Logistic Regression

Load the train data set on which ROSE was applied

```
train_rose <- read.csv("data/train_rose.csv")
test <- read.csv("data/test.csv")
```

Remove the column MOSTYPE.41 created during one-hot encoding because it is redundant.

```
train_rose  <- subset(train_rose, select = -c(MOSTYPE.41))
test <- subset(test, select = -c(MOSTYPE.41))
```

```
print(dim(train_rose))
```

```
## [1] 10948    94
```

Complete model

Fit a logistic regression model using all the variables

```
complete_model <- glm(
    CARAVAN ~ .,
    data = train_rose,
    family = binomial(link = "logit")
    )
```

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

```
summary(complete_model)
```

```
##
## Call:
## glm(formula = CARAVAN ~ ., family = binomial(link = "logit"),
##     data = train_rose)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.758e+00  8.404e-01  -6.851 7.32e-12 ***
## MOSTYPE.1    7.198e+00  5.382e+00   1.337 0.181094
## MOSTYPE.2    4.951e+00  3.423e+00   1.446 0.148074
## MOSTYPE.3    6.259e+00  3.020e+00   2.072 0.038238 *
## MOSTYPE.4   -6.074e+00  3.896e+00  -1.559 0.119013
## MOSTYPE.5    5.204e+00  3.095e+00   1.681 0.092676 .
## MOSTYPE.6    6.211e+00  5.376e+00   1.155 0.247947
## MOSTYPE.7    3.189e+00  3.640e+00   0.876 0.380903
## MOSTYPE.8    1.112e+01  4.121e+00   2.699 0.006964 **
## MOSTYPE.9    2.103e+00  1.876e+00   1.121 0.262317
## MOSTYPE.10   7.769e-02  5.291e+00   0.015 0.988286
## MOSTYPE.11   5.751e+00  3.209e+00   1.792 0.073113 .
## MOSTYPE.12   1.187e+01  4.307e+00   2.756 0.005849 **
## MOSTYPE.13   4.991e+00  3.169e+00   1.575 0.115265
## MOSTYPE.15  -1.290e+02  9.363e+03  -0.014 0.989011
## MOSTYPE.16  -1.389e+02  4.862e+03  -0.029 0.977200
## MOSTYPE.17  -1.477e+02  6.742e+03  -0.022 0.982520
## MOSTYPE.18  -1.382e+02  4.432e+03  -0.031 0.975133
## MOSTYPE.19  -1.429e+02  1.198e+04  -0.012 0.990484
## MOSTYPE.20   6.553e+00  4.117e+00   1.592 0.111443
## MOSTYPE.21  -1.368e+02  5.287e+03  -0.026 0.979356
```

```
## MOSTYPE.22  -2.852e+00  3.533e+00  -0.807 0.419610
## MOSTYPE.23  -7.690e+00  2.656e+00  -2.895 0.003786 **
## MOSTYPE.24  -1.134e+00  3.389e+00  -0.335 0.737841
## MOSTYPE.25  -3.325e+00  4.626e+00  -0.719 0.472276
## MOSTYPE.26  -4.396e+00  5.473e+00  -0.803 0.421891
## MOSTYPE.27  -4.655e+00  5.066e+00  -0.919 0.358133
## MOSTYPE.28  -1.414e+02  3.830e+03  -0.037 0.970548
## MOSTYPE.29  -3.838e+00  2.913e+00  -1.317 0.187692
## MOSTYPE.30   1.248e+00  3.465e+00   0.360 0.718823
## MOSTYPE.31   1.085e+00  4.399e+00   0.247 0.805251
## MOSTYPE.32   4.875e+00  4.368e+00   1.116 0.264414
## MOSTYPE.33   4.514e+00  2.047e+00   2.205 0.027431 *
## MOSTYPE.34   2.725e+00  3.159e+00   0.863 0.388385
## MOSTYPE.35  -2.069e-01  2.280e+00  -0.091 0.927701
## MOSTYPE.36   7.606e+00  2.244e+00   3.390 0.000698 ***
## MOSTYPE.37   7.529e+00  1.951e+00   3.859 0.000114 ***
## MOSTYPE.38   7.328e+00  1.777e+00   4.124 3.72e-05 ***
## MOSTYPE.39   7.084e+00  2.095e+00   3.382 0.000719 ***
## MOSTYPE.40  -1.355e+02  2.255e+03  -0.060 0.952073
## MAANTHUI    -1.875e+00  7.686e-01  -2.439 0.014720 *
## MGEMOMV     -3.843e-01  2.244e-01  -1.713 0.086796 .
## MGEMLEEF     1.186e+00  2.148e-01   5.520 3.38e-08 ***
## MGODRK      -2.828e-01  2.557e-01  -1.106 0.268804
## MGODPR       6.611e-01  1.561e-01   4.235 2.29e-05 ***
## MGODOV       5.679e-01  2.256e-01   2.517 0.011827 *
## MRELGE       7.581e-01  1.959e-01   3.870 0.000109 ***
## MRELSA       7.279e-01  3.381e-01   2.153 0.031327 *
## MFGEKIND    -6.768e-01  1.680e-01  -4.028 5.61e-05 ***
## MOPLHOOG     2.376e+00  2.678e-01   8.872  < 2e-16 ***
## MOPLMIDD     1.117e+00  1.912e-01   5.844 5.11e-09 ***
## MBERHOOG     9.540e-01  3.366e-01   2.834 0.004597 **
## MBERZELF     1.041e+00  3.750e-01   2.775 0.005523 **
## MBERBOER    -1.295e-01  3.831e-01  -0.338 0.735360
## MBERMIDD     8.913e-01  3.223e-01   2.765 0.005688 **
## MBERARBG    -2.133e-02  3.125e-01  -0.068 0.945589
## MBERARBO     4.432e-01  3.134e-01   1.414 0.157330
## MSKA        -7.654e-01  3.729e-01  -2.052 0.040139 *
## MSKB1       -3.766e-01  3.527e-01  -1.068 0.285598
## MSKB2       -1.375e-01  3.193e-01  -0.431 0.666728
## MSKC         1.190e+00  3.446e-01   3.453 0.000555 ***
## MSKD        -3.765e-01  3.338e-01  -1.128 0.259380
## MHHUUR      -2.533e-01  1.004e-01  -2.522 0.011667 *
## MAUT1        4.639e-01  2.144e-01   2.164 0.030491 *
## MAUT2       -9.575e-02  2.558e-01  -0.374 0.708188
## MZFONDS      3.369e-01  1.638e-01   2.057 0.039659 *
## MINKM30      5.020e-01  3.658e-01   1.372 0.169938
## MINK3045     7.467e-01  3.503e-01   2.132 0.033046 *
## MINK4575     4.230e-01  3.519e-01   1.202 0.229377
## MINK7512     6.947e-01  3.639e-01   1.909 0.056250 .
## MINK123M    -2.227e+00  5.191e-01  -4.290 1.79e-05 ***
## MINKGEM      8.550e-01  3.660e-01   2.336 0.019487 *
## MKOOPKLA    -4.631e-01  1.266e+00  -0.366 0.714445
## PWAPART      1.095e+00  2.690e-01   4.071 4.69e-05 ***
## PWABEDR     -4.859e-01  7.166e-01  -0.678 0.497720
```

```
## PWALAND     -3.439e+00  6.254e-01  -5.499 3.82e-08 ***
## PPERSAUT     2.078e+00  7.754e-02  26.799  < 2e-16 ***
## PBESAUT      3.010e-01  4.498e-01   0.669 0.503370
## PMOTSCO     -7.989e-01  2.334e-01  -3.422 0.000621 ***
## PVRAAUT     -3.187e+01  1.100e+03  -0.029 0.976884
## PAANHANG     1.327e+00  1.147e+00   1.157 0.247217
## PTRACTOR     7.668e-02  4.649e-01   0.165 0.868985
## PWERKT      -5.407e+01  1.145e+03  -0.047 0.962325
## PBROM       -2.433e+00  3.738e-01  -6.509 7.57e-11 ***
## PLEVEN      -1.932e-01  2.133e-01  -0.906 0.365029
## PPERSONG    -1.508e+00  1.409e+00  -1.070 0.284664
## PGEZONG      2.144e+00  9.297e-01   2.306 0.021115 *
## PWAOREG      2.770e+00  5.038e-01   5.499 3.83e-08 ***
## PBRAND       1.130e+00  1.456e-01   7.760 8.51e-15 ***
## PZEILPL     -7.506e+00  9.552e+00  -0.786 0.432007
## PPLEZIER     7.329e+00  7.759e-01   9.446  < 2e-16 ***
## PFIETS       6.029e+00  1.190e+00   5.065 4.08e-07 ***
## PINBOED     -1.680e+00  1.103e+00  -1.523 0.127797
## PBYSTAND     1.726e+00  4.259e-01   4.053 5.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15177  on 10947  degrees of freedom
## Residual deviance: 11897  on 10854  degrees of freedom
## AIC: 12085
##
## Number of Fisher Scoring iterations: 15
```

Obviusly, this model is not interpretable because it uses all the variables. We notice that many variable could be not significant and could be removed, many p-values are greater than 0.05. According to the great number of variables, we could have multicollinearity.

Let's ceck the performance of the model on the test set.

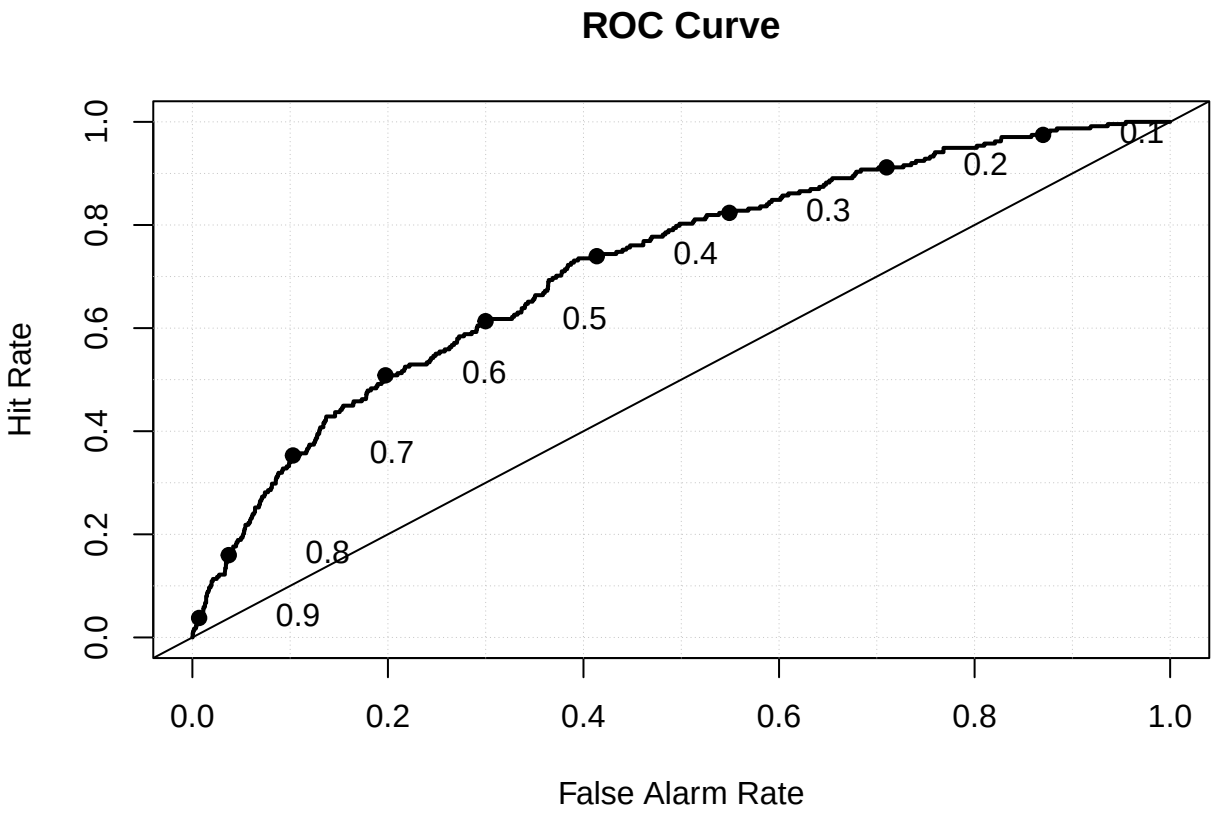
Confusion matrix on the test data set for the complete model

```
pred <- predict(complete_model, newdata = test, type = "response")
```

ROC curve for the complete model on the test set

```
roc.plot(test$CARAVAN, pred)
```

```
## Warning in roc.plot.default(test$CARAVAN, pred): Large amount of unique
## predictions used as thresholds. Consider specifying thresholds.
```



Confusion matrix for the complete model on the test set

```
pred <- ifelse(pred > 0.5, 1, 0)
confusion_matrix <-  confusionMatrix(as.factor(test$CARAVAN), as.factor(pred), positive = "1")
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2633 1129
##          1   91  147
##
##                Accuracy : 0.695
##                  95% CI : (0.6805, 0.7092)
##     No Information Rate : 0.681
##     P-Value [Acc > NIR] : 0.0295
##
##                   Kappa : 0.1044
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.11520
##             Specificity : 0.96659
##          Pos Pred Value : 0.61765
##          Neg Pred Value : 0.69989
##              Prevalence : 0.31900
##          Detection Rate : 0.03675
##    Detection Prevalence : 0.05950
```

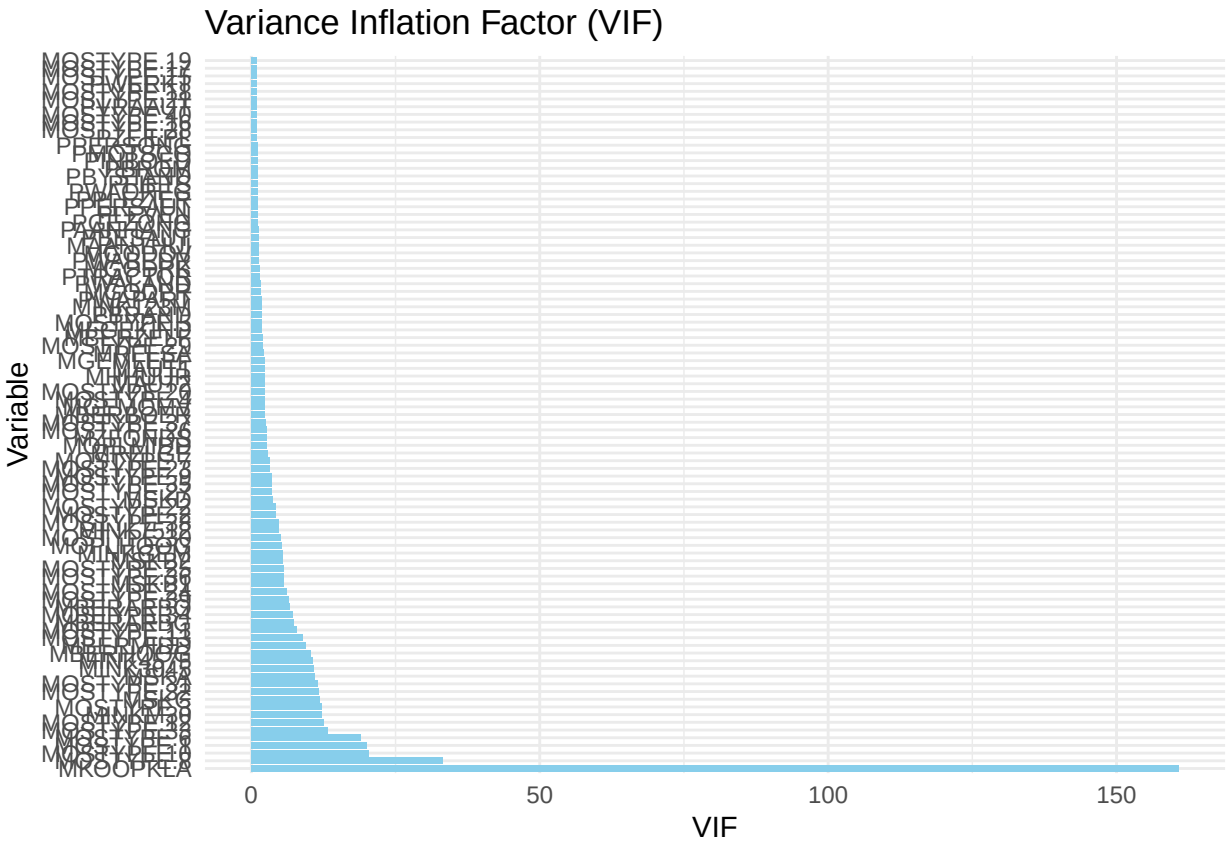
```
##        Balanced Accuracy : 0.54090
##
##         'Positive' Class : 1
##
```

Our goal is to reduce the number of variables and to make the model more interpretable.

We can use the Variance Inflation Factor (VIF) to check for multicollinearity. The VIF measures how much the variance of the estimated regression coefficients are inflated as compared to when the predictor variables are not linearly related. A VIF of 1 indicates no multicollinearity, while between 1 and 10 indicates moderate multicollinearity, and greater than 10 indicates severe multicollinearity.

```
vif_visualizer <- function(model) {
  # Calculate the VIF for the complete model
  vif_values <- vif(model)
  # Create a data frame for plotting
  vif_data <- data.frame(variable = names(vif_values), vif = vif_values)
  # Plot VIF
  ggplot(vif_data, aes(x = reorder(variable, -vif), y = vif)) +
    geom_bar(stat = "identity", fill = "skyblue") +
    coord_flip() +
    labs(title = "Variance Inflation Factor (VIF)",
         x = "Variable",
         y = "VIF") +
    theme_minimal()
}

vif_visualizer(complete_model)
```



The VIF plot shows that are variables that cause multicollinearity. We have to remove some variables to reduce the multicollinearity and to make the model model more interpretable.

Reduced model

We can start by removing the variables that have VIF greater than 10.

```
vif_values <- vif(complete_model)
small_VIF_vars <- names(vif_values[vif_values < 10])
print(small_VIF_vars)
```

```
##  [1] "MOSTYPE.2"  "MOSTYPE.4"  "MOSTYPE.5"  "MOSTYPE.7"  "MOSTYPE.9"
##  [6] "MOSTYPE.11" "MOSTYPE.13" "MOSTYPE.15" "MOSTYPE.16" "MOSTYPE.17"
## [11] "MOSTYPE.18" "MOSTYPE.19" "MOSTYPE.20" "MOSTYPE.21" "MOSTYPE.22"
## [16] "MOSTYPE.23" "MOSTYPE.24" "MOSTYPE.25" "MOSTYPE.26" "MOSTYPE.27"
## [21] "MOSTYPE.28" "MOSTYPE.29" "MOSTYPE.30" "MOSTYPE.34" "MOSTYPE.35"
## [26] "MOSTYPE.36" "MOSTYPE.37" "MOSTYPE.38" "MOSTYPE.39" "MOSTYPE.40"
## [31] "MAANTHUI"   "MGEMOMV"    "MGEMLEEF"   "MGODRK"     "MGODPR"
## [36] "MGODOV"     "MRELGE"     "MRELSA"     "MFGEKIND"   "MOPLHOOG"
## [41] "MOPLMIDD"   "MBERZELF"   "MBERBOER"   "MBERMIDD"   "MBERARBG"
## [46] "MBERARBO"   "MSKB1"      "MSKB2"      "MSKD"       "MHHUUR"
## [51] "MAUT1"      "MAUT2"      "MZFONDS"    "MINK7512"   "MINK123M"
## [56] "MINKGEM"    "PWAPART"    "PWABEDR"    "PWALAND"    "PPERSAUT"
## [61] "PBESAUT"    "PMOTSCO"    "PVRAAUT"    "PAANHANG"   "PTRACTOR"
## [66] "PWERKT"     "PBROM"      "PLEVEN"     "PPERSONG"   "PGEZONG"
## [71] "PWAOREG"    "PBRAND"     "PZEILPL"    "PPLEZIER"   "PFIETS"
## [76] "PINBOED"    "PBYSTAND"
```

Let's drop the variables that are likely to be not different from zero and with a very small effect on the response variable. Select the variables that have a p-value less than 0.10 and a coefficient greater than 0.01. This seems to be a good compromise between the number of variables and the model's performance.

```
significant_vars <- names(which(summary(complete_model)$coefficients[,4] < 0.10 & summary(complete_model
print(significant_vars)
```

```
##  [1] "MOSTYPE.3"  "MOSTYPE.5"  "MOSTYPE.8"  "MOSTYPE.11" "MOSTYPE.12"
##  [6] "MOSTYPE.33" "MOSTYPE.36" "MOSTYPE.37" "MOSTYPE.38" "MOSTYPE.39"
## [11] "MGEMLEEF"   "MGODPR"     "MGODOV"     "MRELGE"     "MRELSA"
## [16] "MOPLHOOG"   "MOPLMIDD"   "MBERHOOG"   "MBERZELF"   "MBERMIDD"
## [21] "MSKC"       "MAUT1"      "MZFONDS"    "MINK3045"   "MINK7512"
## [26] "MINKGEM"    "PWAPART"    "PPERSAUT"   "PGEZONG"    "PWAOREG"
## [31] "PBRAND"     "PPLEZIER"   "PFIETS"     "PBYSTAND"
```

```
#Evaluate the intersection of the two sets of variables
vars <- intersect(small_VIF_vars, significant_vars)
print(vars)
```

```
##  [1] "MOSTYPE.5"  "MOSTYPE.11" "MOSTYPE.36" "MOSTYPE.37" "MOSTYPE.38"
##  [6] "MOSTYPE.39" "MGEMLEEF"   "MGODPR"     "MGODOV"     "MRELGE"
## [11] "MRELSA"     "MOPLHOOG"   "MOPLMIDD"   "MBERZELF"   "MBERMIDD"
## [16] "MAUT1"      "MZFONDS"    "MINK7512"   "MINKGEM"    "PWAPART"
## [21] "PPERSAUT"   "PGEZONG"    "PWAOREG"    "PBRAND"     "PPLEZIER"
## [26] "PFIETS"     "PBYSTAND"
```

Fit a logistic regression model using the selected variables

```
vars <- c(vars, "CARAVAN")
reduced_model <- glm(
```

```
    CARAVAN ~ .,
    data = train_rose[, vars],
    family = "binomial"
    )
```

```
summary(reduced_model)
```

```
##
## Call:
## glm(formula = CARAVAN ~ ., family = "binomial", data = train_rose[,
##     vars])
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.970447   0.244575 -20.323  < 2e-16 ***
## MOSTYPE.5    0.987707   2.306797   0.428 0.668526
## MOSTYPE.11   1.393457   1.176395   1.185 0.236209
## MOSTYPE.36   5.598584   0.972320   5.758 8.51e-09 ***
## MOSTYPE.37   3.030771   1.241305   2.442 0.014622 *
## MOSTYPE.38   5.476546   0.850342   6.440 1.19e-10 ***
## MOSTYPE.39   3.115341   0.838647   3.715 0.000203 ***
## MGEMLEEF     0.813093   0.159003   5.114 3.16e-07 ***
## MGODPR       0.744574   0.128974   5.773 7.79e-09 ***
## MGODOV       0.376911   0.204539   1.843 0.065368 .
## MRELGE       1.005706   0.149521   6.726 1.74e-11 ***
## MRELSA      -0.008608   0.283174  -0.030 0.975750
## MOPLHOOG     1.526649   0.154993   9.850  < 2e-16 ***
## MOPLMIDD     0.483275   0.132465   3.648 0.000264 ***
## MBERZELF     0.147340   0.270875   0.544 0.586482
## MBERMIDD     0.908141   0.117728   7.714 1.22e-14 ***
## MAUT1        0.999002   0.148825   6.713 1.91e-11 ***
## MZFONDS      0.262118   0.127751   2.052 0.040191 *
## MINK7512     0.549745   0.217425   2.528 0.011457 *
## MINKGEM      0.710885   0.238285   2.983 0.002851 **
## PWAPART      1.363284   0.239302   5.697 1.22e-08 ***
## PPERSAUT     2.088404   0.073166  28.544  < 2e-16 ***
## PGEZONG      1.169344   0.819366   1.427 0.153542
## PWAOREG      2.707559   0.424351   6.380 1.77e-10 ***
## PBRAND       0.913632   0.126524   7.221 5.16e-13 ***
## PPLEZIER     6.177039   0.679209   9.094  < 2e-16 ***
## PFIETS       7.393364   1.115308   6.629 3.38e-11 ***
## PBYSTAND     2.194998   0.405270   5.416 6.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15177  on 10947  degrees of freedom
## Residual deviance: 12608  on 10920  degrees of freedom
## AIC: 12664
##
## Number of Fisher Scoring iterations: 5
```

Confusion matrix on the test data set for the reduced model

```r
pred <- predict(reduced_model, newdata = test, type = "response")
pred <- ifelse(pred > 0.5, 1, 0)
confusion_matrix <- confusionMatrix(as.factor(test$CARAVAN), as.factor(pred), positive = "1")
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2567 1195
##          1   86  152
##
##                Accuracy : 0.6798
##                  95% CI : (0.665, 0.6942)
##     No Information Rate : 0.6632
##     P-Value [Acc > NIR] : 0.01395
##
##                   Kappa : 0.1009
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.1128
##             Specificity : 0.9676
##          Pos Pred Value : 0.6387
##          Neg Pred Value : 0.6823
##              Prevalence : 0.3367
##          Detection Rate : 0.0380
##    Detection Prevalence : 0.0595
##       Balanced Accuracy : 0.5402
##
##        'Positive' Class : 1
##
```

We removed 61 variables from the complete model, and the model performance metrics did not change significantly.

Stepwise regression

Considering the variables of the reduced model, perform a stepwise regression in order to select the best subset of variables. We use the Bayesian Information Criterion (BIC) to select the best model because we have a large number of variables and the BIC penalizes the number of variables in the model more than Akaike Information Criterion.

```r
# Stepwise regression
stepwise_model <- step(
    glm(
        CARAVAN ~ .,
        data = train_rose[, vars],
        family = "binomial",
    ),
    k = log(nrow(train_rose)),
    direction = "both"
)
```

```r
summary(stepwise_model)
```

```
##
## Call:
## glm(formula = CARAVAN ~ MOSTYPE.36 + MOSTYPE.38 + MOSTYPE.39 +
##     MGEMLEEF + MGODPR + MRELGE + MOPLHOOG + MOPLMIDD + MBERMIDD +
##     MAUT1 + MINKGEM + PWAPART + PPERSAUT + PWAOREG + PBRAND +
##     PPLEZIER + PFIETS + PBYSTAND, family = "binomial", data = train_rose[,
##     vars])
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.64807    0.16589 -28.018  < 2e-16 ***
## MOSTYPE.36   5.39975    0.96818   5.577 2.44e-08 ***
## MOSTYPE.38   5.53579    0.83983   6.592 4.35e-11 ***
## MOSTYPE.39   2.97858    0.83748   3.557 0.000376 ***
## MGEMLEEF     0.83130    0.14508   5.730 1.01e-08 ***
## MGODPR       0.66973    0.12128   5.522 3.35e-08 ***
## MRELGE       0.92614    0.12514   7.401 1.35e-13 ***
## MOPLHOOG     1.34621    0.13403  10.044  < 2e-16 ***
## MOPLMIDD     0.39078    0.12372   3.158 0.001586 **
## MBERMIDD     0.94975    0.11365   8.357  < 2e-16 ***
## MAUT1        0.94600    0.14669   6.449 1.13e-10 ***
## MINKGEM      1.07557    0.18766   5.732 9.95e-09 ***
## PWAPART      1.41908    0.23688   5.991 2.09e-09 ***
## PPERSAUT     2.07929    0.07267  28.615  < 2e-16 ***
## PWAOREG      2.67156    0.42184   6.333 2.40e-10 ***
## PBRAND       0.89802    0.12476   7.198 6.11e-13 ***
## PPLEZIER     6.25528    0.68387   9.147  < 2e-16 ***
## PFIETS       7.38016    1.10747   6.664 2.67e-11 ***
## PBYSTAND     2.25831    0.40124   5.628 1.82e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15177  on 10947  degrees of freedom
## Residual deviance: 12634  on 10929  degrees of freedom
## AIC: 12672
##
## Number of Fisher Scoring iterations: 5
```

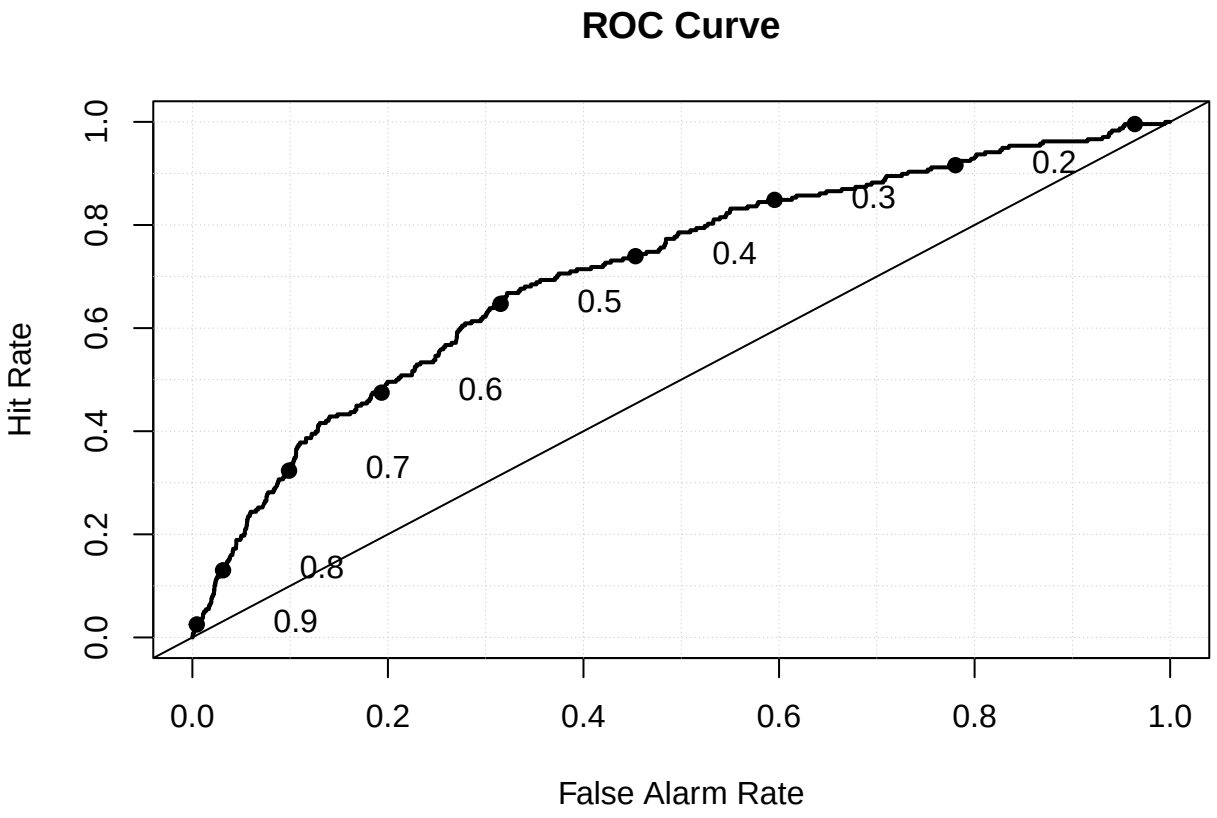
Confusion matrix on the test data set for the stepwise model

```r
pred <- predict(stepwise_model, newdata = test, type = "response")
```

ROC curve for the stepwise model

```r
roc.plot(test$CARAVAN, pred)
```

```
## Warning in roc.plot.default(test$CARAVAN, pred): Large amount of unique
## predictions used as thresholds. Consider specifying thresholds.
```



Confusion matrix for the stepwise model

```
pred <- ifelse(pred > 0.5, 1, 0)
confusion_matrix <- confusionMatrix(as.factor(test$CARAVAN), as.factor(pred), positive = "1")
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2576 1186
##          1   84  154
##
##                Accuracy : 0.6825
##                  95% CI : (0.6678, 0.6969)
##     No Information Rate : 0.665
##     P-Value [Acc > NIR] : 0.009728
##
##                   Kappa : 0.1047
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.1149
##             Specificity : 0.9684
##          Pos Pred Value : 0.6471
##          Neg Pred Value : 0.6847
##              Prevalence : 0.3350
##          Detection Rate : 0.0385
##    Detection Prevalence : 0.0595
```

```
##       Balanced Accuracy : 0.5417
##
##        'Positive' Class : 1
##
```

Bayesian Information Criterion Summary

Calculate the BIC for the complete, reduced models, and the stepwise model

```
BIC(complete_model, reduced_model, stepwise_model)
```

```
##                df      BIC
## complete_model 94 12771.04
## reduced_model  28 12868.38
## stepwise_model 19 12810.30
```

Interpretation and Conclusion

We reached the goal to reduce the number of variables without worsening the model performance.

The stepwise model, with 28 variables, reaches the best balanced accuracy on the test set, 0.5430. The factors that result to influence more the target variable are:

```
-PPLEZIER Contribution boat policies, if PPLEZIER increases by 1,
the log odds of the target varaible to be 1 increases by 6.25/9 = 0.69

-PFIETS Contribution bicycle policies, if PFIETS increases by 1, the log odds
of the target varaible to be 1 increases by 7.38/9 = 0.82

-PPERSAUT Contribution car policies, if the variable increases by 1, the
log odds of the target variable to be 1 increases by 2.07/9 = 0.23.

-MOSTYPE.36 Couples with teens, Married with children,
if this variable is 1 (True), the log odds of the
target variable to be 1 increases by 5.39.

-MOSTYPE.38 Traditional families,  if this variable
is 1 (True), the log odds of the target variable
to be 1 increases by 5.5
```

In particular, these variables have a positive effect on the target variable, meaning that the higher the value of the variable, the higher the probability of the target variable to be 1. The interpretation of the coefficient is note very straightforward because of the encoding and the normalization process applied to the data.