Analytic Plan

Daniel Jackson

2024-05-19

Project Overview

Customer: ABC Hotels.

Business Need: ABC Hotels would like to identify bookings that have a high risk of cancellation. The risk of cancellation should be a value between 0 and 1, so it can be interpreted as the probability of cancellation. With this capability, hotel management can target bookings that have a high risk (i.e., probability) of cancellation with additional advertisements and/or offers in an effort to prevent them from being cancelled.

Data: ABC Hotels has provided a data set containing over 35,000 bookings for which it is known whether or not the booking was cancelled. Students are required to use this data set provided in the zipped folder below.

Questions to Answer

The first step in the machine learning process is to carefully consider the objectives (i.e., the business needs) in the context of the available data, appropriate and applicable machine learning methods, as well as the expected analytic and informational outcomes. We will be addressing the questions below:

1.) What is the label (i.e., the target or dependent variable) for the supervised classification problem?

2.) What data processing is needed and how will it be performed?

3.) What features will be initially included?

4.) What are the expected analytic and informational outcomes to be produced?

5.) How will the model be used in practice?

Below, we included the R code of our exploratory data analysis. Within this analysis, we will be addressing and answering the questions above.

Analytic Plan Approach and R Code

```
# Read in data
library(dplyr)
```

```r
getwd()
```

```
## [1] "/Users/doojerthekid/Documents/Merrimack Grad School
Documents/DSE6211/Final_Project"
```

```r
# Read in CSV
hotel_df = read.csv("project_data.csv")

# Check for NA values
colSums(is.na(hotel_df))
```

```
##                            Booking_ID                          no_of_adults
##                                     0                                     0
##                        no_of_children                 no_of_weekend_nights
##                                     0                                     0
##                     no_of_week_nights                    type_of_meal_plan
##                                     0                                     0
##           required_car_parking_space                   room_type_reserved
##                                     0                                     0
##                              lead_time                          arrival_date
##                                     0                                     0
##                   market_segment_type                         repeated_guest
##                                     0                                     0
##         no_of_previous_cancellations no_of_previous_bookings_not_canceled
##                                     0                                     0
##                    avg_price_per_room                no_of_special_requests
##                                     0                                     0
##                        booking_status
##                                     0
```

```r
# No NA values in data frame. This is great!

# Check unique values of booking status. This will be our response variable
for our analysis.
unique(hotel_df$booking_status)
```

```
## [1] "not_canceled" "canceled"
```

Our response variable will be whether the customer canceled, or did not cancel their reservation. We want our response variable to be a binary variable. Let us have 0 represent not canceled and 1 represent canceled and let us convert booking_status to binary response variable. We also will remove the Booking_ID column as we will not be interested in what their booking identification number will be for our analysis.

```r
hotel_df = hotel_df %>%
  mutate(booking_status = ifelse(booking_status == "canceled", 1, 0))

# We will not need the booking ID code for each observation. Let us remove
that variable
hotel_df = hotel_df[, -which(names(hotel_df) == "Booking_ID")]
```

Now, we are left with16 total variables. With 1 response variable, we have 15 predictors to use in our modeling. Let us look at a summary of our variables.

Variable Summary

booking_status: This is our response variable. It is a binary variable with values 0 and 1. This is a classification variable, meaning we will be constructing qualitative predictive models in our analysis.

no_of_adults: This represents the number of adults in the reservation. This is a discrete quantitative variable. Let us change the name of the variable from no_of_adults to just adults to help with syntax during modeling. We will have a snippet of code below for all of the variables that get a column name change.

no_of_children: This represents the number of children in the reservation. This is a discrete quantitative variable. We will change name to children for easier syntax.

no_of_weekend_nights: This represents the number of weekend nights in the reservation. This is a discrete quantitative variable. We will change name to weekend_nights for easier syntax.

no_of_week_nights: This represents the number of week nights in the reservation. This is a discrete quantitative variable. We will change name to week_nights for easier syntax.

type_of_meal_plan: This variable represents type of meal plan for each reservation. Let us look at the values.

```
unique(hotel_df$type_of_meal_plan)

## [1] "meal_plan_1"  "not_selected" "meal_plan_2"  "meal_plan_3"

# There is four options: not selected, 1, 2, or 3
# Check counts of each value
hotel_df %>%
  count(type_of_meal_plan)

##   type_of_meal_plan     n
## 1       meal_plan_1 27802
## 2       meal_plan_2  3302
## 3       meal_plan_3     5
## 4      not_selected  5129

#  type_of_meal_plan     n
# 1       meal_plan_1 27802
# 2       meal_plan_2  3302
# 3       meal_plan_3     5
# 4       not_selected  5129
# Let us change values to none, one, two or three
hotel_df = hotel_df %>%
  mutate(type_of_meal_plan = ifelse(type_of_meal_plan ==
                                      "not_selected", "none",
```

```
type_of_meal_plan)) %>%
  mutate(type_of_meal_plan = ifelse(type_of_meal_plan ==
                                      "meal_plan_1", "one",
type_of_meal_plan)) %>%
  mutate(type_of_meal_plan = ifelse(type_of_meal_plan ==
                                      "meal_plan_2", "two",
type_of_meal_plan)) %>%
  mutate(type_of_meal_plan = ifelse(type_of_meal_plan ==
                                      "meal_plan_3", "three",
type_of_meal_plan))
# Check count again to make sure code worked
hotel_df %>%
  count(type_of_meal_plan)

##   type_of_meal_plan     n
## 1              none  5129
## 2               one 27802
## 3             three     5
## 4               two  3302

# The code worked.
# This variable is a qualitative variable with four unique values.
# Let us change variable name to meal_plan
```

required_car_parking_space: This represents number of parking spaces needed. This is a discrete quantitative variable. Let us change the variable name to parking_space.

room_type_reserved: This represents type of room reserved. Let us check values

```
unique(hotel_df$room_type_reserved)

## [1] "room_type1" "room_type4" "room_type2" "room_type6" "room_type5"
## [6] "room_type7" "room_type3"

# Check counts of each
hotel_df %>%
  count(room_type_reserved)

##   room_type_reserved     n
## 1         room_type1 28105
## 2         room_type2   692
## 3         room_type3     7
## 4         room_type4  6049
## 5         room_type5   263
## 6         room_type6   964
## 7         room_type7   158

# There are 7 unique values. Let us change the names of unique values
hotel_df = hotel_df %>%
  mutate(room_type_reserved = ifelse(room_type_reserved ==
                                      "room_type1", "one",
```

```
room_type_reserved)) %>%
  mutate(room_type_reserved = ifelse(room_type_reserved ==
                                       "room_type2", "two",
room_type_reserved)) %>%
  mutate(room_type_reserved = ifelse(room_type_reserved ==
                                       "room_type3", "three",
room_type_reserved)) %>%
  mutate(room_type_reserved = ifelse(room_type_reserved ==
                                       "room_type4", "four",
room_type_reserved)) %>%
  mutate(room_type_reserved = ifelse(room_type_reserved ==
                                       "room_type5", "five",
room_type_reserved)) %>%
  mutate(room_type_reserved = ifelse(room_type_reserved ==
                                       "room_type6", "six",
room_type_reserved)) %>%
  mutate(room_type_reserved = ifelse(room_type_reserved ==
                                       "room_type7", "seven",
room_type_reserved))
hotel_df %>%
  count(room_type_reserved)

##   room_type_reserved     n
## 1               five   263
## 2               four  6049
## 3                one 28105
## 4              seven   158
## 5                six   964
## 6              three     7
## 7                two   692

# Code worked.
# This is a qualitative variable with seven different unique values.
# We will change the variable name to room_type
```

lead_time: We will be removing the lead_time predictor. We will treat this as we did with the booking ID variable. This is more of a time stamp observation on the reservation. Therefore we will remove it.

```
hotel_df = hotel_df[, -which(names(hotel_df) == "lead_time")]
```

arrival_date: This represents arrival date of each customer. We will also be removing this predictor as it will not have any impact on predicting whether a customer will cancel their reservation or not.

```
hotel_df = hotel_df[, -which(names(hotel_df) == "arrival_date")]
```

market_segment_type: This is a qualitative variable with five unique values that represents how the engaged customer booked.

```
unique(hotel_df$market_segment_type)
```

```
## [1] "offline"       "online"        "corporate"     "aviation"
## [5] "complementary"
```

The customer either booked offline (phone call), online, corporate, aviation or complementary. Let us change the variable to market_type

repeated_guest: This is a binary qualitative variable with 0 representing not a repeat guest and 1 representing repeat guest.

```
unique(hotel_df$repeated_guest)
```

```
## [1] 0 1
```

Let us change variable name to repeat_guest.

no_of_previous_cancellations: This represents the number of previous cancellations by customer.

```
unique(hotel_df$no_of_previous_cancellations)
```

```
## [1]  0  3  1  2 11  4  5 13  6
```

This is a continuous quantitative variable. Let us change name to previous_cancellations

no_of_previous_bookings_not_canceled: This represents number of previous bookings that were not canceled.

```
unique(hotel_df$no_of_previous_bookings_not_canceled)
```

```
##  [1]  0  5  1  3  4 12 19  2 15 17  7 20 16 50 13  6 14 34 18  8 10 23 11
49 47
## [26] 53  9 33 22 24 52 21 48 28 39 25 31 38 26 51 42 37 35 56 44 27 32 55
45 30
## [51] 57 46 54 43 58 41 29 40 36
```

This is a continuous quantitative variable. Let us change name to prev_not_cancel.

avg_price_per_room: This represents average booking price per room. This is a continuous quantitative variable.We will not change this variable for now.

no_of_special_requests: This represents number of special requests made by each customer. This is a continuous quantitative variable. We will change this variable to spec_requests.

Let us make the variable name changes:

```
hotel_df = hotel_df %>%
  rename(adults = no_of_adults,
         children = no_of_children,
         weekend_nights = no_of_weekend_nights,
         week_nights = no_of_week_nights,
         meal_plan = type_of_meal_plan,
         parking_spaces = required_car_parking_space,
         room_type = room_type_reserved,
```

```
        market_type = market_segment_type,
        repeat_guest = repeated_guest,
        prev_cancel = no_of_previous_cancellations,
        prev_not_cancel = no_of_previous_bookings_not_canceled,
        spec_requests = no_of_special_requests)
```

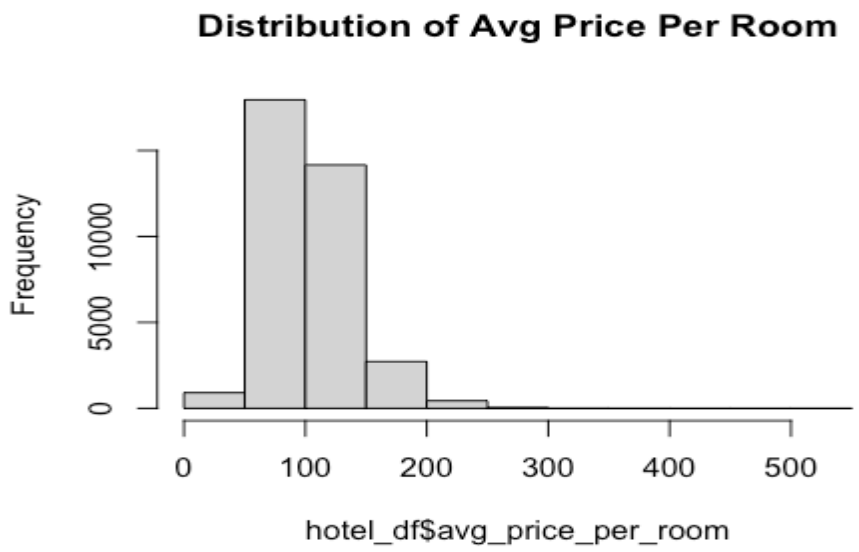
We have now summarized and cleaned up our data. We have one response variable and 13 predictors.

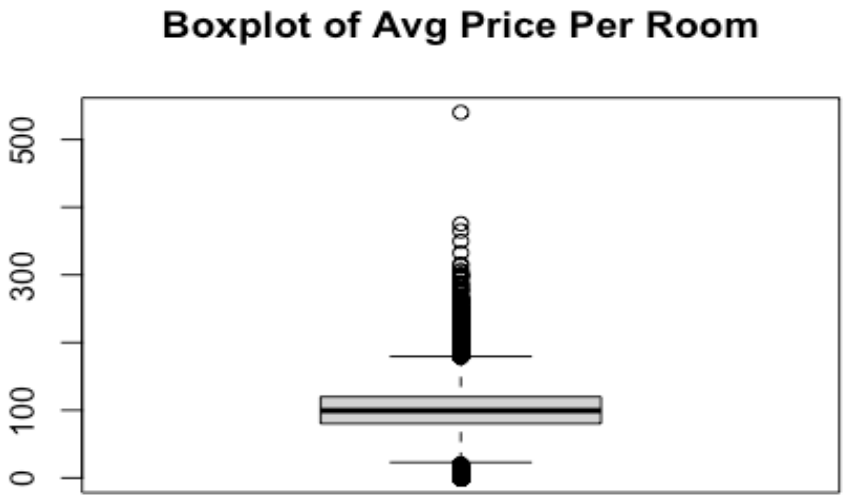
Variable Selection and Feature Engineering

We have already removed the customer ID variable, lead time variable and the arrival date variable.

Let us look at the average price per room variable. Looking at all of the other continuous variables, this specific continuous variable has the most unique values. Let us look at the distribution of it and see if we need to make any transformations to make distribution more normal:

```
hist(hotel_df$avg_price_per_room, main = "Distribution of Avg Price Per
Room")
boxplot(hotel_df$avg_price_per_room, main = "Boxplot of Avg Price Per Room")
```

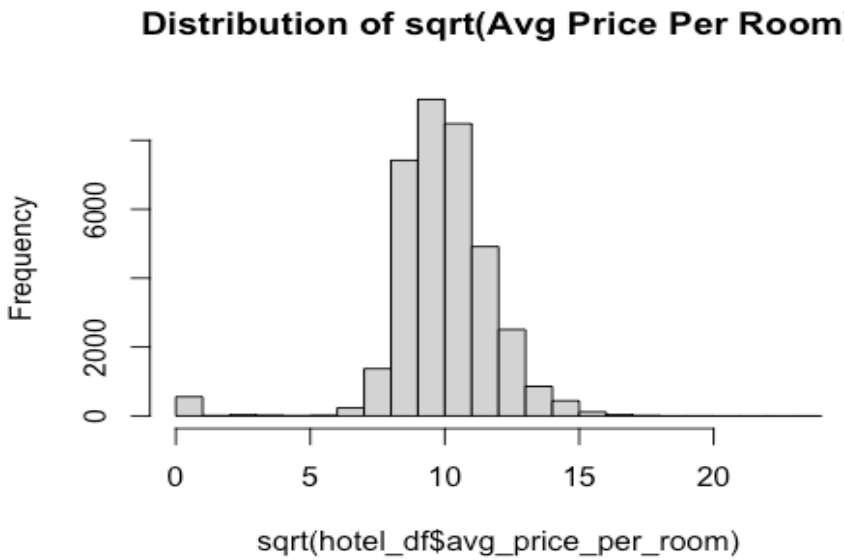


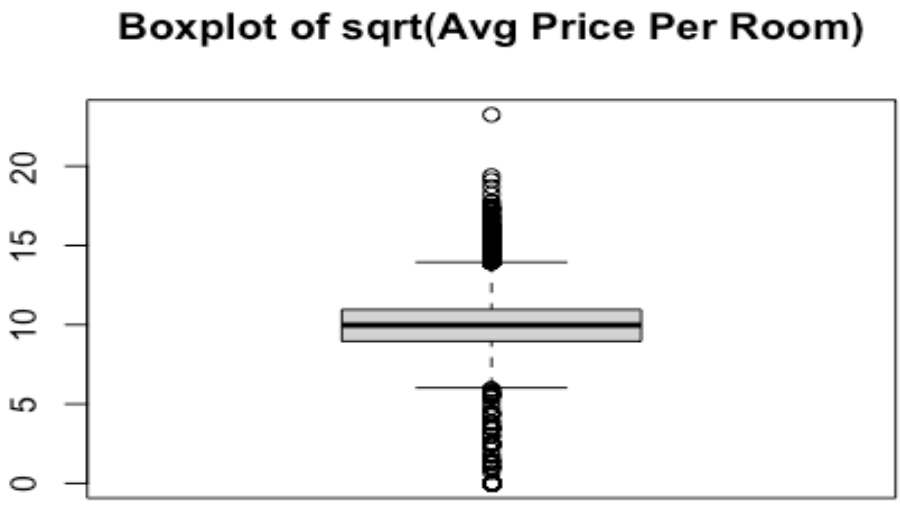


We see that the boxplot is heavily skewed right with a lot of outliers. Let us try some transformations to make the distribution more normal and minimize the number of outliers.

Square root:

```
hist(sqrt(hotel_df$avg_price_per_room), main = "Distribution of sqrt(Avg Price Per Room)")
boxplot(sqrt(hotel_df$avg_price_per_room), main = "Boxplot of sqrt(Avg Price Per Room)")
```



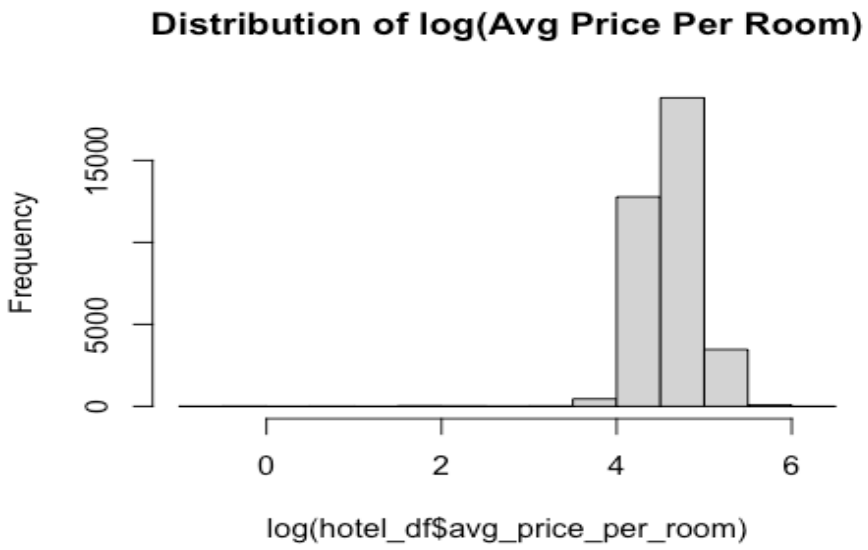


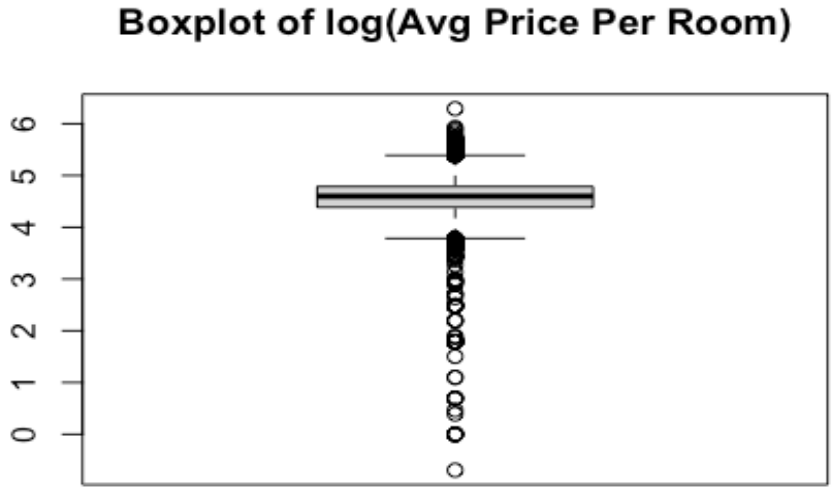
This made the distribution slightly more normal. Still a lot of outliers.

Log:

```
par(mfrow = c(1, 2))
hist(log(hotel_df$avg_price_per_room), main = "Distribution of log(Avg Price
Per Room)")
boxplot(log(hotel_df$avg_price_per_room), main = "Boxplot of log(Avg Price
Per Room)")

## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out =
## z$out[z$group == : Outlier (-Inf) in boxplot 1 is not drawn
```

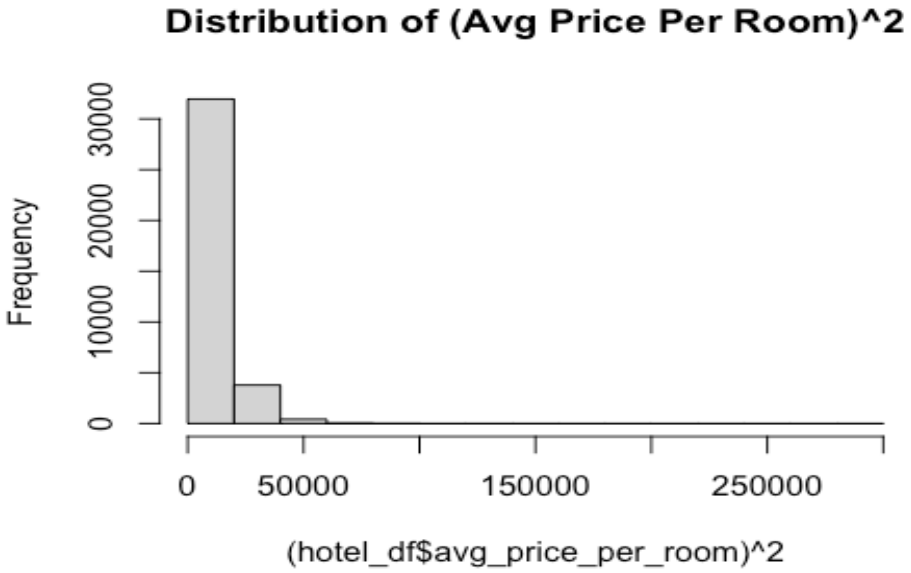


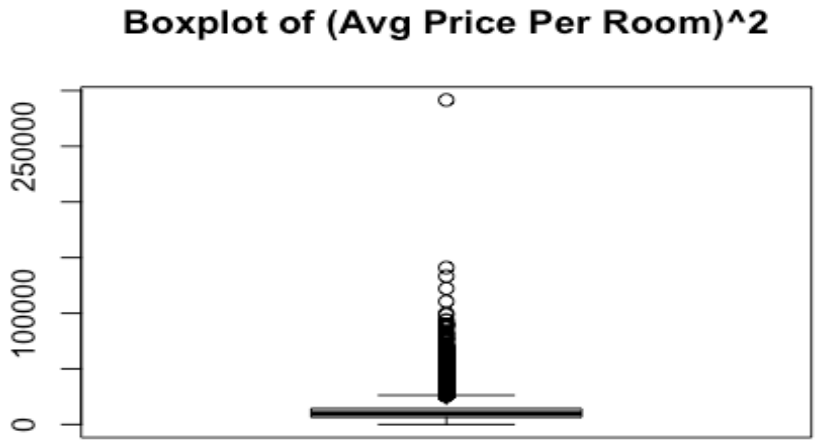


This did not make the distribution more normal.

Squared:

```
par(mfrow = c(1, 2))
hist((hotel_df$avg_price_per_room)^2, main = "Distribution of (Avg Price Per
Room)^2")
boxplot((hotel_df$avg_price_per_room)^2, main = "Boxplot of (Avg Price Per
Room)^2")
```


Distribution of (Avg Price Per Room)^2
Frequency
0
10000
20000
30000
0
50000
150000
250000
(hotel_df$avg_price_per_room)^2


Boxplot of (Avg Price Per Room)^2
0
100000
250000

This did not make the distribution more normal. The transformations did not help minimize the outliers or make the distribution more normal. Therefore, we will not be transforming this variable for our analysis.

Analysis Gameplan

This will be a qualitative regression analysis. Our response variable is a binary qualitative variable with two values: 0 for not-cancelled and 1 for cancelled reservation. We have 14 predictor variables and the 1 response variable for our analysis.

What is our goal?

Our goal is to fit a predictive model to help ABC Hotels identify bookings that have a high risk of cancellation using the data set given. The risk of cancellation will be a value between 0 and 1. The closer the probability is to 1, the higher risk of cancellation. Since we are trying to predict a qualitative response variable, we will fit at least one dense neural network model. In this portion of the project, we will specify the following aspects of the neural network(s) and discuss why/how they were chosen: number of layers, number of units for each layer, activation functions for each layer, loss function and optimization algorithm.

We will create a training and test data set (see next section). We will train the model using the training data set and will use that trained model to predict the response variable in the test set. We will then be using confidence matrices to measure our prediction rate.

We will evaluate the neural networks using learning curves on training and validation sets. We will check to see if the models are underfitting or overfitting the data. Based on this, changes will be made to the architecture of the dense neural networks. Based on the evaluation of the preliminary models, we will provide further data processing and feature engineering steps that will be implemented and investigated for the Final Report.

Training and Test Data

Let us look at the dimension of our data set:

```
dim(hotel_df)

## [1] 36238    14
```

There are 36238 observations and 14 variables. Let us define our training and test data. Since we have a good amount of observations, will be using 70% of our data to be the training data and 30% to be our test data.

```
set.seed(1)
train = sample(nrow(hotel_df), 0.7 * nrow(hotel_df))
train_df = hotel_df[train, ]
test_df = hotel_df[-train, ]
```

We have 25366 observations in our training data and 10872 observations in our test data. These data sets will be used to train our neural networks. We will use the trained neural network models to predict our booking_status response variables in the the test data.

Analytic Plan Conclusion

Now that we have done some data exploration and identified the business needs for ABC Hotel, we can now begin creating our neural network models to best predict the probability of a customer cancelling their reservation. Ideally, our findings will help ABC Hotel establish business strategies to minimize cancellations.