Project 2: Classification

Code

Link to “Hotel booking demand”: https://www.kaggle.com/jessemostipak/hotel-booking-demand (https://www.kaggle.com/jessemostipak/hotel-booking-demand)

Initial Data Exploration and Cleaning

Hide

```
#load the data
hb <- read.csv("hotel_bookings.csv")
attach(hb)
```

Check out the Kaggle link for more information on the data set and its variables

Hide

```
#exploration function 1
str(hb)
```

```
'data.frame':	119390 obs. of  32 variables:
 $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hot
el" ...
 $ 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" ...
```

This data set contains booking information for a city hotel and a resort hotel. I am aiming to predict the possibility of booking or if is_canceled is 0 or 1 (1 if canceled). I decided to start off exploring the data with the str() function to get an idea of the structure of the data and see a list of all the columns. We can see that this is a data set containing 119390 observations and 32 attributes. We also see that most of the data types for the attributes are either int or chr. The ints make sense as their are mostly counts but some of the chrs may have to be changed to factors as they are options or categories. We can also see a few null values for agent and company which we will also further explore in the next section. Before that, however, let's drop some columns so we can focus on the most relevant attributes.

Hide

```
#data cleaning: dropping unnecessary numerical columns
hb <- subset(hb, select = -c(arrival_date_year, arrival_date_day_of_month, booking_changes, days
_in_waiting_list, agent, company))
#data cleaning: dropping unnecessary categorical columns
hb <- subset(hb, select = -c(country, assigned_room_type, reservation_status, reservation_status
_date))
```

I first looked into which numerical columns might not be the most necessary. Arrival date year and day of month are unecessary as we will be using arrival week. Booking data and days in waiting list could both change over time and may not be useful for modeling. Finally agent and company are both id numbers that don't have much pertinence to the cancellation factor.

Next I looked into which categorical attributes were necessary. Here country has many levels that may not generalize well in the model, something I learned with my regression work. Next assigned room type is quite similar to reserved room type so it is deemed redundant. Next reservation status and its date are first directly related to the cancel factor so we can get rid of multicolinearity early on here.

Hide

```
#exploration function 2
summary(hb)
```

```
   hotel            is_canceled       lead_time   arrival_date_month
 Length:119390      Min.   :0.0000   Min.   :  0   Length:119390     
 Class :character   1st Qu.:0.0000   1st Qu.: 18   Class :character  
 Mode  :character   Median :0.0000   Median : 69   Mode  :character  
                    Mean   :0.3704   Mean   :104                     
                    3rd Qu.:1.0000   3rd Qu.:160                     
                    Max.   :1.0000   Max.   :737                     
                                                                     
 arrival_date_week_number stays_in_weekend_nights
 Min.   : 1.00            Min.   : 0.0000        
 1st Qu.:16.00            1st Qu.: 0.0000        
 Median :28.00            Median : 1.0000        
 Mean   :27.17            Mean   : 0.9276        
 3rd Qu.:38.00            3rd Qu.: 2.0000        
 Max.   :53.00            Max.   :19.0000        
                                                 
 stays_in_week_nights     adults          children     
 Min.   : 0.0         Min.   : 0.000   Min.   : 0.0000  
 1st Qu.: 1.0         1st Qu.: 2.000   1st Qu.: 0.0000  
 Median : 2.0         Median : 2.000   Median : 0.0000  
 Mean   : 2.5         Mean   : 1.856   Mean   : 0.1039  
 3rd Qu.: 3.0         3rd Qu.: 2.000   3rd Qu.: 0.0000  
 Max.   :50.0         Max.   :55.000   Max.   :10.0000  
                                       NA's   :4        
     babies              meal           market_segment    
 Min.   : 0.000000   Length:119390      Length:119390     
 1st Qu.: 0.000000   Class :character   Class :character  
 Median : 0.000000   Mode  :character   Mode  :character  
 Mean   : 0.007949                                        
 3rd Qu.: 0.000000                                        
 Max.   :10.000000                                        
                                                          
 distribution_channel is_repeated_guest previous_cancellations
 Length:119390        Min.   :0.00000   Min.   : 0.00000      
 Class :character     1st Qu.:0.00000   1st Qu.: 0.00000      
 Mode  :character     Median :0.00000   Median : 0.00000      
                      Mean   :0.03191   Mean   : 0.08712      
                      3rd Qu.:0.00000   3rd Qu.: 0.00000      
                      Max.   :1.00000   Max.   :26.00000      
                                                              
 previous_bookings_not_canceled reserved_room_type deposit_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                                                     
                                                                     
 customer_type           adr          required_car_parking_spaces
 Length:119390      Min.   :  -6.38   Min.   :0.00000            
 Class :character   1st Qu.:  69.29   1st Qu.:0.00000            
 Mode  :character   Median :  94.58   Median :0.00000            
                    Mean   : 101.83   Mean   :0.06252            
```

```
                    3rd Qu.: 126.00   3rd Qu.:0.00000  
                    Max.   :5400.00   Max.   :8.00000  

 total_of_special_requests
 Min.   :0.0000           
 1st Qu.:0.0000           
 Median :0.0000           
 Mean   :0.5714           
 3rd Qu.:1.0000           
 Max.   :5.0000           
```

With the summary function we want to move our focus from the structure of the date to the columns themselves. Specifically, if they have the right data types and NA values. First, as mentioned earlier, we have to change a few character types and even ints to factor variables.

Hide

```
#cleaning: changing variable data types
hb$hotel <- as.factor(hb$hotel)
hb$is_canceled <- as.factor(hb$is_canceled)
hb$meal <- as.factor(hb$meal)
hb$market_segment <- as.factor(hb$market_segment)
hb$distribution_channel <- as.factor(hb$distribution_channel)
hb$is_repeated_guest  <- as.factor(hb$is_repeated_guest )
hb$reserved_room_type <- as.factor(hb$reserved_room_type)
hb$deposit_type <- as.factor(hb$deposit_type)
hb$customer_type <- as.factor(hb$customer_type)
hb$adr[hb$adr==5400] <- 540
```

The following variables were converted to factors: hotel, is_canceled, meal, market_segment, distribution_channel, is_repeated_guest, reserved_room_type, deposit_type, and customer_type. Many variables were changed but the rules were the same accross all cases; an attribute was only changed to a factor if it was out of a few categories and was discretely separate for each option. Other variables such as babies are also discrete but have the possibility of increasing past their certain limits; variables similar to this case were left unchanged. One more thing, the max of adr, Average Daily Rate, is 5400, which is not possible so it must be a input error. This is fixed to 540.

Hide

```
#cleaning: dealing with missing values and obs with no guests
hb$children[is.na(hb$children)] <- 0
hb <- hb[ which((hb$adults + hb$children + hb$babies)!=0), ] #double check
```

There are four columns that have null/NA values: children, country, agent, and company. Luckily, we actually dropped three of these leaving children. Theoretically, if we did have to deal with these NAs I would still get rid of the columns as there is no sound way to guess the specific values here. For the children variables, it is safe to assume that null values means there are no children. Outside of these NA issues, a similar issue that has to be addressed are rows that have 0 guests (adults+children+babies). These are either typos/errors or input that wasn't cleared properly. Either way these rows are removed in this part of the cleaning.

Hide

```
#exploration function 3
head(hb)
```

	hotel <fctr>	is_canceled <fctr>	lead_time <int>	arrival_date_month <chr>	arrival_date_week_number <int>
1	Resort Hotel	0	342	July	27
2	Resort Hotel	0	737	July	27
3	Resort Hotel	0	7	July	27
4	Resort Hotel	0	13	July	27
5	Resort Hotel	0	14	July	27
6	Resort Hotel	0	14	July	27

6 rows | 1-6 of 22 columns

Hide

```
tail(hb)
```

	hotel <fctr>	is_canceled <fctr>	lead_time <int>	arrival_date_month <chr>	arrival_date_week_number <int>
119385	City Hotel	0	21	August	35
119386	City Hotel	0	23	August	35
119387	City Hotel	0	102	August	35
119388	City Hotel	0	34	August	35
119389	City Hotel	0	109	August	35
119390	City Hotel	0	205	August	35

6 rows | 1-6 of 22 columns

Next, with the head/tail functions we can look at the beginning and end of the data in the format of rows. An instance in this context is the booking information of a customer. This is also a good time to see if there are any unreasonable data points at the ends of the data. Just a quick look tells us the beginning is mostly resort hotel bookings in July, week 27, whereas the tail is city hotel and week 35, in August. Another comparable variable is the market_segment which is Direct, Corporate, and Online TA for the beginning and offline TA/TO for the end. The data seems to be many years with week numbers for different years.

Hide

```
#exploration function 4
table(hotel, is_canceled)
```

```
              is_canceled
hotel              0     1
  City Hotel   46228 33102
  Resort Hotel 28938 11122
```

Hide

```
table(hotel, is_canceled)[3]/table(hotel, is_canceled)[1]
```

```
[1] 0.7160595
```

Hide

```
table(hotel, is_canceled)[4]/table(hotel, is_canceled)[2]
```

```
[1] 0.3843389
```

Hide

```
table(customer_type, is_canceled)
```

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

Hide

```
table(customer_type, is_canceled)[5]/table(customer_type, is_canceled)[1]
```

```
[1] 0.4484719
```

Hide

```
table(customer_type, is_canceled)[6]/table(customer_type, is_canceled)[2]
```

```
[1] 0.1138996
```

Hide

```
table(customer_type, is_canceled)[7]/table(customer_type, is_canceled)[3]
```

```
[1] 0.6876589
```

Hide

```
table(customer_type, is_canceled)[8]/table(customer_type, is_canceled)[4]
```

```
[1] 0.3410195
```

Out of all the factor variables, I was most interested in how hotel type and customer type were conditioned with cancellation. Using the table function I can observe exactly that. We see that there is a much higher number of cancellations for the city hotel than the resort hotel. For customer types, we see that transient customers have the highest cancellation rate followed by contract, transient party, and group at a very small 11.39%.

Hide

```
#exploration function 5
library(caret)
```

```
Loading required package: lattice
Loading required package: ggplot2
package 㤼㸱ggplot2㤼㸲 was built under R version 4.0.4RStudio Community is a great place to get help:
https://community.rstudio.com/c/tidyverse
```

Hide

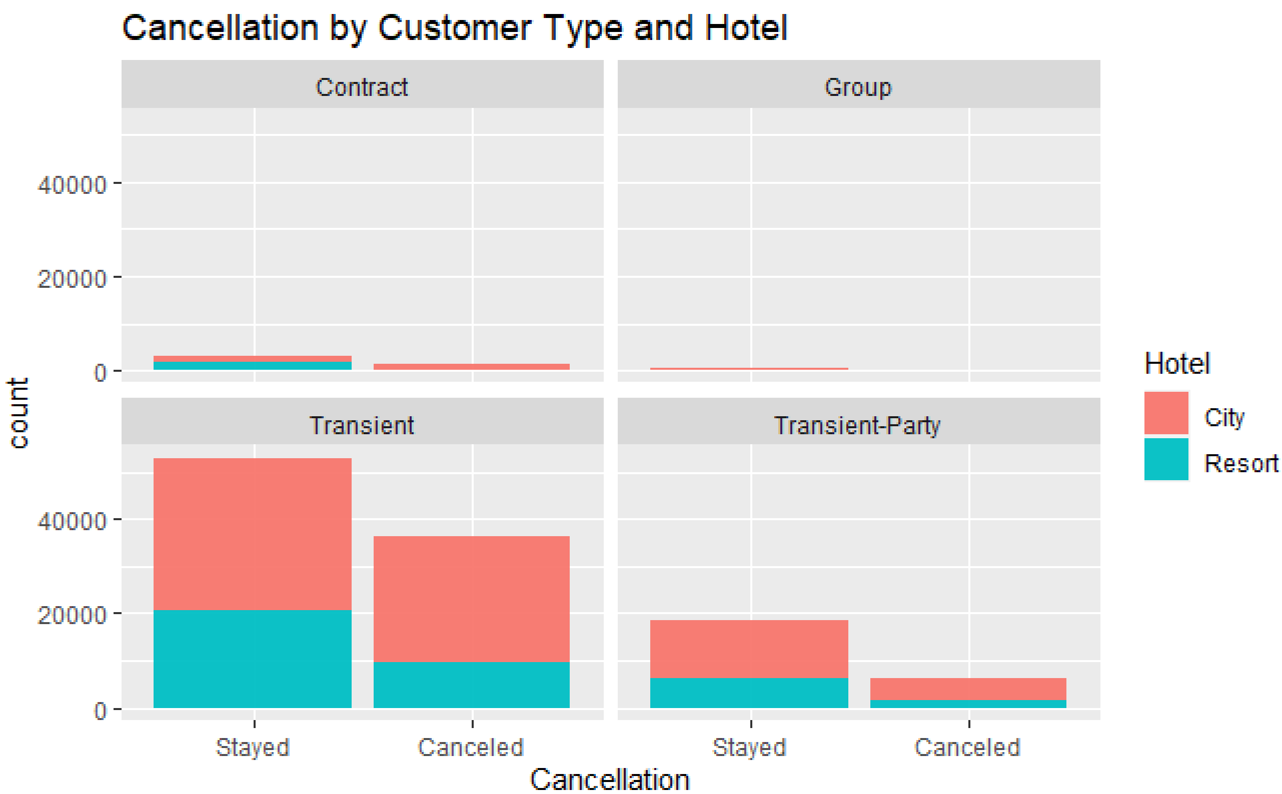
```
library(mlbench)
corMatrix1 <- cor(hb[sapply(hb,is.numeric)])
findCorrelation(corMatrix1, cutoff=0.5, verbose=TRUE)
```

```
All correlations <= 0.5
integer(0)
```

For the last part of data exploration, I decided to start feature selection early and looking into multicolinearity. I wasn’t able to simple look at just my target variable, is_canceled, as it was a factor and using other libraries just made it too difficult to discern between all the levels in the data set. Instead, performing feature selection with caret’s findCorrelation function allowed me to see which variables are highly correlated and must be dealt with. However, after some confusion I realized the output of integer(0) mean there are no correlations that meet the criteria of the cutoff 0.5; we are pretty safe from multicolinearity here. Next let’s explore the data further with a couple graphs.

Hide

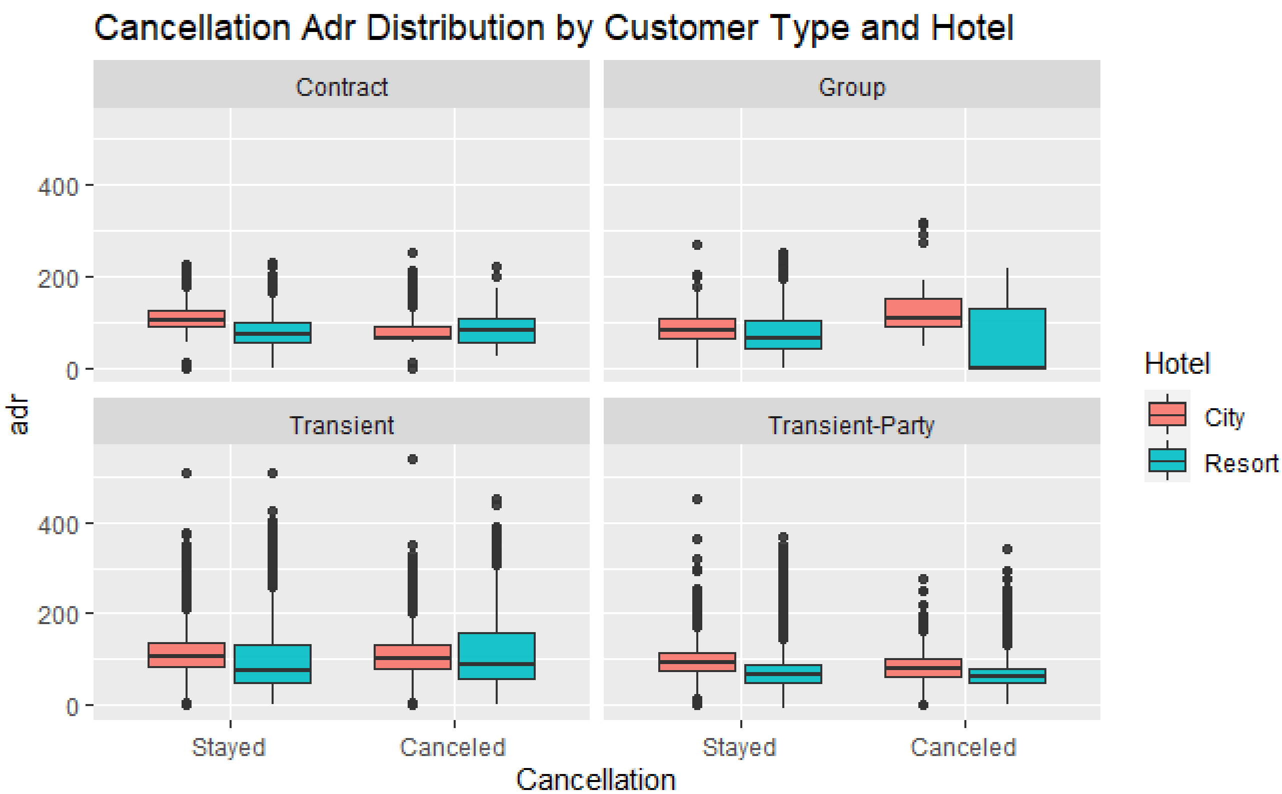
```
#exploration graph 1
library(ggplot2)
ggplot(data = hb, mapping = aes(x = is_canceled, fill = hotel)) +
  geom_bar() +
  facet_wrap(~ customer_type) +
  labs(c("0", "1"), title = "Cancellation by Customer Type and Hotel") + scale_x_discrete(name =
"Cancellation", labels=c("Stayed", "Canceled")) +
  scale_fill_discrete(name = "Hotel", labels = c("City", "Resort"))
```



For the first graph, I decided to explore the relationships between my main factors, is_canceled, hotel, customer type. Using fill and facet wrap, we can explore all factors at once. Some obvious observations are that the transient and transient party customers are the largest groups in that order. Contract is much smaller than both and group is very small, almost nonexistent. Coming back to our earlier observation comparing cancellation proportion among groups, we can see this much more clearly with transient have a large chunk of canceled bookings and then contract proportionally having almost half, transient party slightly less than half, and group's cancellation being extremely small. AS for the hotel breakdown, we can clearly see that city hotel observations make a much larger portion of the data with more than half of transient and transient party observation coming from city hotels and essentially the same for contract customers.

Hide

```
#exploration graph 2
ggplot(data = hb, mapping = aes(is_canceled, adr, fill = hotel)) +
  geom_boxplot() +
  facet_wrap(~ customer_type) +
  labs(c("0", "1"), title = "Cancellation Adr Distribution by Customer Type and Hotel") + scale_
x_discrete(name = "Cancellation", labels=c("Stayed", "Canceled")) +
  scale_fill_discrete(name = "Hotel", labels = c("City", "Resort"))
```



This dataset doesn't have too many numerical variables but I decided to explore just Adr, Average Daily Rate (the sum of transactions divided by the number of nights stayed), for my other graph. Instead of just exploring it's own distribution I decided to also compare it with hotel, customer type, and cancellation conditions. Starting off with comparing just the magnitude of the values, except for canceled contract bookings, it seems that median City Adr is greater than Resort Adr in all other cases. There is also a larger interquartile range for transient customers, which may be due to their large observation number. Between stayed or canceled bookings, the adr is essentially the same except for canceled resort groups, which is much lower than their staying resort counterparts. Coming to the hotel types, we see that resorts generally have higher variation with adr than city hotels. All in all, it seems that cities have higher adrs than resorts, staying customers more than canceled, and arguably transient above others who are pretty much tied.

Modeling

Hide

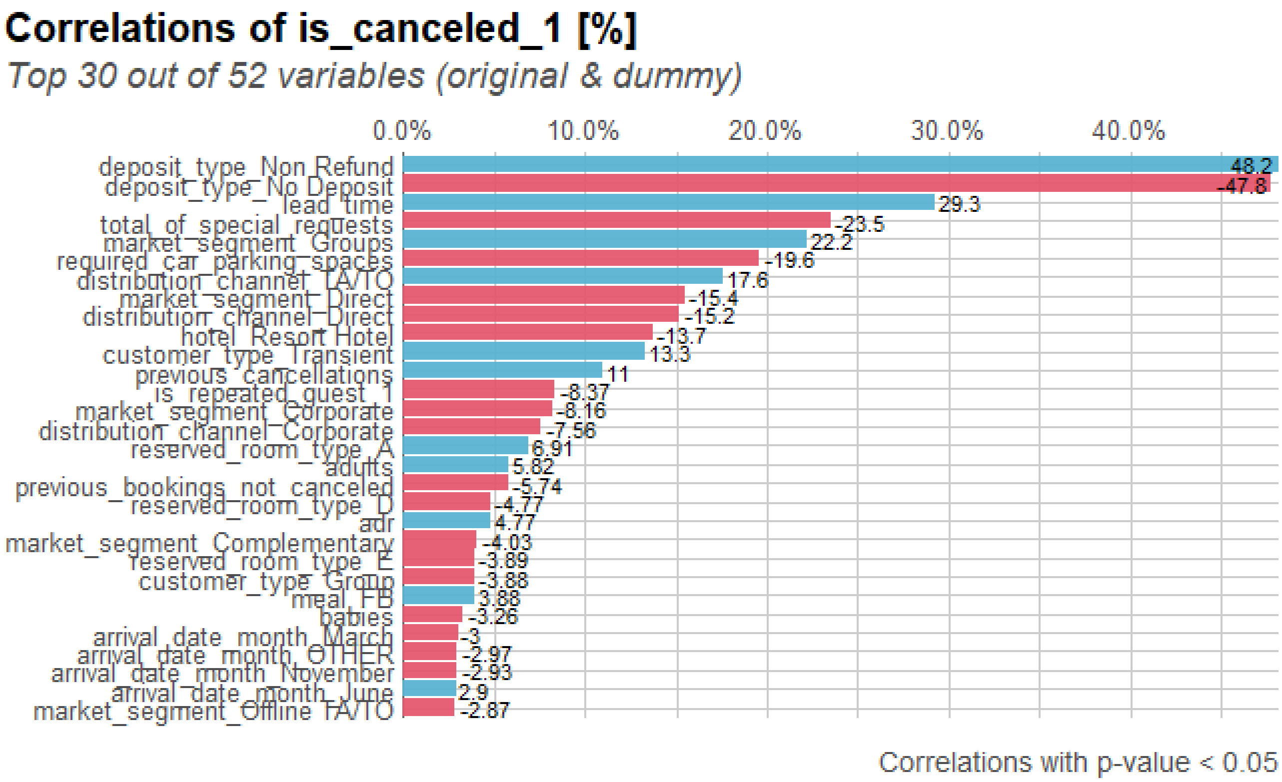
```
#feature selecting decisions
#attrEval from CORElearn
library(CORElearn)
sort(attrEval("is_canceled", hb, estimator="ReliefFexpRank",  ReliefIterations=30))
```

```
       distribution_channel          stays_in_week_nights 
              -0.0155329521                 -0.0093357140 
                  lead_time previous_bookings_not_canceled 
              -0.0083764040                 -0.0001608672 
                     adults         previous_cancellations 
               0.0000000000                  0.0008884956 
    stays_in_weekend_nights                       children 
               0.0013108706                  0.0059546144 
                       meal                         babies 
               0.0073686772                  0.0074243502 
         reserved_room_type             arrival_date_month 
               0.0092844970                  0.0108056533 
   arrival_date_week_number              is_repeated_guest 
               0.0166969251                  0.0172617198 
                      hotel    required_car_parking_spaces 
               0.0223487677                  0.0256982834 
              customer_type                            adr 
               0.0417250602                  0.0418498237 
  total_of_special_requests                   deposit_type 
               0.0565062853                  0.0810776449 
             market_segment 
               0.1147092190 
```

Hide

```
#corr_var from lares
library(lares)
corr_var(hb, is_canceled, max_pvalue = 0.05)
```

```
Maybe you meant one of: "is_canceled_1"Automatically using 'is_canceled_1
Not a valid input: is_canceled was transformed or does not exist.Automatically reduced results t
o the top 30 variables. Use the 'top' parameter to override this limit.
```



Before we move onto create the models, let's decide what features we should use. Many of the inductive learning feature selection methods in the handbook didn't scale well for this large data set and froze/hung up when running them. However, after doing some research, I learned about the CORElearn package that helps with feature selection for large datasets and the lares package that works efficiently as well. The attrEval function for CORElearn evaluates all the attributes in relation to a target. By sorting this output, we can see that deposit_type, customer_type, lead_time, market_segment, required_car_parking_spaces, is_repeated_guest, reserved_room_type, and 5 more variables are greatest and above 1. The corr_var function from the lares package is a little more familiar as it works with correlation. Here I plotted the top 30 most significant correlations with is_canceled. We see that deposit_type, lead_type, market_segment, required_car_parking_spaces, distribution_channel, hotel, customer_type, previous_cancellations, is_repeated guests, reserved_room_type, and adr perhaps being the greatest in the curve. Essentially all the features have something to add but the top 3 seem to be deposit type, total_of_special_requests, and market_segment. Though I was originally going to use just these 3, I decided to use all as everything may contribute in different ways. This feature exploration was still a good experience to work with CORElearn and lares as well as learn more about the top attributes according to each.

Hide

```
#divide train and test
set.seed(1234)
i <- sample(1:nrow(hb), nrow(hb)*0.75, replace=FALSE)
train <- hb[i,]
test <- hb[-i,]

#build models

#logistic regression model
glm1 <- glm(is_canceled~., data=train, family=binomial)
```

```
glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Hide

```
#naive bayes model
library(e1071)
nb1 <- naiveBayes(is_canceled~., data=train)

#decision tree model
library(tree)
tree2 <- tree(is_canceled~., data=train)
```

```
NAs introduced by coercion
```

Metrics/Evaluation

Hide

```
#logistic regression metrics
summary(glm1)
```

```
Call:
glm(formula = is_canceled ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.7398  -0.4329   0.2052   6.1628

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -2.525e+00  2.476e-01 -10.198  < 2e-16 ***
hotelResort Hotel               1.102e-01  2.331e-02   4.728 2.27e-06 ***
lead_time                       4.161e-03  1.129e-04  36.855  < 2e-16 ***
arrival_date_monthAugust        9.990e-02  1.216e-01   0.821 0.411526
arrival_date_monthDecember      7.122e-01  2.365e-01   3.011 0.002600 **
arrival_date_monthFebruary      5.776e-03  7.284e-02   0.079 0.936797
arrival_date_monthJanuary      -1.834e-01  1.007e-01  -1.822 0.068458 .
arrival_date_monthJuly         -4.406e-02  9.437e-02  -0.467 0.640609
arrival_date_monthJune         -1.169e-02  7.028e-02  -0.166 0.867885
arrival_date_monthMarch        -1.412e-01  5.181e-02  -2.725 0.006434 **
arrival_date_monthMay          -2.757e-02  4.887e-02  -0.564 0.572740
arrival_date_monthNovember      5.052e-01  2.080e-01   2.429 0.015139 *
arrival_date_monthOctober       3.330e-01  1.789e-01   1.862 0.062646 .
arrival_date_monthSeptember     7.159e-02  1.522e-01   0.470 0.638005
arrival_date_week_number       -1.543e-02  6.575e-03  -2.347 0.018916 *
stays_in_weekend_nights         5.777e-02  1.027e-02   5.624 1.87e-08 ***
stays_in_week_nights            4.245e-02  5.437e-03   7.808 5.83e-15 ***
adults                          1.225e-01  1.797e-02   6.814 9.48e-12 ***
children                        1.364e-01  2.826e-02   4.827 1.38e-06 ***
babies                          4.651e-02  9.692e-02   0.480 0.631320
mealFB                          5.908e-01  1.231e-01   4.801 1.58e-06 ***
mealHB                         -1.780e-01  3.123e-02  -5.698 1.21e-08 ***
mealSC                          1.650e-01  2.995e-02   5.508 3.64e-08 ***
mealUndefined                  -7.024e-01  1.138e-01  -6.170 6.84e-10 ***
market_segmentComplementary     5.202e-01  2.764e-01   1.882 0.059801 .
market_segmentCorporate        -7.413e-02  2.180e-01  -0.340 0.733886
market_segmentDirect           -6.960e-03  2.383e-01  -0.029 0.976699
market_segmentGroups           -1.354e-01  2.267e-01  -0.598 0.550155
market_segmentOffline TA/TO    -7.330e-01  2.274e-01  -3.224 0.001265 **
market_segmentOnline TA         6.072e-01  2.266e-01   2.679 0.007374 **
market_segmentUndefined         2.580e+00  6.879e+03   0.000 0.999701
distribution_channelDirect     -3.864e-01  1.081e-01  -3.574 0.000352 ***
distribution_channelGDS        -8.211e-01  2.336e-01  -3.516 0.000439 ***
distribution_channelTA/TO       1.992e-01  8.118e-02   2.454 0.014111 *
distribution_channelUndefined   1.953e+01  2.185e+03   0.009 0.992869
is_repeated_guest1             -6.593e-01  9.916e-02  -6.649 2.95e-11 ***
previous_cancellations          2.918e+00  7.029e-02  41.513  < 2e-16 ***
previous_bookings_not_canceled -5.695e-01  3.251e-02 -17.518  < 2e-16 ***
reserved_room_typeB             1.378e-02  8.819e-02   0.156 0.875844
reserved_room_typeC             1.382e-02  1.050e-01   0.132 0.895272
reserved_room_typeD            -5.673e-02  2.546e-02  -2.229 0.025844 *
reserved_room_typeE            -1.428e-02  4.136e-02  -0.345 0.729844
reserved_room_typeF            -4.628e-01  6.642e-02  -6.968 3.23e-12 ***
```

```
reserved_room_typeG           -2.437e-01  7.824e-02  -3.114 0.001843 **
reserved_room_typeH           -1.621e-01  1.292e-01  -1.255 0.209510
reserved_room_typeL            4.629e-01  1.231e+00   0.376 0.706887
deposit_typeNon Refund         5.608e+00  1.306e-01  42.944  < 2e-16 ***
deposit_typeRefundable         1.925e-02  2.408e-01   0.080 0.936282
customer_typeGroup            -2.403e-01  2.026e-01  -1.186 0.235736
customer_typeTransient         7.809e-01  6.223e-02  12.549  < 2e-16 ***
customer_typeTransient-Party   2.994e-01  6.592e-02   4.542 5.57e-06 ***
adr                            5.768e-03  2.855e-04  20.200  < 2e-16 ***
required_car_parking_spaces   -3.401e+01  6.123e+01  -0.555 0.578574
total_of_special_requests     -7.500e-01  1.334e-02 -56.214  < 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: 117904  on 89406  degrees of freedom
Residual deviance:  77086  on 89353  degrees of freedom
AIC: 77194

Number of Fisher Scoring iterations: 17
```

Hide

```
probs <- predict(glm1, newdata=test, type="response")
glmpred <- ifelse(probs>0.5, 1, 0)
table(glmpred, test$is_canceled)
```

```


glmpred     0     1
      0 17460  4451
      1  1294  6598
```

Hide

```
glmacc <- mean(glmpred==test$is_canceled)
print(paste("acc: ", glmacc))
```

```
[1] "acc:  0.807234171056605"
```

Looking at the logistic regression summary we can see that many of the variables/their levels were significant in the model. We see a large drop from null deviance to residual deviance, 117904 to 77086, which means that our predictors were good predictors compared to using just the intercept. Looking at the models table of predictions and actual values, we see that there are much more TPs and TNs than FNs and FPs and this translates to the accuracy of 80.72%, which is quite good.

Hide

```
#naive bayes metrics
nbpred <- predict(nb1, newdata=test, type="class")
confusionMatrix(nbpred, test$is_canceled)
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0  2737   166
         1 16017 10883

               Accuracy : 0.457
                 95% CI : (0.4513, 0.4627)
    No Information Rate : 0.6293
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1011

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.14594
            Specificity : 0.98498
         Pos Pred Value : 0.94282
         Neg Pred Value : 0.40457
             Prevalence : 0.62927
         Detection Rate : 0.09184
   Detection Prevalence : 0.09741
      Balanced Accuracy : 0.56546

       'Positive' Class : 0
```

Hide

```
table(nbpred, test$is_canceled)
```

```
nbpred     0     1
     0  2737   166
     1 16017 10883
```

Hide

```
library(caret)
nbacc <- mean(nbpred==test$is_canceled)
print(paste("acc: ", nbacc))
```

```
[1] "acc:  0.457000973056404"
```

The naive bayes algorithm, however, didn't perform so well. With much more FNs than TPs, the sensitivity was much worse. Interestingly, there were much more TNs than FPs leading to a very high specificity. This may be due the "naiveness" of naive bayes which return false regardless of the given sample most of the time. This low sensitivity is seen in the accuracy, which is 45.7%.

Hide

```
#decision tree metrics
summary(tree2)
```

```
Classification tree:
tree(formula = is_canceled ~ ., data = train)
Variables actually used in tree construction:
[1] "deposit_type"              "lead_time"
[3] "market_segment"            "previous_cancellations"
[5] "total_of_special_requests"
Number of terminal nodes:  6
Residual mean deviance:  0.9029 = 80720 / 89400
Misclassification error rate: 0.2013 = 17998 / 89407
```

Hide

```
tree_pred2 <- predict(tree2, newdata=test, type="class")
```

```
NAs introduced by coercion
```

Hide

```
table(tree_pred2, test$is_canceled)
```

```
tree_pred2     0     1
         0 17052  4359
         1  1702  6690
```

Hide

```
treeacc <- mean(tree_pred2 == test$is_canceled)
print(paste("acc: ", treeacc))
```

```
[1] "acc:  0.796631211622991"
```

Finally, the decision tree pick it back up with table values much like logistic regression. Interestingly, the model only used deposit_type, lead_time, market_segment, previous_cancellations, and total_of_special_requests. It had a low misclassification error rate of .2013 and the accuracy is very close to logistic regression at 79.66%.

Ranking the algorithms from best to worst accuracy, we have logistic regression, decision tree, and naive bayes. Naive bayes ran the slowest out of the three because it is more simplistic probability learning and is generally meant for small data sets. Moreover, some of the predictors may not have been independent so the naive assumption that they are may have limited the performance of the algorithm. This is most likely the reason it was outperformed by logistic regression and the decision tree. Logistic regression searches for a single linear decision boundary whereas the decision tree partitions the feature space into half spaces for a boundary but in this case the effect was more or less the same. However, because decision trees are so flexible, the model may have been prone to overfitting and logistic regression was less susceptible here. Maybe if any pruning was done, the accuracy could have increased. All in all, this was a battle of bias-variance tradeoff and logistic won, very slightly, and naive bayes struggled against the size of the data set.

Logistic regression may have won as it assumed the relationship between the predictors and cancellation to be linear and was very close in this case. All in all, this classification study on hotel booking cancellation was introspective in that it highlighted the importance of various predictors if not all. In the future, when predicting the booking status of a customer, perhaps all the data of a customer should be holistically considered. However, we also learned that deposit_type, lead_time, and market_segment were some of the top predictors and previous_cancellations and total_of_special_requests were runner ups. Lead time and deposit type really show the customer's interest when actually confirming the booking and may be the most directly associated attributes. These type of variables show what kind of customer, either very inclined or normal, is booking and coupled with special requests we can see if the customer is really planning ahead to stay. Previous cancellation can also signify similar connotations either showing that they often hold rooms as an option or actually follow through on previous bookings. Finally, market segment is somewhat different showing more where the customer may be coming from but is nonetheless similar; if I was really interested in a trip I may book with a travel agent than just directly. This project was also a good introduction to feature engineering packages such as CORElearn and lares. In the future, classification models such as this can help hotels prioritize their services and even retain more customers by targeting those more prone to cancel.