807]  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0
[833]  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  2  0  0  2  2
[859]  0  0  0  1  0  0  0  0  2  1  2  0  0  0  0  0  0  1  0  0  0  0  0  0  2  0
[885]  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[911]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  1  0  0
[937]  2  1  0  0  0  0  0  2  0  0  0  0  0  0  2  1  0  0  0  0  0  0  0  0  0  2
[963]  0  0  0  0  0  0  0  0  0  2  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  1
[989]  0  0  0  0  0  0  0  0  0  0  1  0
[ reached getOption("max.print") -- omitted 118390 entries ]
```

```
> sort(table(x))
x
   10     3     2     1     0
    1    76  3652  4861 110796
>
> #10 children in outlier
> which(grepl(10,df2$children))
[1] 329
>
> grep("children",colnames(df2))
[1] 11
>
> df2[329,11]
[1] 10
>
> dim(df2)
[1] 119390     32
> #Removing 10 from dataframe
> df2=df2[-c(329),]
> dim(df2)
[1] 119389     32
>
> df2[329,11]
[1] 0
>
> ############################
>
> x<-df2$children
> sort(table(x))
x
    3     2     1     0
```

```
  76   3652   4861 110796
>
> #mode
> names(table(x))[table(x)==max(table(x))]
[1] "0"
>
> #Replacing na with mode
> df2$children[is.na(df2$children)]<-names(table(x))[table(x)==max(table(x))]
> #checking if any na's
> sum(is.na(df2$children))
[1] 0
>
> #checking for country column
> x_country<-df2$country
> sort(table(x_country))
x_country
 AIA  ASM  ATF  BDI  BFA  BHS  BWA  CYM  DJI  DMA  FJI  GUY  HND  KIR
   1    1    1    1    1    1    1    1    1    1    1    1    1    1
 LCA  MDG  MLI  MMR  MRT  NAM  NCL  NIC  NPL  PLW  PYF  SDN  SLE  SMR
   1    1    1    1    1    1    1    1    1    1    1    1    1    1
 UMI  VGB  ABW  ATA  COM  GLP  IMN  KHM  KNA  LAO  MWI  MYT  RWA  SLV
   1    1    2    2    2    2    2    2    2    2    2    2    2    2
 STP  SYC  TGO  UGA  ZMB  BEN  ETH  GGY  LIE  SYR  TMP  BRB  GAB  GHA
   2    2    2    2    2    3    3    3    3    3    3    4    4    4
 GTM  MCO  PRY  UZB  ZWE  BHR  CAF  FRO  MNE  SUR  TZA  CIV  JAM  KEN
   4    4    4    4    4    5    5    5    5    5    5    6    6    6
 AND  LKA  MUS  ARM  CUB  JEY  LBY  VNM  GNB  PAN  TJK  BOL  CMR  MKD
   7    7    7    8    8    8    8    8    9    9    9   10   10   10
 SEN  ALB  BGD  MDV  PRI  BIH  DOM  IRQ  PAK  QAT  KWT  MAC  AZE  GIB
  11   12   12   12   12   13   14   14   14   15   16   16   17   18
 MLT  OMN  CRI  KAZ  JOR  GEO  CPV  BLR  VEN  ECU  MYS  HKG  PER  LBN
```

```
 18   18   19   19   21   22   24   26   26   27   28   29   29   31
EGY  URY  NGA  IDN  SGP  TUN  PHL  SAU  ARE  CYP  TWN  LVA  ISL  SVN
 32   32   34   35   39   39   40   48   51   51   51   55   57   57
THA  CHL  SVK  MOZ  UKR  COL  NZL  BGR  ZAF  LTU  EST  IRN  MEX  HRV
 59   65   65   67   68   71   74   75   80   81   83   83   85  100
SRB  DZA  GRC  KOR  IND  CZE  JPN  ARG  HUN  TUR  MAR  LUX  AGO  AUS
101  103  128  133  152  171  197  214  230  248  259  287  362  426
DNK  FIN NULL  ROU  NOR  RUS  ISR  POL  CHN  SWE  AUT   CN  CHE  USA
435  447  488  500  607  632  669  919  999 1024 1263 1279 1730 2097
NLD  BRA  BEL  IRL  ITA  DEU  ESP  FRA  GBR  PRT
2104 2224 2342 3375 3766 7287 8568 10415 12129 48589
> names(table(x_country))[table(x_country)==max(table(x_country))]
[1] "PRT"
>
> df2$country[df2$country=="NULL"]<-
names(table(x_country))[table(x_country)==max(table(x_country))]
>
> #missing agent can be made 0 since bookings are made private
> df2$agent[df2$agent=="NULL"]<-"0"
> sum(df2$agent=="NULL")
[1] 0
>
> #similarly for company
> df2$company[df2$company=="NULL"]<-"0"
> sum(df2$company=="NULL")
[1] 0
>
>
> str(df2)
'data.frame':    119389 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 : chr "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 "0" "0" "0" "304" ...

$ company : chr "0" "0" "0" "0" ...

$ 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" ...

```
> class(df2$children)="integer"
> typeof(df2$children)
[1] "integer"
> typeof(df2$babies)
[1] "integer"
> #dataset for correlation matrix
> a<-c()
> k<-1
> for(i in 1:length(colnames(df2))){
+   if(typeof(df2[[i]])=="integer"  || typeof(df2[[i]])=="double"){
+     a[k]=i
+     k=k+1
+   }
+ }
> a
 [1]  2  3  4  6  7  8  9 10 11 12 17 18 19 22 26 28 29 30
> temp<-df2[a]
> View(temp)
> #Total guest
> temp["guest_stayed"]=temp[["adults"]]+temp[["children"]]+temp[["babies"]]
> View(temp)
>
> #create col with total nights booked
> temp["nights_stayed"]=temp[["stays_in_week_nights"]]+temp[["stays_in_weekend_nights"]]
>
> #checking correlation matrix again
> data<-cor(temp)
> t<-data[,"is_canceled"]
> t[order(t,decreasing = TRUE)]
                is_canceled                lead_time
                1.000000000                0.293130709
```

```
      previous_cancellations                    adults
                 0.110134724                0.060014948
       days_in_waiting_list                       adr
                 0.054187654                0.047550243
              guest_stayed          stays_in_week_nights
                 0.046407938                0.024723488
             nights_stayed            arrival_date_year
                 0.017735523                0.016678017
   arrival_date_week_number                    children
                 0.008146651                0.004777499
    stays_in_weekend_nights      arrival_date_day_of_month
                -0.001824760               -0.006125404
                    babies previous_bookings_not_canceled
                -0.032490431               -0.057357134
         is_repeated_guest             booking_changes
                -0.084792051               -0.144416296
 required_car_parking_spaces     total_of_special_requests
                -0.195496479               -0.234665636
>
> # The strongest positive correlations (0.1 or more) are:
> #
> #   lead_time
> #   previous_cancellations
> #
> # The strongest negative correlations (-0.1 or less) are:
> #
> #   total_of_special_requests
> #   required_car_parking_spaces
> #   booking_changes
>
> # install.packages("kdensity")
```

```
> # library("kdensity")
>
>
> hist(temp$lead_time,xlab = "Lead time days",col = "blue")
> ?hist
>
>
>
> # install.packages("plyr")
> # library("plyr")
> # count(lead_time_1)
> lead_time_1=temp[temp[,"lead_time"]<100,]
> nrow(lead_time_1)
[1] 71682
> lead_time_2=temp[temp[,"lead_time"]<365,]
> nrow(lead_time_2)
[1] 116176
> print(nrow(lead_time_2)-nrow(lead_time_1))
[1] 44494
> lead_time_3=temp[temp[,"lead_time"]>=365,]
> nrow(lead_time_3)
[1] 3213
>
>
> #Cancelation increased if lead time increased
> lead_cancel_1=table(lead_time_1$is_canceled)
> total<-lead_cancel_1[1]+lead_cancel_1[2]
> per<-lead_cancel_1[2]/total
> cat("Percentage of cancelled booking  between: 0 to 99:",per)
Percentage of cancelled booking between: 0 to 99: 0.2768756>
lead_cancel_2=table(lead_time_2$is_canceled)
```

```
> total<-lead_cancel_2[1]+lead_cancel_2[2]

> per<-lead_cancel_2[2]/total

> cat("Percentage of cancelled booking between:100 to 364:",per)

Percentage of cancelled booking between:100 to 364: 0.3618561>
lead_cancel_3=table(lead_time_3$is_canceled)

> total<-lead_cancel_3[1]+lead_cancel_3[2]

> per<-lead_cancel_3[2]/total

> cat("Percentage of cancelled booking between: 365 or more",per)

Percentage of cancelled booking between: 365 or more 0.6797386>

>

> #Previous cancellations rates

> x=table(df2$previous_cancellations)

>

> cat("Never canceled: ",mean(df2$previous_cancellations==0)*100)

Never canceled:  94.56901> cat("Cancelled once: ",mean(df2$previous_cancellations==1)*100)

Cancelled once:  5.068306> cat("Cancelled more than 10 times:
",mean(df2$previous_cancellations>=10)*100)

Cancelled more than 10 times:  0.1507677>

> #Booking space canceled on no parking space

> x=table(df2$required_car_parking_spaces)

> no_park_cancel=df2[df2["is_canceled"]==1,]

> no_park_cancel=no_park_cancel[no_park_cancel["required_car_parking_spaces"]==0,]

> park_cancel=df2[df2["is_canceled"]==0,]

> park_cancel=park_cancel[park_cancel["required_car_parking_spaces"]==0,]

>
Percentage_BookingCanceled_NoParkingSapce=(nrow(no_park_cancel))/(nrow(no_park_cancel)+nr
ow(park_cancel))*100

> cat("Percentage of booking space canceled on no parking
space:",Percentage_BookingCanceled_NoParkingSapce)

Percentage of booking space canceled on no parking space: 39.49434>

>

> #Data Cleaning

> df3=train_df
```

```
> library(dplyr)
> df3["guest_stayed"]=df3[["adults"]]+df3[["children"]]+df3[["babies"]]
> View(df3)
> df3=df3[df3$guest_stayed>0,]
> table(df3$guest_stayed)

   1    2    3    4    5    6   10   12   20   26   27   40   50   55
16923 61580  7836  2960   104    1    2    1    2    2    2    1    1    1
> #dropping adults,children,babies column
> df3[,c("adults","children","babies")]<-list(NULL)
> View(df3)
>
> #Defining X_train,y_train
> X_train=df3[-c(2)]
> View(X_train)
> y_train=df3["is_canceled"]
>
> # Removing the Following Columns:
> #   Numerical Attributes:
> #     arrival_date_year: This category references towards certain years. This could be problematic for instances during years that do not appear in the training data, or perhaps have bias towards certain years specifically due to the unequal amounts of observations in the training data.
> #     arrival_date_day_of_month: The column arrival date week of month generalizes this.
> #     booking_changes: Could change over time, potentially causing data leakage.
> #     days_in_waiting_list: Could constantly change over time. Additionally, there are many instances. This could prevent the model from generalizing.
> #     agent & company: Represented by an ID. These columns are uninformative since they contain a substantial amount of various numerical values without having an actual numerical meaning. Since other columns (such as market segment) indicate the type of reservation, these columns won't be needed.
> # Categorical Attributes:
> #     country: There are many categories, most with few instances. In order to make a model that generalizes, it is better to dismiss this category.
```

```
> #     assigned_room_type: Similar to reserved_room_type and seems like the reserved room is a more suitable choice.

> #     reservation_status: Major data leakage! The categories are Check-Out, Canceled and No-Show. This is exactly what we are trying to predict.

> #     reservation_status_date: This is the date when the reservation status was last changed, and therefore is irrelevant.

>

> num_features=c("lead_time","stays_in_weekend_nights","stays_in_week_nights","adults",
"children","babies",

+             "is_repeated_guest","previous_cancellations","previous_bookings_not_canceled","adr",

+             "required_car_parking_spaces","total_of_special_requests")

> cat_features=c("hotel","meal","market_segment",

+             "distribution_channel","reserved_room_type","deposit_type","customer_type")

>

> #Removing features

>
X_train[,c("arrival_date_year","arrival_date_day_of_month","booking_changes","days_in_waiting_li
st","agent","company",

+          "country","assigned_room_type","reservation_status","reservation_status_date")]<-
list(NULL)

> View(X_train)

>

>

> #SC and undefined are same in meal,hence replacing Undefined with meal

> library(stringi)

> table(X_train$meal)


      BB       FB       HB       SC Undefined

   69128      622    10885     7911      870

> X_train["meal"]=stri_replace(X_train$meal,"SC",regex = "Undefined")

> table(X_train$meal)


  BB   FB   HB   SC
```

```
69128   622 10885  8781
> #checking na value in hotel column
> sum(is.na(X_train$hotel))
[1] 2
> X_train[is.na(X_train$hotel),]="City Hotel"
> sum(is.na(X_train$hotel))
[1] 0
> sum(is.na(X_train))
[1] 0
>
> #install.packages("mltools")
> library(mltools)
> library(data.table)
> #install.packages("caret")
> #library(caret)
> # library("reshape2")
>
> #Removing months and arriving week
> X_train=X_train[-c(3,4)]
> View(X_train)
>
> #Memory limit reached
> X_dum=X_train[1:10000,]
> X_dum=cbind(ID=1:nrow(X_dum),X_dum)
>
> #One hot encoding
> dmy <- dummyVars(X_dum[cat_features], data = X_dum)
> trsf <- data.frame(predict(dmy, newdata = X_dum))
> View(trsf)
> colnames(trsf)
 [1] "mealBB"                    "mealFB"
```

```
 [3] "mealHB"                    "mealSC"
 [5] "market_segmentAviation"      "market_segmentComplementary"
 [7] "market_segmentCorporate"     "market_segmentDirect"
 [9] "market_segmentGroups"        "market_segmentOffline.TA.TO"
[11] "market_segmentOnline.TA"     "distribution_channelCorporate"
[13] "distribution_channelDirect"  "distribution_channelGDS"
[15] "distribution_channelTA.TO"   "reserved_room_typeA"
[17] "reserved_room_typeB"         "reserved_room_typeC"
[19] "reserved_room_typeD"         "reserved_room_typeE"
[21] "reserved_room_typeF"         "reserved_room_typeG"
[23] "reserved_room_typeH"         "deposit_typeNo.Deposit"
[25] "deposit_typeNon.Refund"      "deposit_typeRefundable"
[27] "customer_typeContract"       "customer_typeGroup"
[29] "customer_typeTransient"      "customer_typeTransient.Party"
>
> X_dum[,c("hotel", "meal", "market_segment",
+          "distribution_channel", "reserved_room_type", "deposit_type", "customer_type")]<-
list(NULL)
> X_dum=cbind(X_dum,trsf)
> X_dum
      ID lead_time stays_in_weekend_nights stays_in_week_nights is_repeated_guest
51663   1      158                    0                   2                   0
57870   2       81                    2                   1                   0
2986    3       79                    1                   5                   0
29925   4       49                    2                   1                   0
95246   5        9                    2                   1                   0
103065  6        4                    0                   4                   0
68293   7      104                    1                   2                   0
62555   8      253                    2                   1                   0
45404   9       72                    0                   3                   0
65161  10       38                    2                   5                   0
```

46435	11	18	0	1	0
104474	12	1	0	2	0
9642	13	248	2	3	0
59134	14	74	0	2	0
52132	15	113	0	3	0
96849	16	9	0	1	0
14183	17	80	2	2	0
15180	18	246	2	5	0
27168	19	108	1	5	0
89709	20	212	2	3	0
9097	21	117	2	5	0
30538	22	0	0	1	0
56219	23	419	0	2	0
94517	24	165	2	4	0

	previous_cancellations	previous_bookings_not_canceled	adr
51663	0	0	130
57870	0	0	119.07
2986	0	0	56.16
29925	0	0	64.8
95246	0	0	148
103065	0	0	88.26
68293	0	0	160
62555	0	0	129.6
45404	0	0	85.67
65161	0	0	80.87
46435	0	0	109
104474	0	0	104.5
9642	0	0	55.8
59134	0	0	132.3
52132	0	0	100
96849	0	0	168

14183	1	0	195
15180	1	0	58.95
27168	0	0	234
89709	0	0	78
9097	0	0	66.02
30538	0	0	93
56219	0	0	62
94517	0	0	72.25

	required_car_parking_spaces	total_of_special_requests	guest_stayed	mealBB
51663	0	0	1	1
57870	0	0	2	1
2986	0	1	2	1
29925	0	3	2	0
95246	0	2	2	0
103065	0	2	2	0
68293	0	0	1	1
62555	0	0	2	0
45404	0	1	2	1
65161	0	0	2	0
46435	0	1	2	1
104474	0	0	3	1
9642	0	0	1	1
59134	0	0	2	1
52132	0	0	1	1
96849	1	3	2	1
14183	0	0	2	0
15180	0	0	2	1
27168	0	1	3	1
89709	0	1	2	1
9097	0	0	2	1
30538	0	1	2	0

56219	0	0	2	1
94517	0	2	2	1

	mealFB	mealHB	mealSC	market_segmentAviation	market_segmentComplementary
51663	0	0	0	0	0
57870	0	0	0	0	0
2986	0	0	0	0	0
29925	0	1	0	0	0
95246	0	0	1	0	0
103065	0	0	1	0	0
68293	0	0	0	0	0
62555	0	1	0	0	0
45404	0	0	0	0	0
65161	0	0	1	0	0
46435	0	0	0	0	0
104474	0	0	0	0	0
9642	0	0	0	0	0
59134	0	0	0	0	0
52132	0	0	0	0	0
96849	0	0	0	0	0
14183	1	0	0	0	0
15180	0	0	0	0	0
27168	0	0	0	0	0
89709	0	0	0	0	0
9097	0	0	0	0	0
30538	0	1	0	0	0
56219	0	0	0	0	0
94517	0	0	0	0	0

	market_segmentCorporate	market_segmentDirect	market_segmentGroups
51663	0	0	1
57870	0	0	0
2986	0	0	0

29925	0	0	0
95246	0	0	0
103065	0	0	0
68293	0	0	1
62555	0	0	0
45404	0	0	1
65161	0	0	0
46435	0	0	0
104474	0	0	0
9642	0	0	0
59134	0	0	0
52132	1	0	0
96849	0	0	0
14183	0	1	0
15180	0	0	0
27168	0	0	0
89709	0	0	0
9097	0	0	0
30538	0	0	0
56219	0	0	1
94517	0	0	0

	market_segmentOffline.TA.TO	market_segmentOnline.TA
51663	0	0
57870	0	1
2986	1	0
29925	0	1
95246	0	1
103065	0	1
68293	0	0
62555	0	1
45404	0	0

65161	0	1
46435	0	1
104474	0	1
9642	0	1
59134	0	1
52132	0	0
96849	0	1
14183	0	0
15180	1	0
27168	0	1
89709	1	0
9097	0	1
30538	0	1
56219	0	0
94517	1	0

	distribution_channelCorporate	distribution_channelDirect
51663	0	0
57870	0	0
2986	0	0
29925	0	0
95246	0	0
103065	0	0
68293	0	0
62555	0	0
45404	0	0
65161	0	0
46435	0	0
104474	0	0
9642	0	0
59134	0	0
52132	0	0

96849	0	0
14183	0	1
15180	0	0
27168	0	0
89709	0	0
9097	0	0
30538	0	0
56219	0	0
94517	0	0

	distribution_channelGDS	distribution_channelTA.TO	reserved_room_typeA
51663	0	1	1
57870	0	1	0
2986	0	1	0
29925	0	1	1
95246	0	1	1
103065	0	1	1
68293	0	1	1
62555	0	1	0
45404	0	1	1
65161	0	1	1
46435	0	1	1
104474	0	1	1
9642	0	1	0
59134	0	1	0
52132	0	1	1
96849	0	1	1
14183	0	0	0
15180	0	1	1
27168	0	1	0
89709	0	1	1
9097	0	1	0

30538	0	1	0
56219	0	1	1
94517	0	1	1

	reserved_room_typeB	reserved_room_typeC	reserved_room_typeD
51663	0	0	0
57870	0	0	1
2986	0	0	0
29925	0	0	0
95246	0	0	0
103065	0	0	0
68293	0	0	0
62555	0	0	1
45404	0	0	0
65161	0	0	0
46435	0	0	0
104474	0	0	0
9642	0	0	0
59134	0	0	1
52132	0	0	0
96849	0	0	0
14183	0	0	0
15180	0	0	0
27168	0	0	1
89709	0	0	0
9097	0	0	0
30538	0	0	0
56219	0	0	0
94517	0	0	0

	reserved_room_typeE	reserved_room_typeF	reserved_room_typeG
51663	0	0	0
57870	0	0	0

2986	1	0	0
29925	0	0	0
95246	0	0	0
103065	0	0	0
68293	0	0	0
62555	0	0	0
45404	0	0	0
65161	0	0	0
46435	0	0	0
104474	0	0	0
9642	1	0	0
59134	0	0	0
52132	0	0	0
96849	0	0	0
14183	0	1	0
15180	0	0	0
27168	0	0	0
89709	0	0	0
9097	1	0	0
30538	1	0	0
56219	0	0	0
94517	0	0	0

	reserved_room_typeH	deposit_typeNo.Deposit	deposit_typeNon.Refund
51663	0	0	1
57870	0	1	0
2986	0	1	0
29925	0	1	0
95246	0	1	0
103065	0	1	0
68293	0	0	1
62555	0	1	0

45404	0	1	0
65161	0	1	0
46435	0	1	0
104474	0	1	0
9642	0	1	0
59134	0	1	0
52132	0	0	1
96849	0	1	0
14183	0	1	0
15180	0	1	0
27168	0	1	0
89709	0	1	0
9097	0	1	0
30538	0	1	0
56219	0	0	1
94517	0	1	0

	deposit_typeRefundable	customer_typeContract	customer_typeGroup
51663	0	0	0
57870	0	0	0
2986	0	1	0
29925	0	0	0
95246	0	0	0
103065	0	0	0
68293	0	0	0
62555	0	0	0
45404	0	0	0
65161	0	0	0
46435	0	0	0
104474	0	0	0
9642	0	0	0
59134	0	0	0

52132	0	0	0
96849	0	0	0
14183	0	0	0
15180	0	0	0
27168	0	0	0
89709	0	0	0
9097	0	0	0
30538	0	0	0
56219	0	0	0
94517	0	0	0

	customer_typeTransient	customer_typeTransient.Party
51663	1	0
57870	1	0
2986	0	0
29925	1	0
95246	1	0
103065	1	0
68293	1	0
62555	1	0
45404	0	1
65161	1	0
46435	1	0
104474	1	0
9642	1	0
59134	1	0
52132	1	0
96849	1	0
14183	1	0
15180	0	1
27168	1	0
89709	0	1

```
9097                0                   1
30538               1                   0
56219               1                   0
94517               0                   1
 [ reached 'max' / getOption("max.print") -- omitted 9976 rows ]
>
> #Normalize
> for(i in 1:length(colnames(X_dum))){
+   if(typeof(X_dum[[i]])!="integer"){
+     class(X_dum[[i]])="integer"
+   }
+   #print(typeof(X_dum[[i]]))
+ }
> X_dum=X_dum[-c(1)]
> View(X_dum)
> normalize=function(x){
+   return((x- min(x)) /(max(x)-min(x)))
+ }
> norm_data=as.data.frame(apply(X_dum,2,normalize))
> View(norm_data)
> y=y_train
> y1=y[1:7000,]
> View(y1)
> #y_test_dum=y_test_dum["is_canceled"]
> norm_data_train=norm_data[1:7000,]
> norm_data_test=norm_data[7001:10000,]
> View(norm_data_train)
> #Contains knn function
> library(class)
> knn_pred_5 <- knn(train=norm_data_train,test=norm_data_test,cl=y1, k=5)
> View(knn_pred_5)
```

```
> #for confusion matrix
> library(caret)
> confusionMatrix(table(knn_pred_5,y[7001:10000,1]))
Confusion Matrix and Statistics


knn_pred_5    0    1
         0 1616  406
         1  256  722

               Accuracy : 0.7793
                 95% CI : (0.7641, 0.7941)
    No Information Rate : 0.624
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.517

 Mcnemar's Test P-Value : 6.995e-09

            Sensitivity : 0.8632
            Specificity : 0.6401
         Pos Pred Value : 0.7992
         Neg Pred Value : 0.7382
             Prevalence : 0.6240
         Detection Rate : 0.5387
   Detection Prevalence : 0.6740
      Balanced Accuracy : 0.7517

       'Positive' Class : 0

> View(knn_pred_5)
```

```
>
>
> #logistic regression
> logistic_data_train=norm_data_train
> logistic_data_train=cbind(logistic_data_train,y1)
> View(logistic_data_train)
> logistic_data_test=norm_data_test
> target=y[7001:10000,1]
> logistic_data_test=cbind(logistic_data_test,target)
> View(logistic_data_test)
> y_t=logistic_data_test["target"]
> logistic<-glm(y1~lead_time+previous_cancellations,logistic_data_train,family = "binomial")
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(logistic)

Call:
glm(formula = y1 ~ lead_time + previous_cancellations, family = "binomial",
    data = logistic_data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.1443  -0.8576  -0.7459   1.2345   1.6938

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          -1.16240    0.03782  -30.74   <2e-16 ***
lead_time             3.55903    0.18133   19.63   <2e-16 ***
previous_cancellations 47.22159    3.75426   12.58   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 9224.3 on 6999 degrees of freedom

Residual deviance: 8430.8 on 6997 degrees of freedom

AIC: 8436.8

Number of Fisher Scoring iterations: 6

```
> res<-predict(logistic,logistic_data_test,type="response")
> View(res)
> confmatrix1<-table(Actual_value=logistic_data_test$target,Predicted_value=res>0.2)
> confmatrix2<-table(Actual_value=logistic_data_test$target,Predicted_value=res>0.3)
> confmatrix3<-table(Actual_value=logistic_data_test$target,Predicted_value=res>0.4)
> confmatrix1
            Predicted_value
Actual_value TRUE
           0 1872
           1 1128
> confmatrix2
            Predicted_value
Actual_value FALSE TRUE
           0  1069  803
           1   318  810
> confmatrix3
            Predicted_value
Actual_value FALSE TRUE
           0  1492  380
           1   625  503
```