Project\_1.3

Ruturaj Solanki, Kunal Vaghela, Satya Akhil Govvala

2023-04-23

## R and RStudio Versions

* R version 4.2.2 (2022-10-31 ucrt)
* RStudio 2022.12.0+353 “Elsbeth Geranium” Release (7d165dcfc1b6d300eb247738db2c7076234f6ef0, 2022-12-03) for Windows, Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) RStudio/2022.12.0+353 Chrome/102.0.5005.167 Electron/19.1.3 Safari/537.36

## R Package Used

* ggplot2 (3.4.1)
* ggthemes (4.2.4)
* caret (6.0.94)
* caTools (1.18.2)
* car (3.1.2)
* DMwR (0.4.1)
* glmnet (4.1.7)
* ResourceSelection (0.3.5)
* naniar (1.0.o)

library("ggplot2")  
library("ggthemes")  
library("naniar")  
library("caret")

## Loading required package: lattice

library("caTools")  
library("car")

## Loading required package: carData

library("DMwR")

## Loading required package: grid

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library("nnet")  
library("glmnet")

## Loading required package: Matrix

## Loaded glmnet 4.1-7

library("ResourceSelection")

## ResourceSelection 0.3-5 2019-07-22

## Data defination and visiualisation

df <- read.csv("HRD.csv", header=TRUE, stringsAsFactors=FALSE)  
head(df)

## Booking\_ID no\_of\_adults no\_of\_children no\_of\_weekend\_nights no\_of\_week\_nights  
## 1 INN00002 2 0 2 3  
## 2 INN00003 1 0 2 1  
## 3 INN00005 2 0 1 1  
## 4 INN00007 2 0 1 3  
## 5 INN00008 2 0 1 3  
## 6 INN00009 3 0 0 4  
## type\_of\_meal\_plan required\_car\_parking\_space room\_type\_reserved lead\_time  
## 1 Not Selected 0 Room\_Type 1 5  
## 2 Meal Plan 1 0 Room\_Type 1 1  
## 3 Not Selected 0 Room\_Type 1 48  
## 4 Meal Plan 1 0 Room\_Type 1 34  
## 5 Meal Plan 1 0 Room\_Type 4 83  
## 6 Meal Plan 1 0 Room\_Type 1 121  
## arrival\_year arrival\_month arrival\_date market\_segment\_type repeated\_guest  
## 1 2018 11 6 Online 0  
## 2 2018 2 28 Online 0  
## 3 2018 4 11 Online 0  
## 4 2017 10 15 Online 0  
## 5 2018 12 26 Online 0  
## 6 2018 7 6 Offline 0  
## no\_of\_previous\_cancellations no\_of\_previous\_bookings\_not\_canceled  
## 1 0 0  
## 2 0 0  
## 3 0 0  
## 4 0 0  
## 5 0 0  
## 6 0 0  
## avg\_price\_per\_room no\_of\_special\_requests booking\_status sqrt\_lead\_time  
## 1 106.68 1 Not\_Canceled 2.236068  
## 2 60.00 0 Canceled 1.000000  
## 3 94.50 0 Canceled 6.928203  
## 4 107.55 1 Not\_Canceled 5.830952  
## 5 105.61 1 Not\_Canceled 9.110434  
## 6 96.90 1 Not\_Canceled 11.000000

n\_miss(df)

## [1] 0

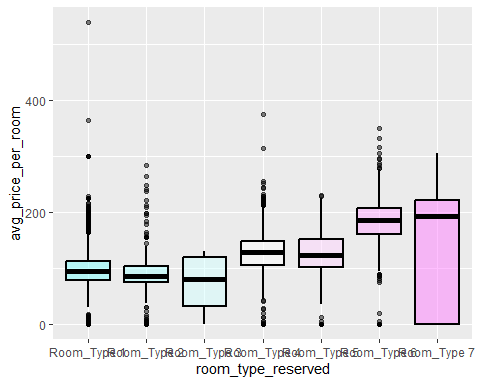
any\_na(df)

## [1] FALSE

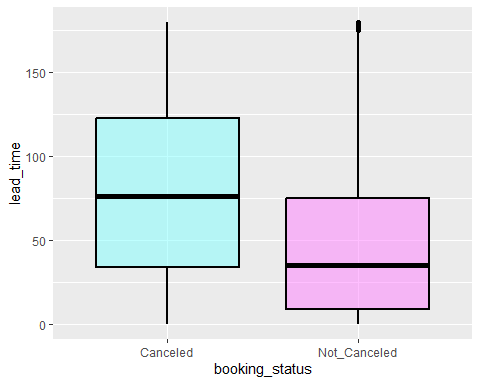
miss\_var\_summary(df)

## # A tibble: 20 × 3  
## variable n\_miss pct\_miss  
## <chr> <int> <dbl>  
## 1 Booking\_ID 0 0  
## 2 no\_of\_adults 0 0  
## 3 no\_of\_children 0 0  
## 4 no\_of\_weekend\_nights 0 0  
## 5 no\_of\_week\_nights 0 0  
## 6 type\_of\_meal\_plan 0 0  
## 7 required\_car\_parking\_space 0 0  
## 8 room\_type\_reserved 0 0  
## 9 lead\_time 0 0  
## 10 arrival\_year 0 0  
## 11 arrival\_month 0 0  
## 12 arrival\_date 0 0  
## 13 market\_segment\_type 0 0  
## 14 repeated\_guest 0 0  
## 15 no\_of\_previous\_cancellations 0 0  
## 16 no\_of\_previous\_bookings\_not\_canceled 0 0  
## 17 avg\_price\_per\_room 0 0  
## 18 no\_of\_special\_requests 0 0  
## 19 booking\_status 0 0  
## 20 sqrt\_lead\_time 0 0

ggplot(df)+  
 geom\_boxplot(aes(room\_type\_reserved, avg\_price\_per\_room), color = "black", fill = cm.colors(7), size = 1, alpha = 0.5)



ggplot(df)+  
 geom\_boxplot(aes(booking\_status, lead\_time), color = "black", fill = cm.colors(2), size = 1, alpha = 0.5)



### Q1 - Identify the explanatory variable.

* Answer: In earlier stages of these projects we have identified two exploratory variables.
* avg\_price\_per\_room
* lead\_time

### Q2 - Identify the response variable.

* Answer: Above two exploratory variables have these two response variables.
* room\_type\_reserved (dependent on avg\_price\_per\_room)
* booking\_status (dependent on lead\_time)

Balancing the Data.

### room\_type\_reserved and booking\_status our response variables.

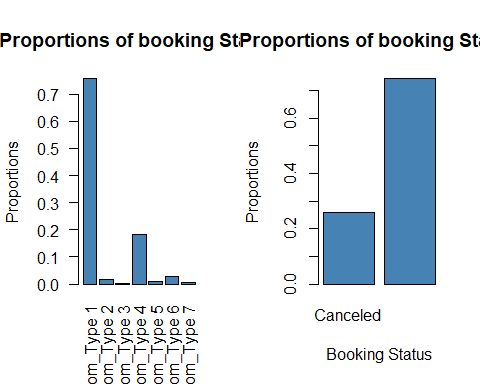
# Create a table of proportions for room types  
prop.table(table(df$room\_type\_reserved))

##   
## Room\_Type 1 Room\_Type 2 Room\_Type 3 Room\_Type 4 Room\_Type 5 Room\_Type 6   
## 0.756855281 0.017388219 0.000225821 0.183657010 0.007806955 0.029034131   
## Room\_Type 7   
## 0.005032583

# Create a table of proportions for booking status  
prop.table(table(df$booking\_status))

##   
## Canceled Not\_Canceled   
## 0.2576295 0.7423705

# Set up the plot area  
par(mfrow = c(1, 2))  
  
# Plot the first barplot  
barplot(prop.table(table(df$room\_type\_reserved)), main = "Proportions of booking Status",  
 ylab = "Proportions", col = "steelblue", las = 2)  
  
# Plot the second barplot  
barplot(prop.table(table(df$booking\_status)), main = "Proportions of booking Status", xlab = "Booking Status",  
 ylab = "Proportions", col = "steelblue")



#### As we can see both of these variables are imbalanced. So we will need to balance both of them before applying logistic regression.

table(df$room\_type\_reserved)

##   
## Room\_Type 1 Room\_Type 2 Room\_Type 3 Room\_Type 4 Room\_Type 5 Room\_Type 6   
## 23461 539 7 5693 242 900   
## Room\_Type 7   
## 156

table(df$booking\_status)

##   
## Canceled Not\_Canceled   
## 7986 23012

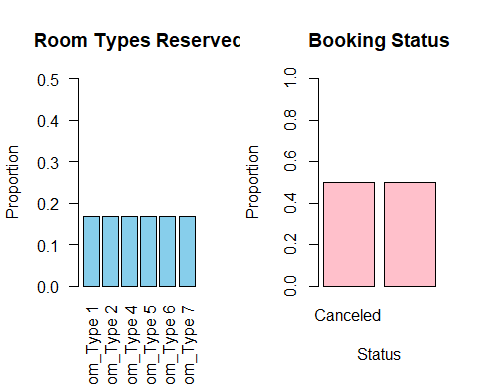
#### We will remove the room\_type 3 as It has only 7 values. Also, there is huge imbalance inbetween catagories. Thus, Many complex balancing techniques will fail such as SMOTE.

df <- df[df$room\_type\_reserved != "Room\_Type 3", ]

##### We will use upsampling for room\_type\_reserved as we would be left with very few value if we did downsampling. Whereas, for booking\_status we will do upsampling.

#Balancing room type reserved using upsample as most of them have very few values.  
balance\_data <- upSample(x = df[, -ncol(df)], y=factor(df$room\_type\_reserved))  
  
#Balancing booking status using downsample  
df\_balanced <- downSample(x = df[, -ncol(df)], y = factor(df$booking\_status))

# Combine the two bar graphs side by side  
par(mfrow = c(1, 2))  
  
# Create the first bar graph for room type  
barplot(prop.table(table(balance\_data$room\_type\_reserved)),   
 main = "Room Types Reserved",  
 ylab = "Proportion",  
 col = "skyblue",  
 ylim = c(0, 0.5),  
 las = 2)  
  
# Create the second bar graph for booking status  
barplot(prop.table(table(df\_balanced$booking\_status)),   
 main = "Booking Status",  
 xlab = "Status",  
 ylab = "Proportion",  
 col = "pink",  
 ylim = c(0, 1))



* Outliers in avg\_price\_per\_room

summary(balance\_data$avg\_price\_per\_room)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 87.3 123.2 128.9 174.0 540.0

* Let’s remove the outliers

P\_q1 = quantile(balance\_data$avg\_price\_per\_room, 0.25)  
P\_q3 = quantile(balance\_data$avg\_price\_per\_room, 0.75)  
  
P\_iqr = IQR(balance\_data$avg\_price\_per\_room)  
  
lowest\_price = P\_q1 - (1.5 \* P\_iqr)  
Highest\_price = P\_q3 + (1.5 \* P\_iqr)

### Q3 - Create a logistic regression model and display the full output of the model. (As we do not have any pair of continuous varible which can be identified as explanatory and response variables. Performing a linear regression is not an option.)

* Answer: First performing the logistic regression on avg\_price\_per\_room and room\_type\_reserved.

# split the data into training and testing sets  
set.seed(100)  
train\_indices <- sample(nrow(balance\_data), 0.7 \* nrow(balance\_data))  
train <- balance\_data[train\_indices, ]  
test <- balance\_data[-train\_indices, ]  
  
# convert room\_type\_reserved to factor  
train$room\_type\_reserved <- factor(train$room\_type\_reserved)  
  
# train the model  
multinom\_model <- multinom(room\_type\_reserved ~ avg\_price\_per\_room, data = train)

## # weights: 18 (10 variable)  
## initial value 176552.811060   
## iter 10 value 162063.394421  
## final value 160883.211068   
## converged

summary(multinom\_model)

## Call:  
## multinom(formula = room\_type\_reserved ~ avg\_price\_per\_room, data = train)  
##   
## Coefficients:  
## (Intercept) avg\_price\_per\_room  
## Room\_Type 2 0.2590519 -0.002908934  
## Room\_Type 4 -1.1956001 0.010702552  
## Room\_Type 5 -1.0567174 0.009602911  
## Room\_Type 6 -4.1813711 0.029995605  
## Room\_Type 7 -2.6650733 0.021064869  
##   
## Std. Errors:  
## (Intercept) avg\_price\_per\_room  
## Room\_Type 2 0.02183412 0.0002050428  
## Room\_Type 4 0.02641910 0.0002136020  
## Room\_Type 5 0.02594157 0.0002122086  
## Room\_Type 6 0.03732690 0.0002472342  
## Room\_Type 7 0.03141125 0.0002282723  
##   
## Residual Deviance: 321766.4   
## AIC: 321786.4

# get the intercepts for all categories  
coef\_mat <- coef(multinom\_model)  
intercepts <- coef\_mat[,1]  
intercepts

## Room\_Type 2 Room\_Type 4 Room\_Type 5 Room\_Type 6 Room\_Type 7   
## 0.2590519 -1.1956001 -1.0567174 -4.1813711 -2.6650733

* Now we will perform logistic regression on booking\_status and lead\_time.

# Create a new binary variable for booking status  
df\_balanced$binary\_booking\_status <- ifelse(df\_balanced$booking\_status == "Canceled", 1, 0)  
  
# Train logistic regression model  
model <- glm(binary\_booking\_status ~ lead\_time, data = df\_balanced, family = "binomial")

summary(model)

##   
## Call:  
## glm(formula = binary\_booking\_status ~ lead\_time, family = "binomial",   
## data = df\_balanced)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.861 -1.024 -0.110 1.111 1.552   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.847719 0.026973 -31.43 <2e-16 \*\*\*  
## lead\_time 0.013243 0.000341 38.84 <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: 22136 on 15967 degrees of freedom  
## Residual deviance: 20438 on 15966 degrees of freedom  
## AIC: 20442  
##   
## Number of Fisher Scoring iterations: 4

### Q4 - Using the variables noted in #1 and #2 above and the results of #3, write the equation for your model.

* Answer 4.1 : log(p(Room\_Type = i) / p(Room\_Type = 1)) = x0 + x1 \* avg\_price\_per\_room ; Where room\_type 1 is refrence catagory and i is other catagories. x0 is the Intercept, and x1 is the estimated coefficient (slope) of avg\_price\_per\_room where negative value suggests that with the increase in avg\_price\_per\_room the likelyhood of being in room\_type i decreases.
* Answer 4.2 : log(p/(1-p)) = -0.854157 + 0.013425 \* lead\_time; Where P is the probability of canceled booking. (-0.854157) is the entercept and 0.013425 is the slope.

### Q5) What the Intercept means in context of the data.

* Answer 5.1 : It means that the intercept represents the log-odds of the probability that an observation belongs to Room\_Type 1 (as it is the refrense catagory), when the average price per room is zero. But we have excluded the 0 values in our dataset as room prices, in most of the cases will be greater than 0.
* Answer 5.2 : the intercept represents the estimated log odds of the dependent variable (binary\_booking\_status) when the value of the predictor variable (lead\_time) is zero. Which basically the baseline probability of booking being canceled or not.

### Q6) Is the intercept a useful/meaningful value in the context of our data? If yes, explain. If not, explain what purpose it serves.

* Answer 6.1 : Yes and No, the intercept is basically the log odds of the refrense catagory(response) when all the predictor are equal to zero. As we have excluded 0 values it does not make much of a difference. But on the other hand, it provides us with the baseline for comparing every catagories (i.e. all the room types).
* Answer 6.2 : Yes, the intercept (-0.85) in this case suggests that when lead\_time is 0 (booking is done near to arrival date) the probability of booking being canceled is very less(i.e. not cancelled).

### Q7) Explain what the slope means in the context of the data.

* Answer 7.1 : The changes in log odds of the room\_type\_reserved with an increase in avg\_price\_per\_room is denoted as slope. When the slope is positive it denotes the increase in log odds of that perticular type and if it’s negative the opposite is true.
* Answer 7.2 : The slope of lead\_time is 0.013425, which means with the increase in lead\_time the log odds of booking being canclled will increase by the value of slope (0.013).

## Model Diagnostics

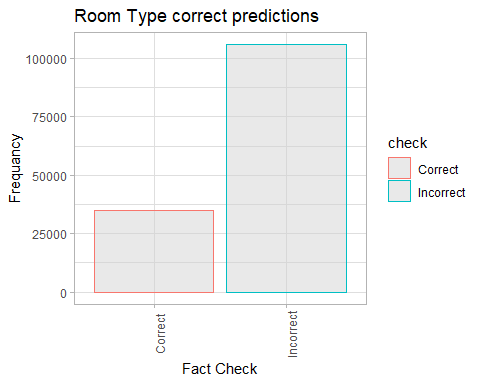
### Q1 Create two new data columns based on your best model: predicted values for your response variable and the corresponding residuals.

* Answer : As these are catagorical variables we will just show that if the prediction is correct or not instead of residuals.

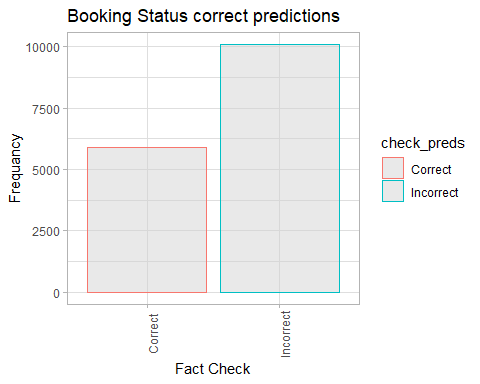
# get predicted probabilities for each category  
probs <- predict(multinom\_model, newdata = balance\_data, type = "probs")  
  
# calculate predicted values (category with maximum predicted probability)  
balance\_data$Room\_Type\_prediction <- paste0("Room\_Type ", apply(probs, 1, which.max))  
  
# Create check column for checking the predictions are true or not?  
balance\_data$check <- ifelse(balance\_data$Class == balance\_data$Room\_Type\_prediction, "Correct", "Incorrect")  
  
# Predict binary\_booking\_status using the trained logistic regression model  
df\_balanced$predictions <- predict(model, newdata = df\_balanced, type = "response")  
  
# Convert predictions to binary values  
df\_balanced$predictions\_binary <- ifelse(df\_balanced$predictions > 0.5, "Not\_Canceled", "Canceled")  
  
# Check if predictions are correct  
df\_balanced$check\_preds <- ifelse(df\_balanced$predictions\_binary == df\_balanced$booking\_status, "Correct", "Incorrect")

#### Checking how both of our models work on the whole dataset.

ggplot(balance\_data)+  
 geom\_bar(aes(check, color = check), fill = "lightgrey", alpha = 0.5)+  
 labs(  
 title = "Room Type correct predictions",  
 x = "Fact Check",  
 y = "Frequancy"  
 )+  
 theme\_light()+  
 theme(axis.text.x = element\_text(angle = 90))



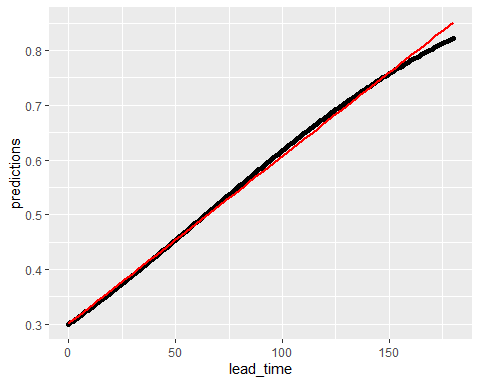
ggplot(df\_balanced)+  
 geom\_bar(aes(check\_preds, color = check\_preds), fill = "lightgrey", alpha = 0.5)+  
 labs(  
 title = "Booking Status correct predictions",  
 x = "Fact Check",  
 y = "Frequancy"  
 )+  
 theme\_light()+  
 theme(axis.text.x = element\_text(angle = 90))



### Q2 Checking for linearity between independent variable and log odds of the dependent variable.

ggplot(df\_balanced, aes(x = lead\_time, y = predictions)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE, color = "red")

## `geom\_smooth()` using formula = 'y ~ x'



* Answer : As we can see, there is clear and consistent linear pattern, it suggests that linearity assumption in correct and our model is correct for our data.

### Q3 - Goodness of Fit of the model

summary(multinom\_model)

## Call:  
## multinom(formula = room\_type\_reserved ~ avg\_price\_per\_room, data = train)  
##   
## Coefficients:  
## (Intercept) avg\_price\_per\_room  
## Room\_Type 2 0.2590519 -0.002908934  
## Room\_Type 4 -1.1956001 0.010702552  
## Room\_Type 5 -1.0567174 0.009602911  
## Room\_Type 6 -4.1813711 0.029995605  
## Room\_Type 7 -2.6650733 0.021064869  
##   
## Std. Errors:  
## (Intercept) avg\_price\_per\_room  
## Room\_Type 2 0.02183412 0.0002050428  
## Room\_Type 4 0.02641910 0.0002136020  
## Room\_Type 5 0.02594157 0.0002122086  
## Room\_Type 6 0.03732690 0.0002472342  
## Room\_Type 7 0.03141125 0.0002282723  
##   
## Residual Deviance: 321766.4   
## AIC: 321786.4

summary(model)

##   
## Call:  
## glm(formula = binary\_booking\_status ~ lead\_time, family = "binomial",   
## data = df\_balanced)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.861 -1.024 -0.110 1.111 1.552   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.847719 0.026973 -31.43 <2e-16 \*\*\*  
## lead\_time 0.013243 0.000341 38.84 <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: 22136 on 15967 degrees of freedom  
## Residual deviance: 20438 on 15966 degrees of freedom  
## AIC: 20442  
##   
## Number of Fisher Scoring iterations: 4

# Conduct Hosmer-Lemeshow goodness-of-fit test  
hoslem.test(df\_balanced$binary\_booking\_status, fitted(model))

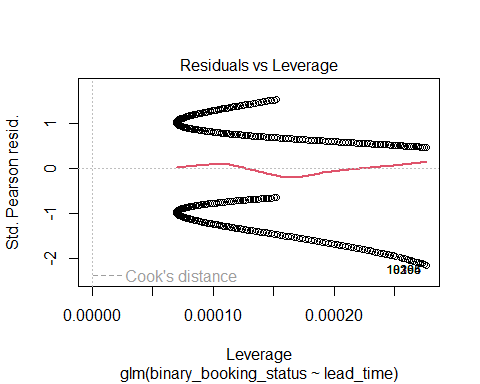
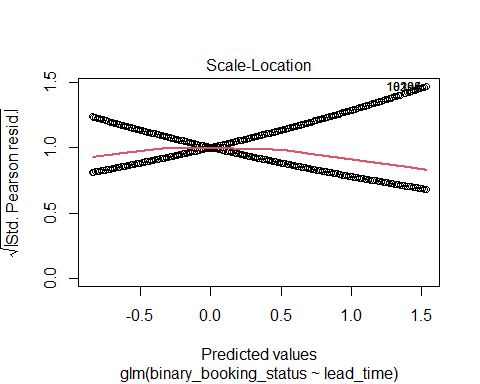
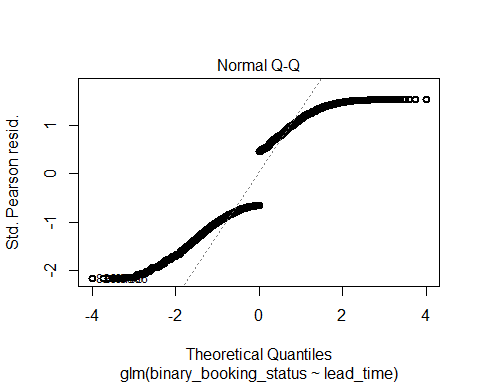
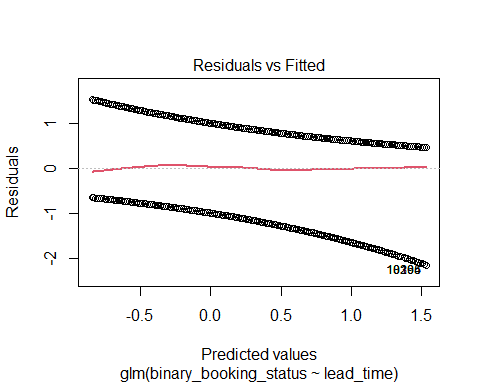
##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: df\_balanced$binary\_booking\_status, fitted(model)  
## X-squared = 264.9, df = 8, p-value < 2.2e-16

* Answer 3.1: As we can see the AIC value and deviance both are quite high, and this suggest that model in not a well fitted one.
* Answer 3.2: As we can see in the output the AIC value is very high which means that the model is not well fitted with data. To add to this, Hosmer and Lemeshow goodness of fit (GOF) test also gives very low p-value which suggests the same. I.e. there are also other factors contributing to booking cancellation other than lead time.

Understanding the model

#### Plotting the model

# plot the model  
plot(model, col = "black", lwd = 2)



# add a title and axis labels  
xlab("Lead Time")

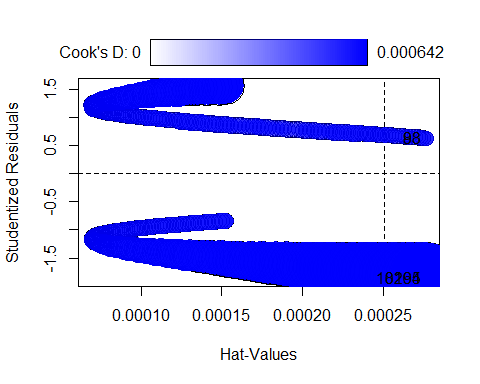
## $x  
## [1] "Lead Time"  
##   
## attr(,"class")  
## [1] "labels"

ylab("Probability of Booking")

## $y  
## [1] "Probability of Booking"  
##   
## attr(,"class")  
## [1] "labels"

#### Plotting Influencial plot to check for the influencial points.

influencePlot(model)



## StudRes Hat CookD  
## 83 0.6244508 0.0002762116 2.974201e-05  
## 98 0.6244508 0.0002762116 2.974201e-05  
## 8294 -1.8609642 0.0002762116 6.419979e-04  
## 10165 -1.8609642 0.0002762116 6.419979e-04

## Conclusion

#### Checking for the accuracy of the model.

# Convert both predicted and actual values to factors with the same levels  
pred <- factor(round(df\_balanced$predictions), levels = c("0", "1"))  
actual <- factor(df\_balanced$binary\_booking\_status, levels = c("0", "1"))  
  
# Calculate the accuracy using confusionMatrix() from the caret package  
accuracy <- confusionMatrix(pred, actual)$overall[1]  
  
# Print the accuracy  
cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.6310746

* In the conclusion, we can safely say that the accuracy of our model is average (63%) and it needs to be hypertuned.
* Also, the response variable’s variablity is not just dependent on our predictor variable, there are other hidden factors affecting the response variable as well.
* The multinominal logistic regression model is also performing poorly. The reason might be the data imbalnce.
* the influence plot show influencial points which are at the top left and bottom right corner.

### Refrences

1. <https://stats.stackexchange.com/questions/297716/how-to-write-a-regression-equation-using-the-output-from-a-summary>
2. <https://r-graph-gallery.com/index.html>
3. <https://www.geeksforgeeks.org/>
4. <https://www.youtube.com/results?search_query=how+to+perform+oversampling+in+R>
5. <https://stats.oarc.ucla.edu/r/dae/multinomial-logistic-regression/>
6. <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>
7. <https://www.analyticsvidhya.com/blog/>
8. <https://stats.stackexchange.com/questions/65244/how-to-determine-the-accuracy-of-logistic-regression-in-r>
9. <https://topepo.github.io/caret/>
10. <https://rdrr.io/cran/car/src/R/influencePlot.R>
11. <https://search.r-project.org/CRAN/refmans/sjPlot/html/plot_model.html>
12. <https://towardsdatascience.com/visualizing-models-101-using-r-c7c937fc5f04>
13. <https://topepo.github.io/caret/subsampling-for-class-imbalances.html>