STA303 - Final Project

Cristina Su Lam

2024-03-09

Load packages

```
suppressMessages({
  suppressWarnings({
    library(knitr)
    library(gmodels)
    library(readr)
    library(magrittr)
    library(dplyr)
    library(tidyr)
    library(glmnet)
    library(dtplyr)
    library(glmnet)
    library(MASS)
    library(rms)
    library(pROC)
  })
})
```

1. Read the Dataset & Drop Missing Values

```
customer_booking <- read_csv("customer_booking.csv")
```

```
## Rows: 50000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): sales_channel, trip_type, flight_day, route, booking_origin
## dbl (9): num_passengers, purchase_lead, length_of_stay, flight_hour, wants_e...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Rename the category "CircleTrip" to "RoundTrip" in the "trip_type" variable
customer_booking <- customer_booking %>%
  mutate(trip_type = ifelse(trip_type == "CircleTrip", "RoundTrip", trip_type))

# Filter the dataset to only include customers from Australia
customer_booking_malaysia <- customer_booking[customer_booking$booking_origin == "Malaysia", ]
```

```
# Remove the column 'booking_origin'
customer_booking_malaysia <- customer_booking_malaysia[, -which(names(customer_booking_malaysia) == "boo

# Check the first few rows of the dataset
head(customer_booking_malaysia)
```

```
## # A tibble: 6 x 13
##   num_pa~1 sales~2 trip_~3 purch~4 lengt~5 fligh~6 fligh~7 route wants~8 wants~9
##      <dbl> <chr>   <chr>     <dbl>   <dbl>   <dbl> <chr>   <chr>   <dbl>   <dbl>
## 1        1 Intern~ RoundT~      15      31      17 Mon     AKLK~       0       0
## 2        1 Intern~ RoundT~      31     274      10 Tue     AKLK~       1       0
## 3        1 Intern~ RoundT~     316      35      16 Tue     AKLK~       1       0
## 4        2 Intern~ RoundT~     232      17       3 Tue     AKLK~       1       1
## 5        1 Intern~ RoundT~     156      19      14 Mon     AKLK~       1       0
## 6        1 Intern~ RoundT~       6     106      19 Tue     AKLK~       0       0
## # ... with 3 more variables: wants_in_flight_meals <dbl>,
## #   flight_duration <dbl>, booking_complete <dbl>, and abbreviated variable
## #   names 1: num_passengers, 2: sales_channel, 3: trip_type, 4: purchase_lead,
## #   5: length_of_stay, 6: flight_hour, 7: flight_day, 8: wants_extra_baggage,
## #   9: wants_preferred_seat
```

```
# Check for missing values in the entire data set
missing_values <- any(is.na(customer_booking_malaysia))

# Print the result
if (missing_values) {
  print("The dataset contains missing values.")
} else {
  print("The dataset does not contain missing values.")
}
```

```
## [1] "The dataset does not contain missing values."
```

2. Logistic Regression Model (All variables)

```
# Fit logistic regression model using original data set
logit_model <- glm(booking_complete ~ sales_channel + trip_type + purchase_lead + length_of_stay + fligh
                      wants_preferred_seat + num_passengers + flight_day + route + wants_extra_baggage +
                      wants_in_flight_meals + flight_duration,
                    family = binomial(link = logit),
                    data = customer_booking_malaysia)
```

3. Model Selection

Perform stepwise AIC selection

```r
sel.var.aic <- step(logit_model, trace = 0, k = 2, direction = "both")
select_var_aic <- attr(terms(sel.var.aic), "term.labels")
select_var_aic
```

```
## [1] "sales_channel"         "length_of_stay"        "wants_preferred_seat"
## [4] "num_passengers"        "wants_extra_baggage"   "wants_in_flight_meals"
## [7] "flight_duration"
```

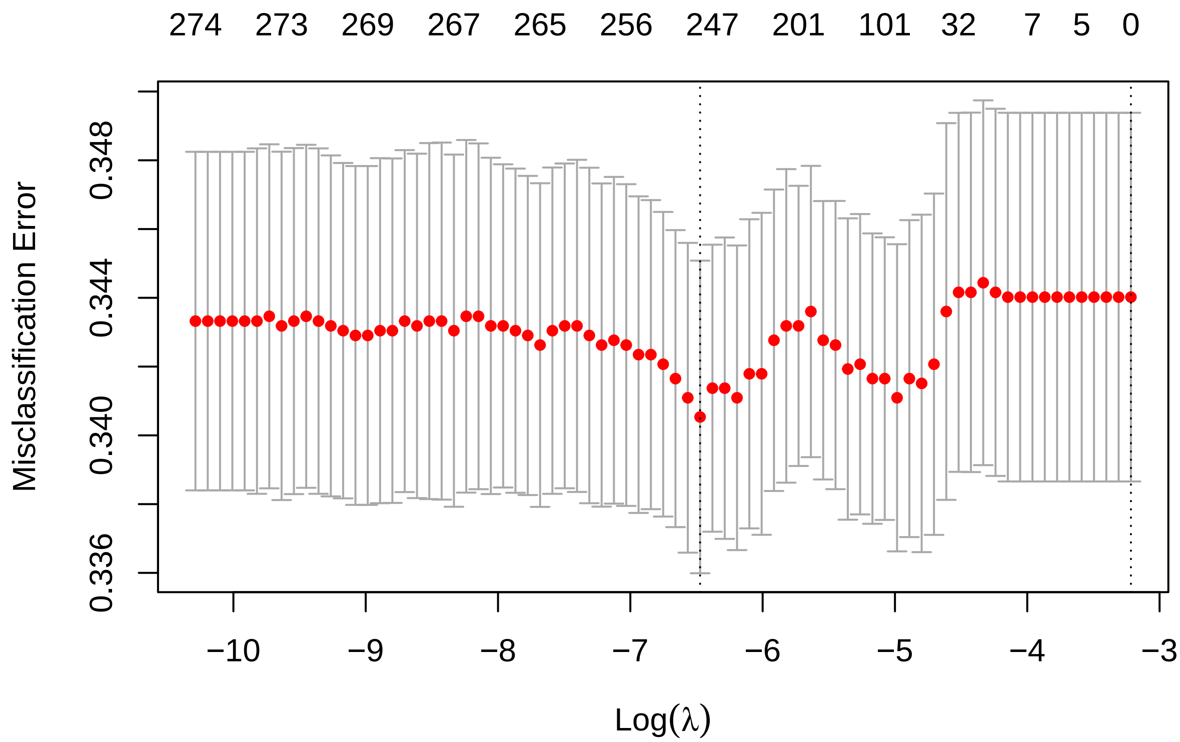
Perform stepwise BIC selection

```r
sel.var.bic <- step(logit_model, trace = 0, k = log(nrow(customer_booking_malaysia)), direction = "both
select_var_bic <- attr(terms(sel.var.bic), "term.labels")
select_var_bic
```

```
## [1] "sales_channel"         "length_of_stay"        "wants_preferred_seat"
## [4] "wants_extra_baggage"   "wants_in_flight_meals" "flight_duration"
```

LASSO Method

```r
set.seed(1007928566)

# x contains the predictors and y contains the response variable
x <- model.matrix(booking_complete ~ ., data = customer_booking_malaysia)[,-1]
y <- customer_booking_malaysia$booking_complete

# Fit the model
fit <- glmnet(x, y, family = "binomial")

# Make predictions for all observations
predictions <- predict(fit, newx = x, type = "class", s = c(0.05, 0.01))

# Evaluate model performance
cv.out <- cv.glmnet(x, y, family = "binomial", type.measure = "class", alpha = 1)

# Plot the cross-validation results
plot(cv.out)
```




```r
# Get the best lambda value
best.lambda <- cv.out$lambda.1se

# Get the coefficients at the selected lambda
co <- coef(cv.out, s = "lambda.1se")

# Threshold for variable selection
thresh <- 0.00

# Select variables
inds <- which(abs(co) > thresh)
variables <- row.names(co)[inds]
sel.var.lasso <- variables[!(variables %in% '(Intercept)')]
sel.var.lasso
```

```
## character(0)
```

6. Model Diagnostics & Validation

6.1.1 Stepwise AIC Model

Fit Logistic Regression Model with AIC selection

```r
aic.logit <- glm(booking_complete ~ sales_channel + length_of_stay + flight_duration + num_passengers +
                   wants_extra_baggage + wants_preferred_seat + wants_in_flight_meals, family = binomial
                 data = customer_booking_malaysia)
summary(aic.logit)
```

```
##
## Call:
## glm(formula = booking_complete ~ sales_channel + length_of_stay +
##     flight_duration + num_passengers + wants_extra_baggage +
##     wants_preferred_seat + wants_in_flight_meals, family = binomial(link = logit),
##     data = customer_booking_malaysia)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2130  -0.9456  -0.8053   1.3303   2.8451
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.4401197  0.1206179  -3.649 0.000263 ***
## sales_channelMobile   -0.4183914  0.0854446  -4.897 9.75e-07 ***
## length_of_stay        -0.0069867  0.0009081  -7.693 1.43e-14 ***
## flight_duration       -0.0599643  0.0150626  -3.981 6.86e-05 ***
## num_passengers        -0.0359921  0.0229337  -1.569 0.116555
## wants_extra_baggage    0.3974108  0.0638962   6.220 4.98e-10 ***
## wants_preferred_seat   0.2143687  0.0569268   3.766 0.000166 ***
## wants_in_flight_meals  0.2346237  0.0542727   4.323 1.54e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9235.3  on 7173  degrees of freedom
## Residual deviance: 9030.1  on 7166  degrees of freedom
## AIC: 9046.1
##
## Number of Fisher Scoring iterations: 4
```

Checking influential points

```r
# DFBETAS
df.aic <- dfbetas(aic.logit)
n <- nrow(customer_booking_malaysia)
beta_cut <- 2 / sqrt(n)
influential_points <- apply(abs(df.aic) > beta_cut, 1, any)
sum(influential_points)
```

```
## [1] 1591
```

Checking outliers

```r
ri.aic <- rstandard(aic.logit)
outliers_obs <- which(abs(ri.aic) > 2)
length(outliers_obs)
```

```
## [1] 13
```

VIF to check for multicollinearity

```r
vif(aic.logit)
```

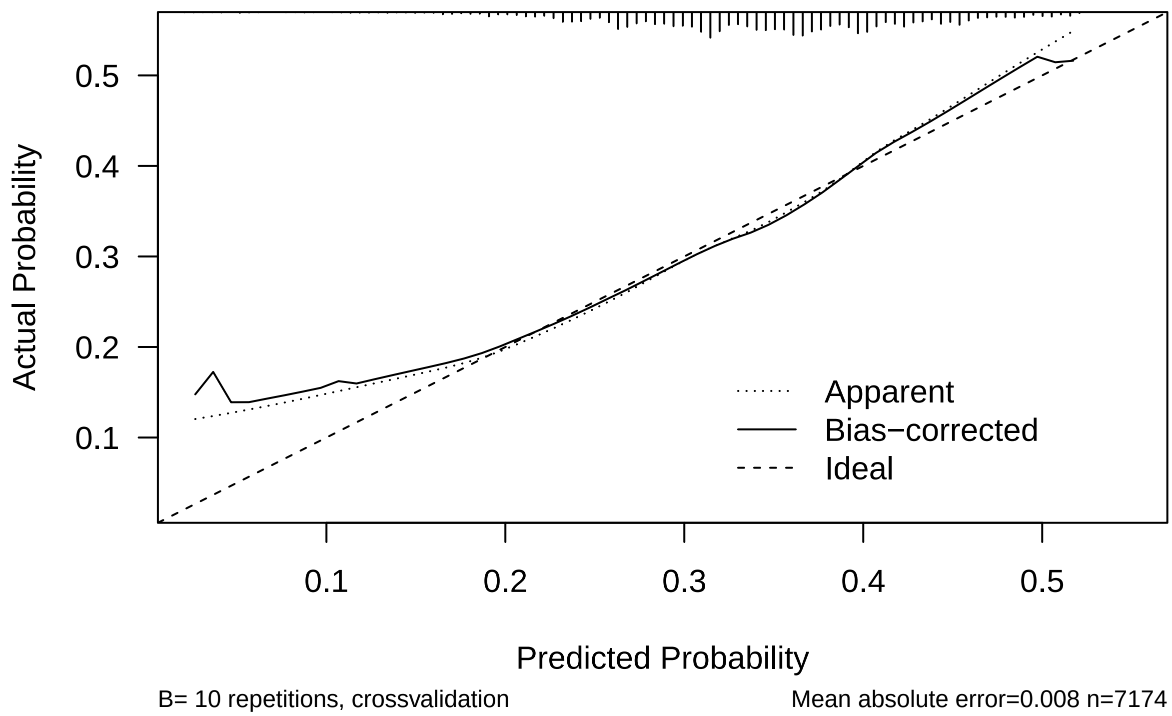
```
##   sales_channelMobile        length_of_stay       flight_duration
##              1.011186              1.098832              1.041416
##        num_passengers   wants_extra_baggage  wants_preferred_seat
##              1.075829              1.112047              1.133795
## wants_in_flight_meals
##              1.142903
```

Cross-Validation and Calibration

```r
set.seed(1007928566)
lrm.aic <- lrm(booking_complete ~ .,
                 data = customer_booking_malaysia[, which(colnames(customer_booking_malaysia) %in% c(se

                 x = TRUE, y = TRUE, model = TRUE)
lrm.aic
```

```
## Logistic Regression Model
##
## lrm(formula = booking_complete ~ ., data = customer_booking_malaysia[,
##     which(colnames(customer_booking_malaysia) %in% c(select_var_aic,
##         "booking_complete"))], model = TRUE, x = TRUE, y = TRUE)
##
##                        Model Likelihood       Discrimination    Rank Discrim.
##                              Ratio Test              Indexes          Indexes
## Obs           7174    LR chi2     205.26       R2       0.039    C       0.606
##  0            4706    d.f.             7      R2(7,7174)0.027    Dxy     0.212
##  1            2468    Pr(> chi2) <0.0001    R2(7,4856.9)0.040    gamma   0.212
## max |deriv| 3e-06                              Brier    0.219    tau-a   0.096
##
##                        Coef    S.E.   Wald Z Pr(>|Z|)
## Intercept              -0.4401 0.1206 -3.65  0.0003
## num_passengers         -0.0360 0.0229 -1.57  0.1166
## sales_channel=Mobile   -0.4184 0.0854 -4.90  <0.0001
## length_of_stay         -0.0070 0.0009 -7.69  <0.0001
## wants_extra_baggage     0.3974 0.0639  6.22  <0.0001
## wants_preferred_seat    0.2144 0.0569  3.77  0.0002
## wants_in_flight_meals   0.2346 0.0543  4.32  <0.0001
## flight_duration        -0.0600 0.0151 -3.98  <0.0001
```

```
cross.calib <- calibrate(lrm.aic, method = "crossvalidation", B = 10)
plot(cross.calib, las=1, xlab = "Predicted Probability")
```




```
##
## n=7174   Mean absolute error=0.008   Mean squared error=0.00012
## 0.9 Quantile of absolute error=0.014
```

AUC and ROC Curve

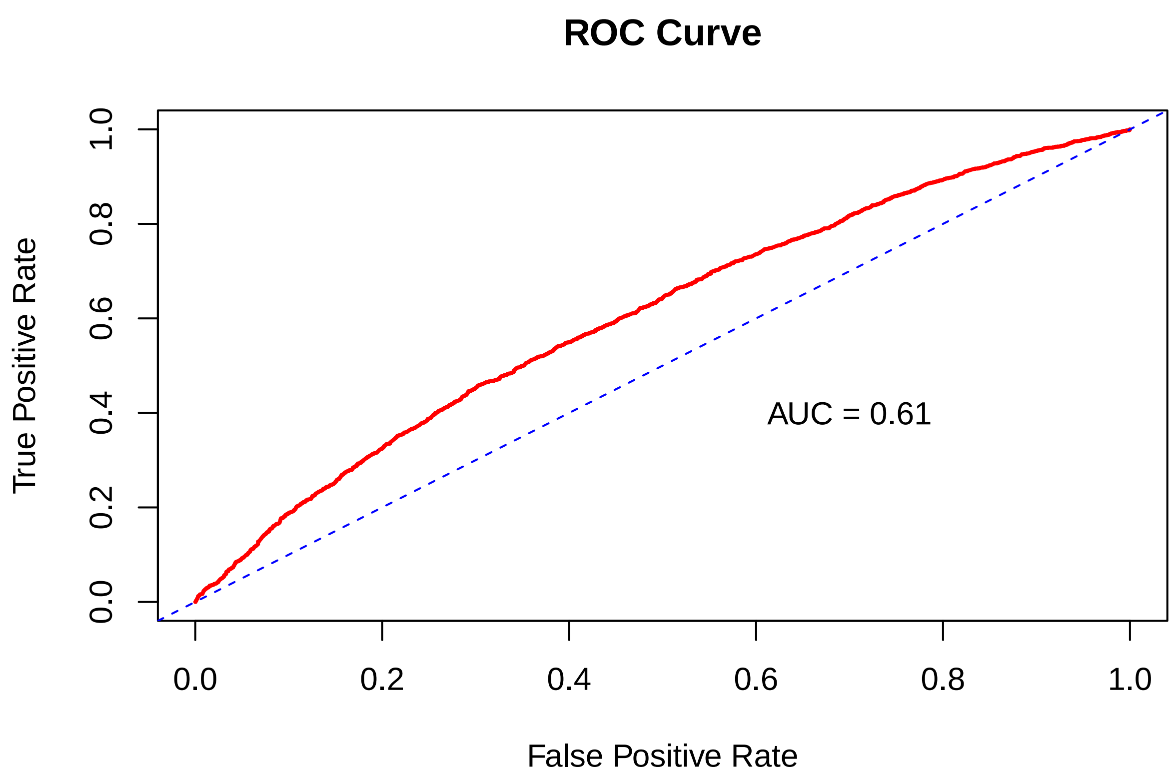
```
# Predicting probabilities using the logistic regression model
p <- predict(lrm.aic, type = "fitted")

# Generating ROC curve
roc_aic.logit <- roc(customer_booking_malaysia$booking_complete ~ p)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Extracting True Positive Rate (TPR) and False Positive Rate (FPR)
TPR <- roc_aic.logit$sensitivities
FPR <- 1 - roc_aic.logit$specificities
```

```r
# Plotting ROC curve
plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2, col = 'red',
     xlab = "False Positive Rate", ylab = "True Positive Rate", main = "ROC Curve")
abline(a = 0, b = 1, lty = 2, col = 'blue')  # Adding diagonal reference line
text(0.7, 0.4, label = paste("AUC =", round(auc(roc_aic.logit), 2)))  # Adding AUC value as text
```




```r
# Calculating and printing the AUC
auc_value <- auc(roc_aic.logit)
print(paste("AUC value:", round(auc_value, 2)))
```

```
## [1] "AUC value: 0.61"
```

6.1.2 Stepwise AIC Model (Outliers Removed)

Fit logistic regression with the cleaned dataset (without outliers)

```r
# Combine influential points and outliers without repetition
all_outliers <- unique(outliers_obs)

# Remove outliers and influential points from the dataset
cleaned_data_aic <- customer_booking_malaysia[-all_outliers, ]

# Fit logistic regression model with all variables
new.logit_model <- glm(booking_complete ~ sales_channel + trip_type + purchase_lead + length_of_stay + 
```

```r
                         wants_in_flight_meals + flight_duration,
                       family = binomial(link = logit),
                       data = cleaned_data_aic)
```

Perform Stepwise AIC selection with 'new.logit_model'

```r
newsel.var.aic <- step(new.logit_model, trace = 0, k = 2, direction = "both")
newselect_var_aic <- attr(terms(newsel.var.aic), "term.labels")
newselect_var_aic
```

```
## [1] "sales_channel"         "length_of_stay"        "wants_preferred_seat"
## [4] "num_passengers"        "wants_extra_baggage"   "wants_in_flight_meals"
## [7] "flight_duration"
```

Fit Logistic Regression Model with new AIC selection

```r
new_aic.logit <- glm(booking_complete ~ sales_channel + length_of_stay + flight_duration + num_passenge
summary(new_aic.logit)
```

```
##
## Call:
## glm(formula = booking_complete ~ sales_channel + length_of_stay +
##     flight_duration + num_passengers + wants_extra_baggage +
##     wants_preferred_seat + wants_in_flight_meals, family = binomial(link = logit),
##     data = cleaned_data_aic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2443  -0.9461  -0.7887   1.3173   2.2080
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.369310   0.121508  -3.039 0.002371 **
## sales_channelMobile   -0.438661   0.085796  -5.113 3.17e-07 ***
## length_of_stay        -0.010080   0.001005 -10.032  < 2e-16 ***
## flight_duration       -0.062329   0.015158  -4.112 3.92e-05 ***
## num_passengers        -0.048247   0.023043  -2.094 0.036278 *
## wants_extra_baggage    0.436447   0.064319   6.786 1.16e-11 ***
## wants_preferred_seat   0.205245   0.057239   3.586 0.000336 ***
## wants_in_flight_meals  0.233041   0.054559   4.271 1.94e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9207.5  on 7160  degrees of freedom
## Residual deviance: 8948.5  on 7153  degrees of freedom
## AIC: 8964.5
##
## Number of Fisher Scoring iterations: 4
```

Checking influential points

```r
# DFBETAS
new_df.aic <- dfbetas(new_aic.logit)
n.aic <- nrow(cleaned_data_aic)
n.beta_cut <- 2 / sqrt(n.aic)
influential_points_naic <- apply(abs(new_df.aic) > n.beta_cut, 1, any)
sum(influential_points_naic)
```

```
## [1] 1570
```

Checking outliers

```r
ri.naic <- rstandard(new_aic.logit)
outliers_obs_naic <- which(abs(ri.naic) > 2)
length(outliers_obs_naic)
```

```
## [1] 8
```

VIF to check for multicollinearity

```r
vif(new_aic.logit)
```

```
##   sales_channelMobile        length_of_stay       flight_duration
##              1.011582              1.105337              1.040238
##        num_passengers   wants_extra_baggage  wants_preferred_seat
##              1.077804              1.116952              1.134999
## wants_in_flight_meals
##              1.142646
```

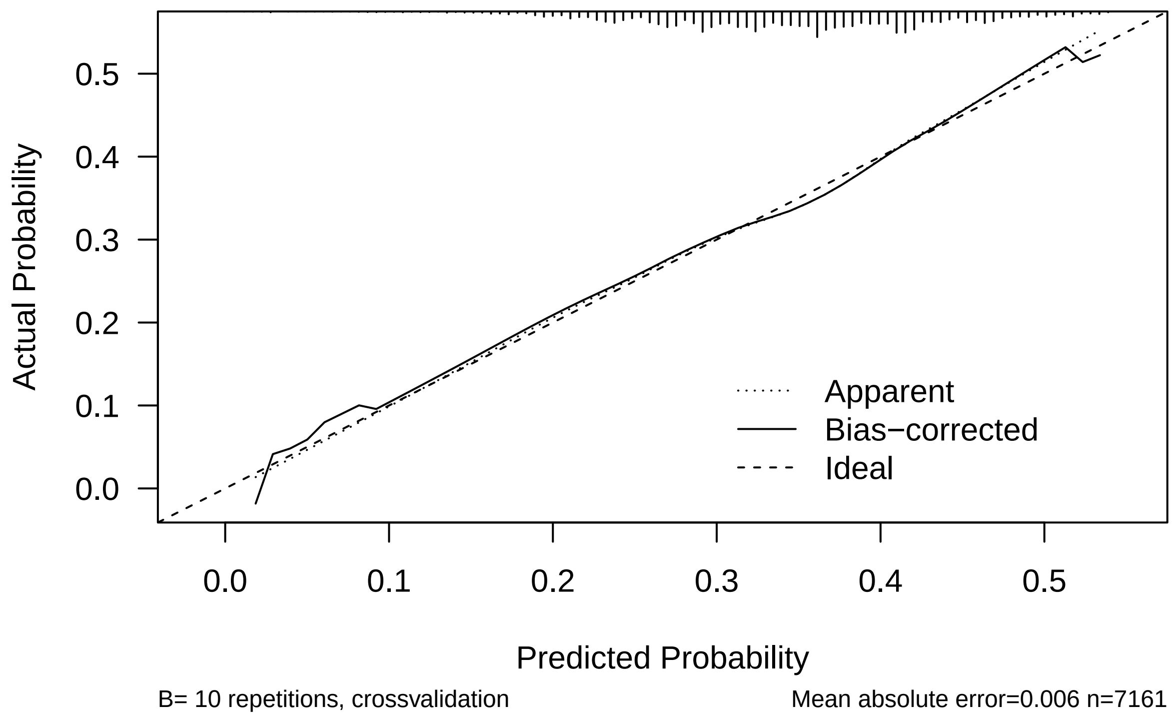
Cross-Validation and Calibration

```r
set.seed(1007928566)
new_lrm.aic <- lrm(booking_complete ~ .,
                   data = cleaned_data_aic[, which(colnames(cleaned_data_aic) %in% c(newselect_var_aic,"bo
                   x = TRUE, y = TRUE, model = TRUE)
new_lrm.aic
```

```
## Logistic Regression Model
##
## lrm(formula = booking_complete ~ ., data = cleaned_data_aic[,
##     which(colnames(cleaned_data_aic) %in% c(newselect_var_aic,
##         "booking_complete"))], model = TRUE, x = TRUE, y = TRUE)
##
##                        Model Likelihood       Discrimination    Rank Discrim.
##                              Ratio Test              Indexes          Indexes
## Obs          7161    LR chi2     259.02       R2        0.049    C       0.610
##  0           4706    d.f.             7      R2(7,7161)0.035    Dxy     0.221
##  1           2455    Pr(> chi2) <0.0001    R2(7,4840.1)0.051    gamma   0.221
## max |deriv| 3e-12                             Brier     0.218    tau-a   0.100
##
##                         Coef    S.E.   Wald Z Pr(>|Z|)
```

```
## Intercept              -0.3693 0.1215  -3.04 0.0024
## num_passengers         -0.0482 0.0230  -2.09 0.0363
## sales_channel=Mobile   -0.4387 0.0858  -5.11 <0.0001
## length_of_stay         -0.0101 0.0010 -10.03 <0.0001
## wants_extra_baggage     0.4364 0.0643   6.79 <0.0001
## wants_preferred_seat    0.2052 0.0572   3.59 0.0003
## wants_in_flight_meals   0.2330 0.0546   4.27 <0.0001
## flight_duration        -0.0623 0.0152  -4.11 <0.0001
```

```
nacross.calib <- calibrate(new_lrm.aic, method = "crossvalidation", B = 10)
plot(nacross.calib, las=1, xlab = "Predicted Probability")
```




```
##
## n=7161   Mean absolute error=0.006   Mean squared error=6e-05
## 0.9 Quantile of absolute error=0.012
```

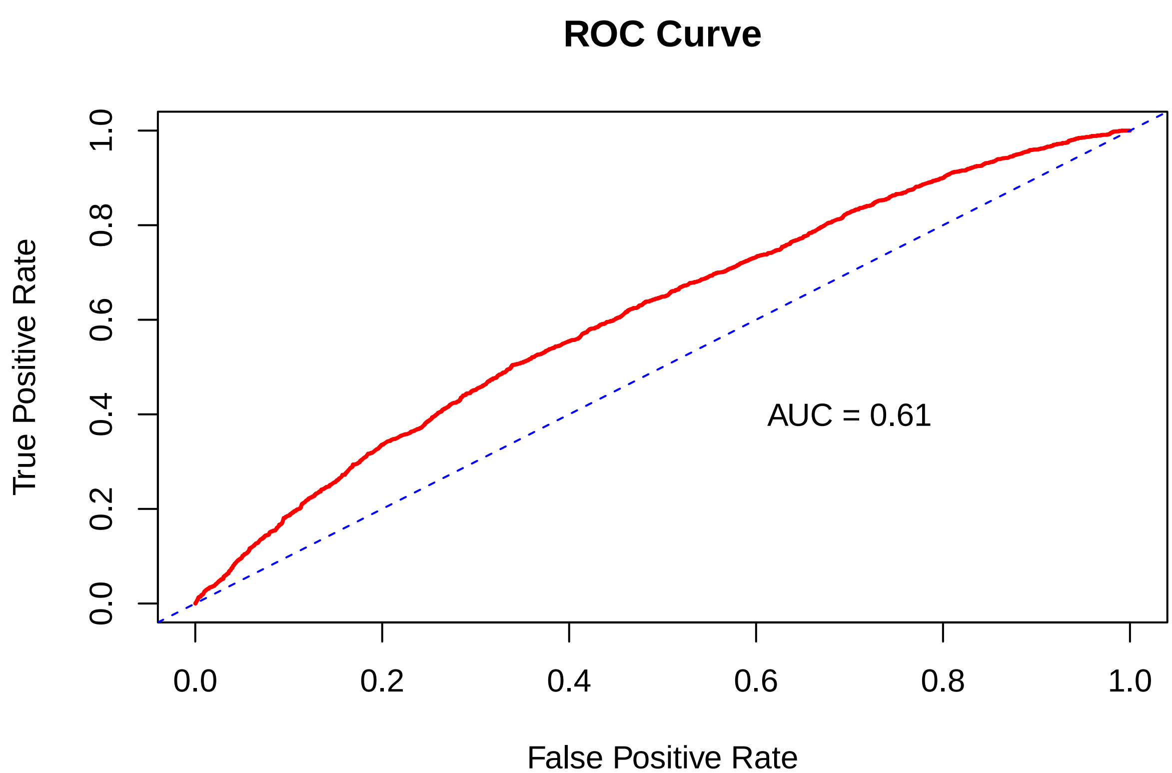
AUC and ROC Curve

```
# Predicting probabilities using the logistic regression model
p <- predict(new_lrm.aic, type = "fitted")

# Generating ROC curve
newroc_aic.logit <- roc(cleaned_data_aic$booking_complete ~ p)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Extracting True Positive Rate (TPR) and False Positive Rate (FPR)
TPR <- newroc_aic.logit$sensitivities
FPR <- 1 - newroc_aic.logit$specificities

# Plotting ROC curve
plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2, col = 'red',
     xlab = "False Positive Rate", ylab = "True Positive Rate", main = "ROC Curve")
abline(a = 0, b = 1, lty = 2, col = 'blue')  # Adding diagonal reference line
text(0.7, 0.4, label = paste("AUC =", round(auc(newroc_aic.logit), 2)))  # Adding AUC value as text
```




```
# Calculating and printing the AUC
newauc_value <- auc(newroc_aic.logit)
print(paste("AUC value:", round(newauc_value, 2)))
```

```
## [1] "AUC value: 0.61"
```

6.2.1 Stepwise BIC Model

Fit Logistic Regression Model with BIC selection

```
bic.logit <- glm(booking_complete ~ sales_channel + length_of_stay + flight_duration + wants_extra_bagga
summary(bic.logit)
```

```
##
## Call:
## glm(formula = booking_complete ~ sales_channel + length_of_stay +
##     flight_duration + wants_extra_baggage + wants_preferred_seat +
##     wants_in_flight_meals, family = binomial(link = logit), data = customer_booking_malaysia)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1977  -0.9453  -0.8061   1.3284   2.8047
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.4880846  0.1166828  -4.183 2.88e-05 ***
## sales_channelMobile   -0.4147637  0.0853910  -4.857 1.19e-06 ***
## length_of_stay        -0.0066706  0.0008809  -7.573 3.66e-14 ***
## flight_duration       -0.0610123  0.0150477  -4.055 5.02e-05 ***
## wants_extra_baggage    0.3828088  0.0631949   6.058 1.38e-09 ***
## wants_preferred_seat   0.2145329  0.0569095   3.770 0.000163 ***
## wants_in_flight_meals  0.2300297  0.0541853   4.245 2.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9235.3  on 7173  degrees of freedom
## Residual deviance: 9032.5  on 7167  degrees of freedom
## AIC: 9046.5
##
## Number of Fisher Scoring iterations: 4
```

Checking influential points

```
df.bic <- dfbetas(bic.logit)
n.bic <- nrow(customer_booking_malaysia)
beta_cut_bic <- 2 / sqrt(n.bic)
influential_points_bic <- apply(abs(df.bic) > beta_cut_bic, 1, any)
sum(influential_points_bic)
```

```
## [1] 1351
```

Checking outliers

```
ri.bic <- rstandard(bic.logit)
outliers_obs_bic <- which(ri.bic > 2 | ri.bic < -2)
length(outliers_obs_bic )
```

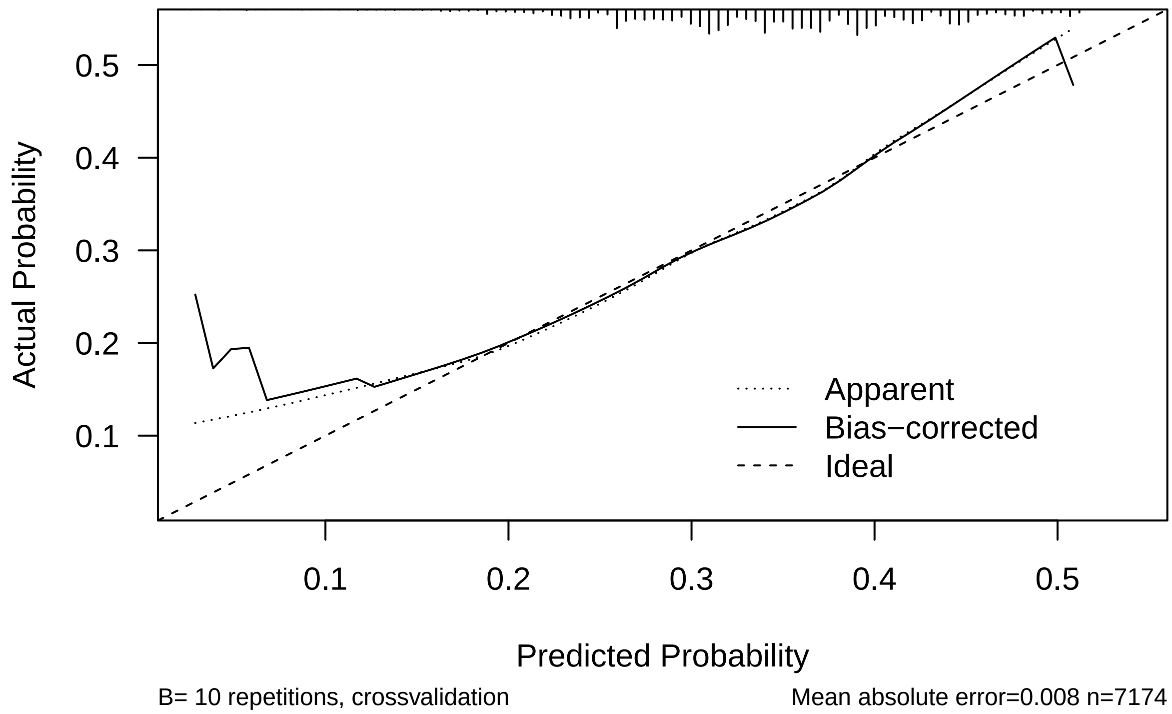
```
## [1] 12
```

Cross-Validation and Calibration

```
lrm.bic <- lrm(booking_complete ~ .,
                 data = customer_booking_malaysia[,which(colnames(customer_booking_malaysia) %in% c(sele
                 x = TRUE, y = TRUE, model = TRUE)
lrm.aic
```

```
## Logistic Regression Model
##
## lrm(formula = booking_complete ~ ., data = customer_booking_malaysia[,
##     which(colnames(customer_booking_malaysia) %in% c(select_var_aic,
##         "booking_complete"))], model = TRUE, x = TRUE, y = TRUE)
##
##                        Model Likelihood       Discrimination    Rank Discrim.
##                              Ratio Test              Indexes          Indexes
## Obs          7174    LR chi2     205.26       R2       0.039    C       0.606
##  0           4706    d.f.             7      R2(7,7174)0.027    Dxy     0.212
##  1           2468    Pr(> chi2) <0.0001    R2(7,4856.9)0.040    gamma   0.212
## max |deriv| 3e-06                             Brier    0.219    tau-a   0.096
##
##                        Coef    S.E.   Wald Z Pr(>|Z|)
## Intercept              -0.4401 0.1206 -3.65  0.0003
## num_passengers         -0.0360 0.0229 -1.57  0.1166
## sales_channel=Mobile   -0.4184 0.0854 -4.90  <0.0001
## length_of_stay         -0.0070 0.0009 -7.69  <0.0001
## wants_extra_baggage     0.3974 0.0639  6.22  <0.0001
## wants_preferred_seat    0.2144 0.0569  3.77  0.0002
## wants_in_flight_meals   0.2346 0.0543  4.32  <0.0001
## flight_duration        -0.0600 0.0151 -3.98  <0.0001
```

```
cross.calib <- calibrate(lrm.bic, method = "crossvalidation", B = 10)
plot(cross.calib, las=1, xlab = "Predicted Probability")
```




```
##
## n=7174   Mean absolute error=0.008   Mean squared error=0.00014
## 0.9 Quantile of absolute error=0.015
```

VIF to check for multicollinearity

```r
vif(bic.logit)
```

```
##   sales_channelMobile        length_of_stay       flight_duration
##              1.010374              1.044478              1.039472
##   wants_extra_baggage  wants_preferred_seat wants_in_flight_meals
##              1.088395              1.133479              1.139540
```

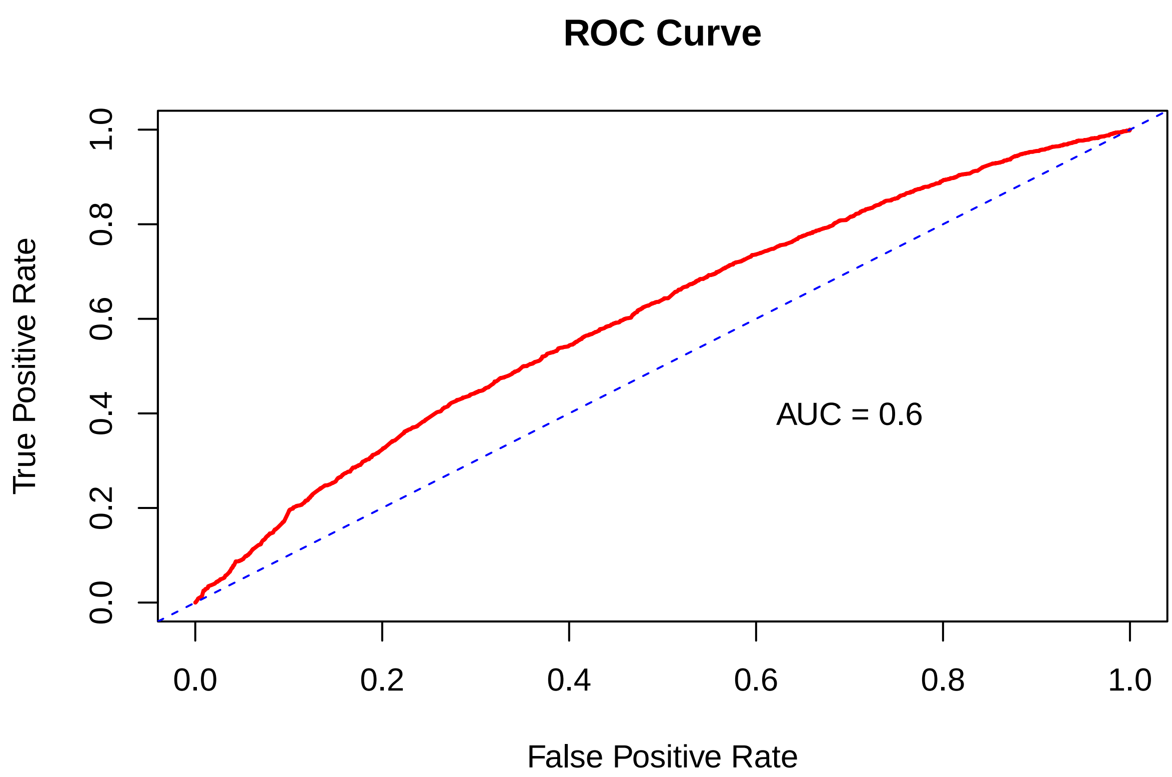
AUC and ROC Curve

```r
# Predicting probabilities using the logistic regression model
p <- predict(lrm.bic, type = "fitted")

# Generating ROC curve
roc_bic.logit <- roc(customer_booking_malaysia$booking_complete ~ p)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
# Extracting True Positive Rate (TPR) and False Positive Rate (FPR)
TPR <- roc_bic.logit$sensitivities
FPR <- 1 - roc_bic.logit$specificities

# Plotting ROC curve
plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2, col = 'red',
     xlab = "False Positive Rate", ylab = "True Positive Rate", main = "ROC Curve")
abline(a = 0, b = 1, lty = 2, col = 'blue')  # Adding diagonal reference line
text(0.7, 0.4, label = paste("AUC =", round(auc(roc_bic.logit), 2)))  # Adding AUC value as text
```




```r
# Calculating and printing the AUC
bauc_value <- auc(roc_bic.logit)
print(paste("AUC value:", round(bauc_value, 2)))
```

```
## [1] "AUC value: 0.6"
```

6.2.2 Stepwise BIC Model (Outliers removed)

Fit logistic regression with the cleaned dataset (without outliers)

```r
# Combine influential points and outliers without repetition
all_outliers <- unique(outliers_obs_bic)
```

```r
# Remove outliers and influential points from the dataset
cleaned_data_bic <- customer_booking_malaysia[-all_outliers, ]

# Fit logistic regression model with all variables
bnew.logit_model <- glm(booking_complete ~ sales_channel + trip_type + purchase_lead + length_of_stay +
                      family = binomial(link = logit),
                      data = cleaned_data_bic)
```

Perform Stepwise BIC selection with 'bnew.logit_model'

```r
newsel.var.bic <- step(bnew.logit_model, trace = 0, k = log(nrow(cleaned_data_bic)), direction = "both")
newselect_var_bic <- attr(terms(newsel.var.bic), "term.labels")
newselect_var_bic
```

```
## [1] "sales_channel"         "length_of_stay"        "wants_preferred_seat"
## [4] "wants_extra_baggage"   "wants_in_flight_meals" "flight_duration"
```

Fit Logistic Regression Model with new BIC selection

```r
new_bic.logit <- glm(booking_complete ~ sales_channel + length_of_stay + flight_duration + wants_extra_b
                    wants_preferred_seat + wants_in_flight_meals, family = binomial(link = logit),
                 data = cleaned_data_bic)
summary(new_bic.logit)
```

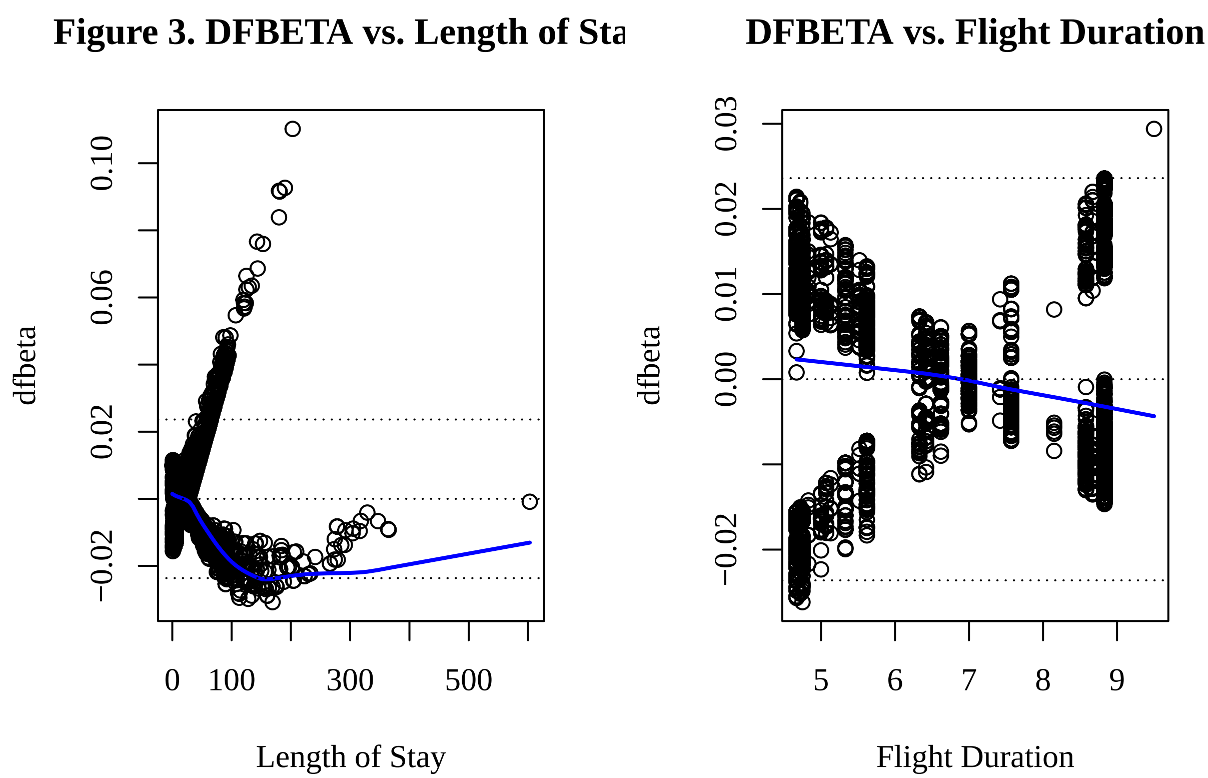
```
##
## Call:
## glm(formula = booking_complete ~ sales_channel + length_of_stay +
##     flight_duration + wants_extra_baggage + wants_preferred_seat +
##     wants_in_flight_meals, family = binomial(link = logit), data = cleaned_data_bic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2217  -0.9451  -0.7890   1.3203   2.1842
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.4392988  0.1174015  -3.742 0.000183 ***
## sales_channelMobile  -0.4332322  0.0857139  -5.054 4.32e-07 ***
## length_of_stay       -0.0094764  0.0009696  -9.773  < 2e-16 ***
## flight_duration      -0.0631684  0.0151372  -4.173 3.01e-05 ***
## wants_extra_baggage   0.4158327  0.0635408   6.544 5.98e-11 ***
## wants_preferred_seat  0.2057085  0.0571947   3.597 0.000322 ***
## wants_in_flight_meals 0.2257962  0.0544483   4.147 3.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9209.7  on 7161  degrees of freedom
## Residual deviance: 8957.7  on 7155  degrees of freedom
## AIC: 8971.7
##
## Number of Fisher Scoring iterations: 4
```

Checking influential points

```
new_df.bic <- dfbetas(new_bic.logit)
nn.bic <- nrow(cleaned_data_bic)
n.beta_cut <- 2 / sqrt(nn.bic)
influential_points_nbic <- apply(abs(new_df.bic) > n.beta_cut, 1, any)
sum(influential_points_nbic)
```

```
## [1] 1304
```

```
par(mfrow = c(1, 2))

# Find the index of the predictor variable "length_of_stay"
predictor_index <- which(names(coef(new_bic.logit)) == "length_of_stay")
df.bic_predictor <- new_df.bic[, predictor_index]

# Plot dfbeta against "length_of_stay"
par(family = 'serif')
plot(cleaned_data_bic$length_of_stay, df.bic_predictor,
     xlab='Length of Stay', ylab='dfbeta',
     main='Figure 3. DFBETA vs. Length of Stay')
lines(lowess(cleaned_data_bic$length_of_stay, df.bic_predictor),
      lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(new_df.bic)), lty='dotted')
abline(h=2/sqrt(nrow(new_df.bic)), lty='dotted')


# Find the index of the predictor variable "flight_duration"
predictor_index <- which(names(coef(new_bic.logit)) == "flight_duration")
df.bic_predictor <- new_df.bic[, predictor_index]

# Plot dfbeta against "flight_duration"
par(family = 'serif')
plot(cleaned_data_bic$flight_duration, df.bic_predictor,
     xlab='Flight Duration', ylab='dfbeta',
     main='DFBETA vs. Flight Duration')
lines(lowess(cleaned_data_bic$flight_duration, df.bic_predictor),
      lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.bic)), lty='dotted')
abline(h=2/sqrt(nrow(df.bic)), lty='dotted')
```



Checking outliers

```
ri.nbic <- rstandard(new_bic.logit)
outliers_obs_nbic <- which(ri.nbic > 2 | ri.nbic < -2)
outliers_obs_nbic
```

```
##   17  197  204  453 1021 2481 3074
##   17  197  204  453 1021 2481 3074
```

VIF to check for multicollinearity

```
vif(new_bic.logit)
```

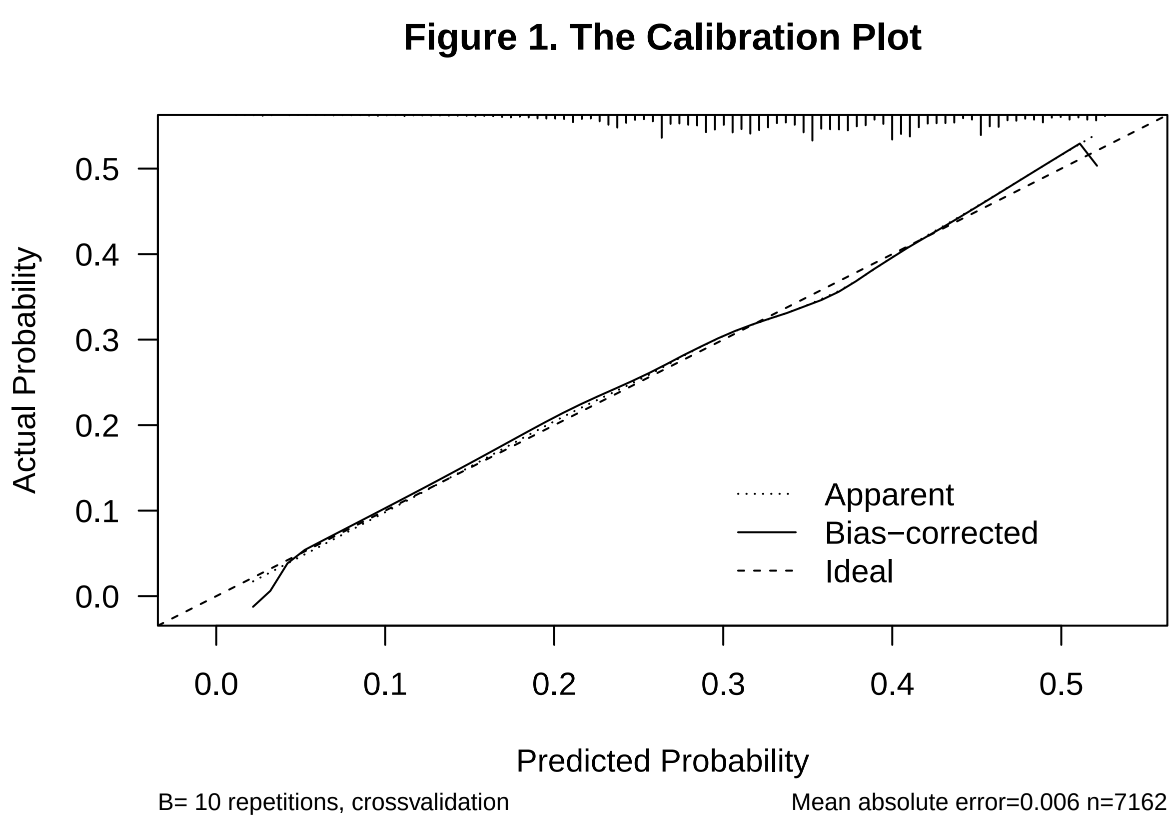
```
##   sales_channelMobile        length_of_stay       flight_duration
##              1.010675              1.049261              1.038505
##   wants_extra_baggage  wants_preferred_seat wants_in_flight_meals
##              1.091465              1.134532              1.139290
```

Cross-Validation and Calibration

```
set.seed(1007928566)
new_lrm.bic <- lrm(booking_complete ~ .,
                   data = cleaned_data_bic[, which(colnames(cleaned_data_bic) %in% c(newselect_var_bic, "
                   x = TRUE, y = TRUE, model = TRUE)
new_lrm.bic
```

```
## Logistic Regression Model
##
## lrm(formula = booking_complete ~ ., data = cleaned_data_bic[,
##     which(colnames(cleaned_data_bic) %in% c(newselect_var_bic,
##         "booking_complete"))], model = TRUE, x = TRUE, y = TRUE)
##
##                        Model Likelihood       Discrimination    Rank Discrim.
##                              Ratio Test              Indexes          Indexes
## Obs            7162    LR chi2      251.94       R2       0.048    C       0.609
##  0             4706    d.f.              6      R2(6,7162)0.034    Dxy     0.218
##  1             2456    Pr(> chi2) <0.0001    R2(6,4841.4)0.050    gamma   0.219
## max |deriv| 6e-12                                Brier    0.218    tau-a   0.098
##
##                       Coef    S.E.   Wald Z Pr(>|Z|)
## Intercept             -0.4393 0.1174 -3.74  0.0002
## sales_channel=Mobile  -0.4332 0.0857 -5.05  <0.0001
## length_of_stay        -0.0095 0.0010 -9.77  <0.0001
## wants_extra_baggage    0.4158 0.0635  6.54  <0.0001
## wants_preferred_seat   0.2057 0.0572  3.60  0.0003
## wants_in_flight_meals  0.2258 0.0544  4.15  <0.0001
## flight_duration       -0.0632 0.0151 -4.17  <0.0001
```

```
nbcross.calib <- calibrate(new_lrm.bic, method = "crossvalidation", B = 10)
plot(nbcross.calib, las=1, xlab = "Predicted Probability", main = "Figure 1. The Calibration Plot")
```




```
##
```

```
## n=7162   Mean absolute error=0.006   Mean squared error=5e-05
## 0.9 Quantile of absolute error=0.012
```

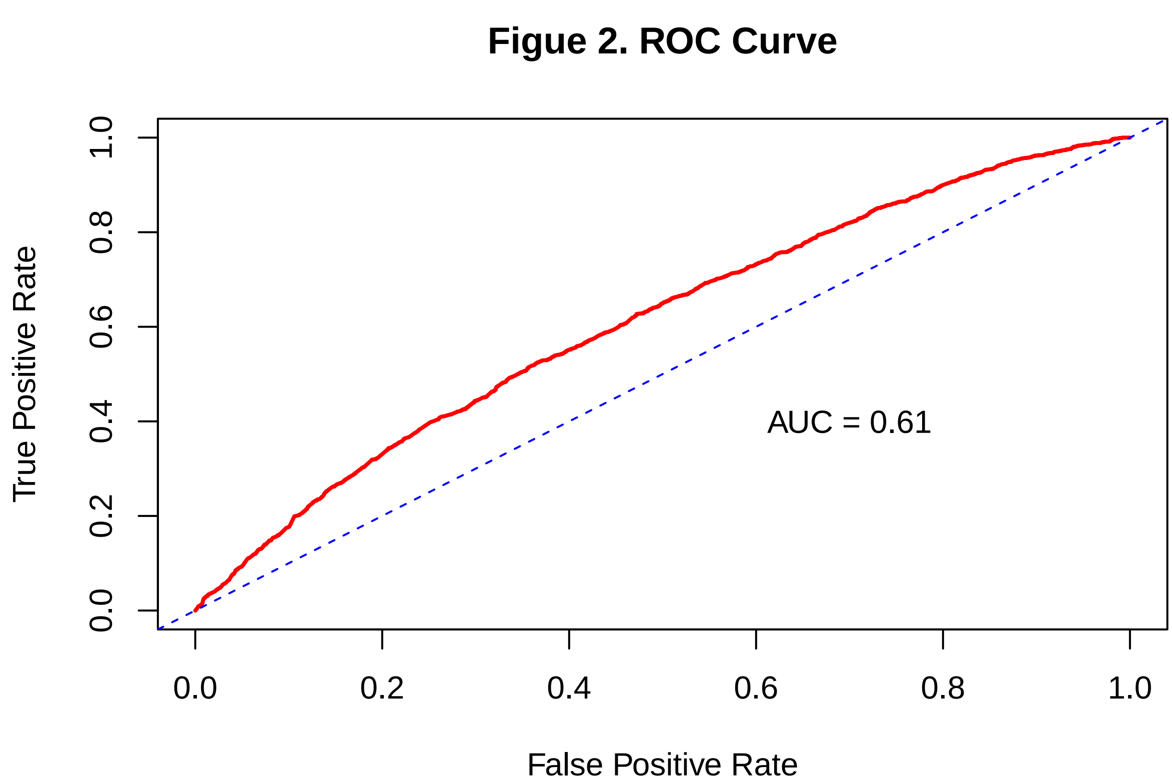
AUC and ROC Curve

```r
# Predicting probabilities using the logistic regression model
p <- predict(new_lrm.bic, type = "fitted")

# Generating ROC curve
newroc_bic.logit <- roc(cleaned_data_bic$booking_complete ~ p)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
# Extracting True Positive Rate (TPR) and False Positive Rate (FPR)
TPR <- newroc_bic.logit$sensitivities
FPR <- 1 - newroc_bic.logit$specificities

# Plotting ROC curve
plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2, col = 'red',
     xlab = "False Positive Rate", ylab = "True Positive Rate", main = "Figue 2. ROC Curve")
abline(a = 0, b = 1, lty = 2, col = 'blue')  # Adding diagonal reference line
text(0.7, 0.4, label = paste("AUC =", round(auc(newroc_bic.logit), 2)))  # Adding AUC value as text
```

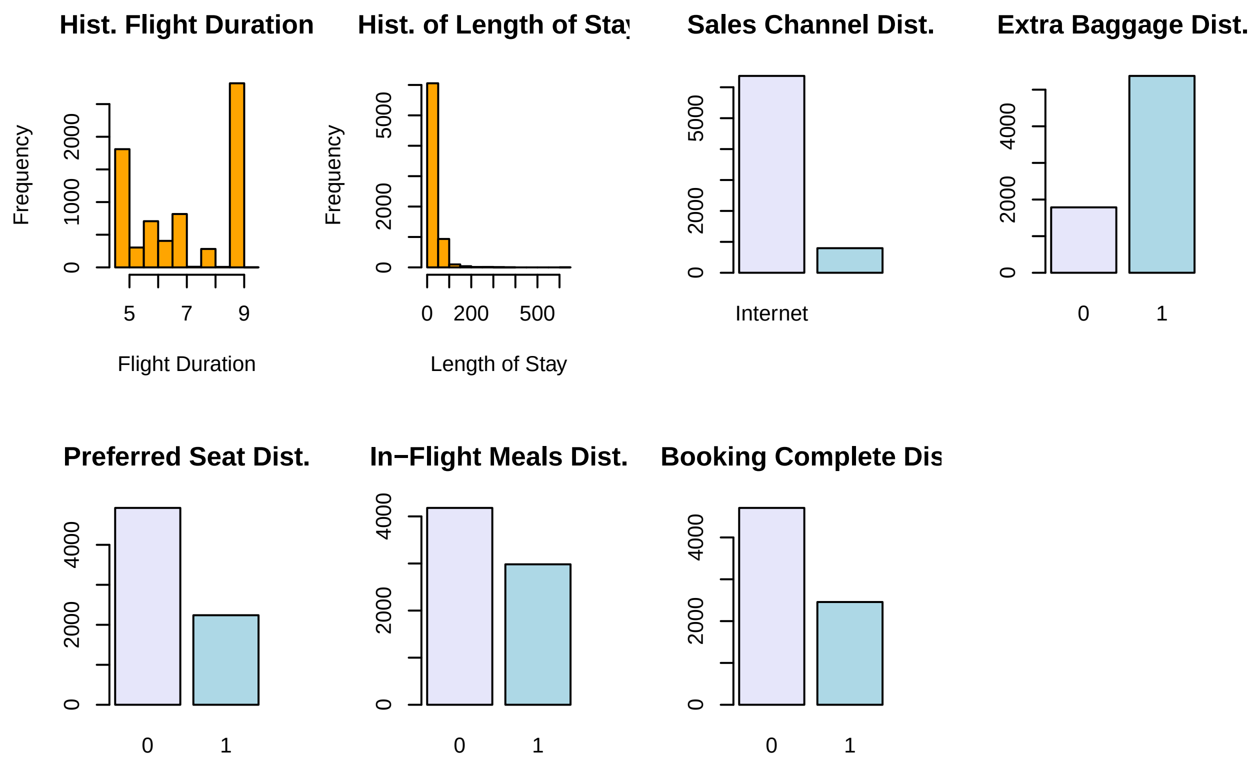


```r
# Calculating and printing the AUC
nbauc_value <- auc(newroc_bic.logit)
print(paste("AUC value:", round(nbauc_value, 2)))
```

```
## [1] "AUC value: 0.61"
```

Exploratory Data Analysis of Stepwise BIC with cleaned dataset (Chosen Model)

```r
# Set up 1x2 plotting window
par(mfrow = c(2, 4))

# Histograms for numerical variables
hist(cleaned_data_bic$flight_duration,
     main = "Hist. Flight Duration",
     xlab = "Flight Duration",
     col = "orange")

hist(cleaned_data_bic$length_of_stay,
     main = "Hist. of Length of Stay",
     xlab = "Length of Stay",
     col = "orange")

# Bar plots for categorical variables
barplot(table(cleaned_data_bic$sales_channel),
        main = "Sales Channel Dist.",
        col = c("lavender", "lightblue"))

barplot(table(cleaned_data_bic$wants_extra_baggage),
        main = "Extra Baggage Dist.",
        col = c("lavender", "lightblue"))

barplot(table(cleaned_data_bic$wants_preferred_seat),
        main = "Preferred Seat Dist.",
        col = c("lavender", "lightblue"))

barplot(table(cleaned_data_bic$wants_in_flight_meals),
        main = "In-Flight Meals Dist.",
        col = c("lavender", "lightblue"))

barplot(table(cleaned_data_bic$booking_complete),
        main = "Booking Complete Dist.",
        col = c("lavender", "lightblue"))
```




```
# Create the contingency table for categorical variable (sales_channel)
bivariate_table <- table(cleaned_data_bic$sales_channel, cleaned_data_bic$booking_complete)

# Add margins (totals) to the table
bivariate_table_with_margins <- addmargins(bivariate_table)

# Rename the last column to "Total"
colnames(bivariate_table_with_margins)[ncol(bivariate_table_with_margins)] <- "Total"

# Rename the last row to "Total"
rownames(bivariate_table_with_margins)[nrow(bivariate_table_with_margins)] <- "Total"

# Print the table with renamed margins
print(bivariate_table_with_margins)
```

```
##
##                0    1 Total
##   Internet 4125 2242  6367
##   Mobile    581  214   795
##   Total    4706 2456  7162
```

```
# Create the contingency table for categorical variables (wants_extra_baggage)
bivariate_table <- table(cleaned_data_bic$wants_extra_baggage, cleaned_data_bic$booking_complete)

# Add margins (totals) to the table
```

```r
bivariate_table_with_margins <- addmargins(bivariate_table)

# Rename the last column to "Total"
colnames(bivariate_table_with_margins)[ncol(bivariate_table_with_margins)] <- "Total"

# Rename the last row to "Total"
rownames(bivariate_table_with_margins)[nrow(bivariate_table_with_margins)] <- "Total"

# Print the table with renamed margins
print(bivariate_table_with_margins)
```

```
##
##           0    1 Total
##   0     1298  488  1786
##   1     3408 1968  5376
##   Total 4706 2456  7162
```

```r
# Create the contingency table for categorical variables (wants_preferred_seat)
bivariate_table <- table(cleaned_data_bic$wants_preferred_seat, cleaned_data_bic$booking_complete)

# Add margins (totals) to the table
bivariate_table_with_margins <- addmargins(bivariate_table)

# Rename the last column to "Total"
colnames(bivariate_table_with_margins)[ncol(bivariate_table_with_margins)] <- "Total"

# Rename the last row to "Total"
rownames(bivariate_table_with_margins)[nrow(bivariate_table_with_margins)] <- "Total"

# Print the table with renamed margins
print(bivariate_table_with_margins)
```

```
##
##           0    1 Total
##   0     3359 1564  4923
##   1     1347  892  2239
##   Total 4706 2456  7162
```

```r
# Create the contingency table for categorical variables (wants_in_flight_meals)
bivariate_table <- table(cleaned_data_bic$wants_in_flight_meals, cleaned_data_bic$booking_complete)

# Add margins (totals) to the table
bivariate_table_with_margins <- addmargins(bivariate_table)

# Rename the last column to "Total"
colnames(bivariate_table_with_margins)[ncol(bivariate_table_with_margins)] <- "Total"

# Rename the last row to "Total"
rownames(bivariate_table_with_margins)[nrow(bivariate_table_with_margins)] <- "Total"

# Print the table with renamed margins
print(bivariate_table_with_margins)
```

```
##
##              0    1 Total
##   0       2878 1302  4180
##   1       1828 1154  2982
##   Total 4706 2456  7162
```