COMPAS Dataset Novel Analysis

Dylan Doby

December 20, 2024

```r
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

Load and Explore the Data

```
# Load the COMPAS dataset
compas_data <- read.csv("~/Downloads/compas-scores-two-years.csv")

# Inspect the dataset
str(compas_data)
```

```
## 'data.frame':    7214 obs. of  53 variables:
##  $ id                     : int  1 3 4 5 6 7 8 9 10 13 ...
##  $ name                   : chr  "miguel hernandez" "kevon dixon" "ed philo" "marcu brown" ...
##  $ first                  : chr  "miguel" "kevon" "ed" "marcu" ...
##  $ last                   : chr  "hernandez" "dixon" "philo" "brown" ...
##  $ compas_screening_date  : chr  "2013-08-14" "2013-01-27" "2013-04-14" "2013-01-13" ...
##  $ sex                    : chr  "Male" "Male" "Male" "Male" ...
##  $ dob                    : chr  "1947-04-18" "1982-01-22" "1991-05-14" "1993-01-21" ...
##  $ age                    : int  69 34 24 23 43 44 41 43 39 21 ...
##  $ age_cat                : chr  "Greater than 45" "25 - 45" "Less than 25" "Less than 25" ...
##  $ race                   : chr  "Other" "African-American" "African-American" "African-American" ..
##  $ juv_fel_count          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ decile_score           : int  1 3 4 8 1 1 6 4 1 3 ...
##  $ juv_misd_count         : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ juv_other_count        : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ priors_count           : int  0 0 4 1 2 0 14 3 0 1 ...
##  $ days_b_screening_arrest: int  -1 -1 -1 NA NA 0 -1 -1 -1 428 ...
##  $ c_jail_in              : chr  "2013-08-13 06:03:42" "2013-01-26 03:45:27" "2013-04-13 04:58:34" "
##  $ c_jail_out             : chr  "2013-08-14 05:41:20" "2013-02-05 05:36:53" "2013-04-14 07:02:04" "
##  $ c_case_number          : chr  "13011352CF10A" "13001275CF10A" "13005330CF10A" "13000570CF10A" ...
##  $ c_offense_date         : chr  "2013-08-13" "2013-01-26" "2013-04-13" "2013-01-12" ...
##  $ c_arrest_date          : chr  "" "" "" "" ...
##  $ c_days_from_compas     : int  1 1 1 1 76 0 1 1 1 308 ...
##  $ c_charge_degree        : chr  "F" "F" "F" "F" ...
##  $ c_charge_desc          : chr  "Aggravated Assault w/Firearm" "Felony Battery w/Prior Convict" "Pos
##  $ is_recid               : int  0 1 1 0 0 0 1 0 0 1 ...
##  $ r_case_number          : chr  "" "13009779CF10A" "13011511MM10A" "" ...
##  $ r_charge_degree        : chr  "" "(F3)" "(M1)" "" ...
##  $ r_days_from_arrest     : int  NA NA 0 NA NA NA 0 NA NA 0 ...
##  $ r_offense_date         : chr  "" "2013-07-05" "2013-06-16" "" ...
```

```
##  $ r_charge_desc         : chr  "" "Felony Battery (Dom Strang)" "Driving Under The Influence" "" .
##  $ r_jail_in             : chr  "" "" "2013-06-16" "" ...
##  $ r_jail_out            : chr  "" "" "2013-06-16" "" ...
##  $ violent_recid         : logi  NA NA NA NA NA NA ...
##  $ is_violent_recid      : int  0 1 0 0 0 0 0 0 0 1 ...
##  $ vr_case_number        : chr  "" "13009779CF10A" "" "" ...
##  $ vr_charge_degree      : chr  "" "(F3)" "" "" ...
##  $ vr_offense_date       : chr  "" "2013-07-05" "" "" ...
##  $ vr_charge_desc        : chr  "" "Felony Battery (Dom Strang)" "" "" ...
##  $ type_of_assessment    : chr  "Risk of Recidivism" "Risk of Recidivism" "Risk of Recidivism" "Risk
##  $ decile_score.1        : int  1 3 4 8 1 1 6 4 1 3 ...
##  $ score_text            : chr  "Low" "Low" "Low" "High" ...
##  $ screening_date        : chr  "2013-08-14" "2013-01-27" "2013-04-14" "2013-01-13" ...
##  $ v_type_of_assessment  : chr  "Risk of Violence" "Risk of Violence" "Risk of Violence" "Risk of Vi
##  $ v_decile_score        : int  1 1 3 6 1 1 2 3 1 5 ...
##  $ v_score_text          : chr  "Low" "Low" "Low" "Medium" ...
##  $ v_screening_date      : chr  "2013-08-14" "2013-01-27" "2013-04-14" "2013-01-13" ...
##  $ in_custody            : chr  "2014-07-07" "2013-01-26" "2013-06-16" "" ...
##  $ out_custody           : chr  "2014-07-14" "2013-02-05" "2013-06-16" "" ...
##  $ priors_count.1        : int  0 0 4 1 2 0 14 3 0 1 ...
##  $ start                 : int  0 9 0 0 0 1 5 0 2 0 ...
##  $ end                   : int  327 159 63 1174 1102 853 40 265 747 428 ...
##  $ event                 : int  0 1 0 0 0 0 1 0 0 1 ...
##  $ two_year_recid        : int  0 1 1 0 0 0 1 0 0 1 ...
```

```
summary(compas_data)
```

```
##        id            name              first               last
##  Min.   :    1   Length:7214        Length:7214        Length:7214
##  1st Qu.: 2735   Class :character   Class :character   Class :character
##  Median : 5510   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 5501
##  3rd Qu.: 8246
##  Max.   :11001
##
##  compas_screening_date     sex                dob                 age
##  Length:7214           Length:7214        Length:7214        Min.   :18.00
##  Class :character      Class :character   Class :character   1st Qu.:25.00
##  Mode  :character      Mode  :character   Mode  :character   Median :31.00
##                                                              Mean   :34.82
##                                                              3rd Qu.:42.00
##                                                              Max.   :96.00
##
##    age_cat              race           juv_fel_count       decile_score
##  Length:7214        Length:7214        Min.   : 0.00000   Min.   : 1.00
##  Class :character   Class :character   1st Qu.: 0.00000   1st Qu.: 2.00
##  Mode  :character   Mode  :character   Median : 0.00000   Median : 4.00
##                                        Mean   : 0.06723   Mean   : 4.51
##                                        3rd Qu.: 0.00000   3rd Qu.: 7.00
##                                        Max.   :20.00000   Max.   :10.00
##
##  juv_misd_count     juv_other_count    priors_count    days_b_screening_arrest
##  Min.   : 0.00000   Min.   : 0.0000   Min.   : 0.000   Min.   :-414.000
##  1st Qu.: 0.00000   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.:  -1.000
```

```
##  Median : 0.00000   Median : 0.0000   Median : 2.000   Median :  -1.000
##  Mean   : 0.09093   Mean   : 0.1094   Mean   : 3.472   Mean   :   3.305
##  3rd Qu.: 0.00000   3rd Qu.: 0.0000   3rd Qu.: 5.000   3rd Qu.:   0.000
##  Max.   :13.00000   Max.   :17.0000   Max.   :38.000   Max.   :1057.000
##                                                        NA's   :307
##   c_jail_in          c_jail_out        c_case_number      c_offense_date
##  Length:7214        Length:7214        Length:7214        Length:7214
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  c_arrest_date      c_days_from_compas c_charge_degree    c_charge_desc
##  Length:7214        Min.   :   0.00    Length:7214        Length:7214
##  Class :character   1st Qu.:   1.00    Class :character   Class :character
##  Mode  :character   Median :   1.00    Mode  :character   Mode  :character
##                     Mean   :  57.73
##                     3rd Qu.:   2.00
##                     Max.   :9485.00
##                     NA's   :22
##     is_recid      r_case_number      r_charge_degree    r_days_from_arrest
##  Min.   :0.0000   Length:7214        Length:7214        Min.   : -1.00
##  1st Qu.:0.0000   Class :character   Class :character   1st Qu.:  0.00
##  Median :0.0000   Mode  :character   Mode  :character   Median :  0.00
##  Mean   :0.4811                                         Mean   : 20.27
##  3rd Qu.:1.0000                                         3rd Qu.:  1.00
##  Max.   :1.0000                                         Max.   :993.00
##                                                         NA's   :4898
##  r_offense_date     r_charge_desc       r_jail_in          r_jail_out
##  Length:7214        Length:7214        Length:7214        Length:7214
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  violent_recid  is_violent_recid vr_case_number     vr_charge_degree
##  Mode:logical   Min.   :0.0000   Length:7214        Length:7214
##  NA's:7214      1st Qu.:0.0000   Class :character   Class :character
##                 Median :0.0000   Mode  :character   Mode  :character
##                 Mean   :0.1135
##                 3rd Qu.:0.0000
##                 Max.   :1.0000
##
##  vr_offense_date    vr_charge_desc     type_of_assessment decile_score.1
##  Length:7214        Length:7214        Length:7214        Min.   : 1.00
##  Class :character   Class :character   Class :character   1st Qu.: 2.00
##  Mode  :character   Mode  :character   Mode  :character   Median : 4.00
##                                                           Mean   : 4.51
##                                                           3rd Qu.: 7.00
##                                                           Max.   :10.00
##
##   score_text         screening_date     v_type_of_assessment v_decile_score
```

```
##  Length:7214        Length:7214        Length:7214         Min.   : 1.000
##  Class :character   Class :character   Class :character    1st Qu.: 1.000
##  Mode  :character   Mode  :character   Mode  :character    Median : 3.000
##                                                            Mean   : 3.692
##                                                            3rd Qu.: 5.000
##                                                            Max.   :10.000
##
##  v_score_text       v_screening_date    in_custody         out_custody
##  Length:7214        Length:7214        Length:7214        Length:7214
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  priors_count.1       start             end              event
##  Min.   : 0.000   Min.   :  0.00   Min.   :   0.0   Min.   :0.0000
##  1st Qu.: 0.000   1st Qu.:  0.00   1st Qu.: 148.2   1st Qu.:0.0000
##  Median : 2.000   Median :  0.00   Median : 530.5   Median :0.0000
##  Mean   : 3.472   Mean   : 11.47   Mean   : 553.4   Mean   :0.3829
##  3rd Qu.: 5.000   3rd Qu.:  1.00   3rd Qu.: 914.0   3rd Qu.:1.0000
##  Max.   :38.000   Max.   :937.00   Max.   :1186.0   Max.   :1.0000
##
##  two_year_recid
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4507
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

```
# Filter for relevant variables
compas_filtered <- compas_data %>%
  dplyr::select(
    age,
    sex,
    race,
    priors_count,
    decile_score,
    is_recid
  ) %>%
  mutate(is_recid = as.factor(is_recid))

# Check for missing data
sum(is.na(compas_filtered))
```

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

```
# Drop rows with missing data
compas_filtered <- na.omit(compas_filtered)
```

Logistic Regression: Simple Model

```r
# Fit a logistic regression model to predict recidivism
simple_mod <- glm(is_recid ~ age + sex + race + priors_count + decile_score,
                  data = compas_filtered,
                  family = binomial)

# Summary of the model
summary(simple_mod)
```

```
##
## Call:
## glm(formula = is_recid ~ age + sex + race + priors_count + decile_score,
##     family = binomial, data = compas_filtered)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9526  -0.9967  -0.5589   1.0405   2.3273
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.288115   0.131296  -2.194   0.0282 *
## age                  -0.031952   0.002694 -11.860  < 2e-16 ***
## sexMale               0.369693   0.066192   5.585 2.33e-08 ***
## raceAsian            -0.166767   0.398882  -0.418   0.6759
## raceCaucasian        -0.005297   0.059086  -0.090   0.9286
## raceHispanic         -0.175832   0.096185  -1.828   0.0675 .
## raceNative American   0.080731   0.539879   0.150   0.8811
## raceOther            -0.080941   0.120440  -0.672   0.5016
## priors_count          0.115929   0.007759  14.941  < 2e-16 ***
## decile_score          0.145308   0.011675  12.446  < 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: 9990.5  on 7213  degrees of freedom
## Residual deviance: 8696.6  on 7204  degrees of freedom
## AIC: 8716.6
##
## Number of Fisher Scoring iterations: 4
```

```r
# Check for multicollinearity
vif(simple_mod)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## age          1.367589  1        1.169440
## sex          1.013284  1        1.006620
## race         1.111453  5        1.010623
## priors_count 1.367598  1        1.169443
## decile_score 1.475638  1        1.214758
```

Adjusted Model with Interaction Terms

```r
# Fit a model with interaction terms
interaction_mod <- glm(is_recid ~ (age + sex + race + priors_count + decile_score)^2,
                       data = compas_filtered,
                       family = binomial)

# Summary of the model
summary(interaction_mod)
```

```
##
## Call:
## glm(formula = is_recid ~ (age + sex + race + priors_count + decile_score)^2,
##     family = binomial, data = compas_filtered)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6814  -0.9659  -0.5649   1.0158   2.5067
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -9.447e-01  3.578e-01  -2.640  0.00828 **
## age                              -2.346e-02  8.729e-03  -2.687  0.00721 **
## sexMale                           7.402e-01  3.314e-01   2.233  0.02553 *
## raceAsian                        -1.175e+01  2.896e+01  -0.406  0.68481
## raceCaucasian                    -4.932e-01  2.926e-01  -1.686  0.09181 .
## raceHispanic                      7.821e-01  4.802e-01   1.629  0.10335
## raceNative American               3.503e+02  4.941e+03   0.071  0.94347
## raceOther                        -1.718e-01  6.753e-01  -0.254  0.79923
## priors_count                      3.475e-01  3.973e-02   8.745  < 2e-16 ***
## decile_score                      2.050e-01  4.486e-02   4.569 4.89e-06 ***
## age:sexMale                      -4.252e-03  7.423e-03  -0.573  0.56683
## age:raceAsian                     7.710e-02  8.564e-02   0.900  0.36794
## age:raceCaucasian                 1.585e-02  6.065e-03   2.614  0.00895 **
## age:raceHispanic                 -6.440e-03  1.046e-02  -0.616  0.53821
## age:raceNative American          -1.020e+01  1.240e+02  -0.082  0.93444
## age:raceOther                    -4.745e-04  1.462e-02  -0.032  0.97410
## age:priors_count                 -2.288e-03  5.811e-04  -3.937 8.26e-05 ***
## age:decile_score                 -1.307e-03  1.047e-03  -1.249  0.21182
## sexMale:raceAsian                 4.132e+00  2.810e+01   0.147  0.88312
## sexMale:raceCaucasian            -2.662e-01  1.501e-01  -1.773  0.07624 .
## sexMale:raceHispanic             -4.784e-01  2.643e-01  -1.810  0.07027 .
## sexMale:raceNative American      -8.666e+01  2.561e+03  -0.034  0.97300
## sexMale:raceOther                -1.614e-02  3.500e-01  -0.046  0.96322
## sexMale:priors_count             -6.108e-02  2.444e-02  -2.499  0.01245 *
## sexMale:decile_score              1.251e-02  3.257e-02   0.384  0.70089
## raceAsian:priors_count            2.211e+00  1.309e+00   1.690  0.09105 .
## raceCaucasian:priors_count        3.188e-04  1.829e-02   0.017  0.98609
## raceHispanic:priors_count        -2.002e-03  3.089e-02  -0.065  0.94832
## raceNative American:priors_count  1.418e+01  1.740e+02   0.082  0.93503
## raceOther:priors_count            1.039e-01  5.523e-02   1.880  0.06008 .
## raceAsian:decile_score            8.986e-01  6.078e-01   1.478  0.13932
## raceCaucasian:decile_score        3.304e-02  2.700e-02   1.224  0.22094
```

```
## raceHispanic:decile_score        -9.009e-02  4.458e-02  -2.021  0.04331 *
## raceNative American:decile_score -6.271e+00  1.430e+02  -0.044  0.96502
## raceOther:decile_score           -1.759e-02  6.663e-02  -0.264  0.79172
## priors_count:decile_score        -1.306e-02  2.749e-03  -4.753 2.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9990.5  on 7213  degrees of freedom
## Residual deviance: 8568.5  on 7178  degrees of freedom
## AIC: 8640.5
##
## Number of Fisher Scoring iterations: 15
```

```
# Compare models using AIC
AIC(simple_mod, interaction_mod)
```

```
##                 df      AIC
## simple_mod      10 8716.572
## interaction_mod 36 8640.538
```

Hierarchical Model Selection Process

```
hierarchical_backward_elimination <- function(model, data) {
  terms <- attr(model$terms, "term.labels")  # Extract initial terms
  best_aic <- AIC(model)  # Track the best AIC
  best_model <- model  # Store the best model

  while (length(terms) > 1) {
    # Test removing each term
    candidate_models <- lapply(terms, function(term) {
      reduced_terms <- terms[!terms %in% term]

      # Skip if removal violates the hierarchical principle
      if (any(grepl(paste0(term, ":"), paste(reduced_terms, collapse = " + ")))) {
        return(NULL)
      }

      # Fit reduced model
      reduced_model <- glm(as.formula(paste("is_recid ~", paste(reduced_terms, collapse = " + "))),
                          data = data,
                          family = binomial)
      list(term = term, model = reduced_model, aic = AIC(reduced_model))
    })

    # Filter valid models
    candidate_models <- Filter(Negate(is.null), candidate_models)
    if (length(candidate_models) == 0) break

    # Find the model with the lowest AIC
```

```r
    best_candidate <- candidate_models[[which.min(sapply(candidate_models, function(x) x$aic))]]

    # Only update if AIC improves
    if (best_candidate$aic < best_aic) {
      cat("Removing term:", best_candidate$term, "with AIC:", best_candidate$aic, "\n")
      best_aic <- best_candidate$aic
      best_model <- best_candidate$model
      terms <- attr(best_model$terms, "term.labels")  # Update terms
    } else {
      break
    }
  }

  return(best_model)
}

# Apply the hierarchical backward elimination process
final_model <- hierarchical_backward_elimination(interaction_mod, compas_filtered)
```

```
## Removing term: sex:decile_score with AIC: 8638.685
## Removing term: age:sex with AIC: 8637.435
## Removing term: age:decile_score with AIC: 8636.955
## Removing term: race:decile_score with AIC: 8636.938
```

```r
summary(final_model)
```

```
##
## Call:
## glm(formula = as.formula(paste("is_recid ~", paste(reduced_terms,
##     collapse = " + "))), family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6854  -0.9738  -0.5470   1.0151   2.3372
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -7.409e-01  1.882e-01  -3.937 8.26e-05 ***
## age                         -3.179e-02  4.460e-03  -7.129 1.01e-12 ***
## sexMale                      6.684e-01  1.121e-01   5.960 2.52e-09 ***
## raceAsian                   -5.103e+00  7.991e+00  -0.639  0.52303
## raceCaucasian               -2.922e-01  2.146e-01  -1.362  0.17334
## raceHispanic                 2.186e-01  3.829e-01   0.571  0.56814
## raceNative American          5.464e+02  7.226e+03   0.076  0.93973
## raceOther                   -3.271e-01  4.906e-01  -0.667  0.50487
## priors_count                 3.614e-01  3.662e-02   9.870  < 2e-16 ***
## decile_score                 1.777e-01  1.380e-02  12.880  < 2e-16 ***
## age:raceAsian                5.980e-03  6.115e-02   0.098  0.92209
## age:raceCaucasian            1.471e-02  5.322e-03   2.763  0.00572 **
## age:raceHispanic             4.771e-03  9.116e-03   0.523  0.60075
## age:raceNative American     -1.816e+01  2.157e+02  -0.084  0.93289
## age:raceOther                3.868e-03  1.200e-02   0.322  0.74717
```

```
## age:priors_count                  -2.580e-03  5.409e-04  -4.769 1.85e-06 ***
## sexMale:raceAsian                  2.892e+00  7.514e+00   0.385  0.70028
## sexMale:raceCaucasian             -3.148e-01  1.453e-01  -2.167  0.03023 *
## sexMale:raceHispanic              -5.568e-01  2.607e-01  -2.136  0.03269 *
## sexMale:raceNative American       -1.347e+02  3.793e+03  -0.036  0.97167
## sexMale:raceOther                 -3.822e-02  3.493e-01  -0.109  0.91287
## sexMale:priors_count              -6.024e-02  2.134e-02  -2.822  0.00477 **
## raceAsian:priors_count             1.648e+00  7.204e-01   2.287  0.02217 *
## raceCaucasian:priors_count         7.924e-03  1.653e-02   0.479  0.63165
## raceHispanic:priors_count         -3.298e-02  2.686e-02  -1.228  0.21942
## raceNative American:priors_count  2.207e+01  2.591e+02   0.085  0.93212
## raceOther:priors_count             9.569e-02  5.098e-02   1.877  0.06051 .
## priors_count:decile_score         -1.390e-02  2.625e-03  -5.295 1.19e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9990.5  on 7213  degrees of freedom
## Residual deviance: 8580.9  on 7186  degrees of freedom
## AIC: 8636.9
##
## Number of Fisher Scoring iterations: 16
```

Evaluating Simple Model Performance

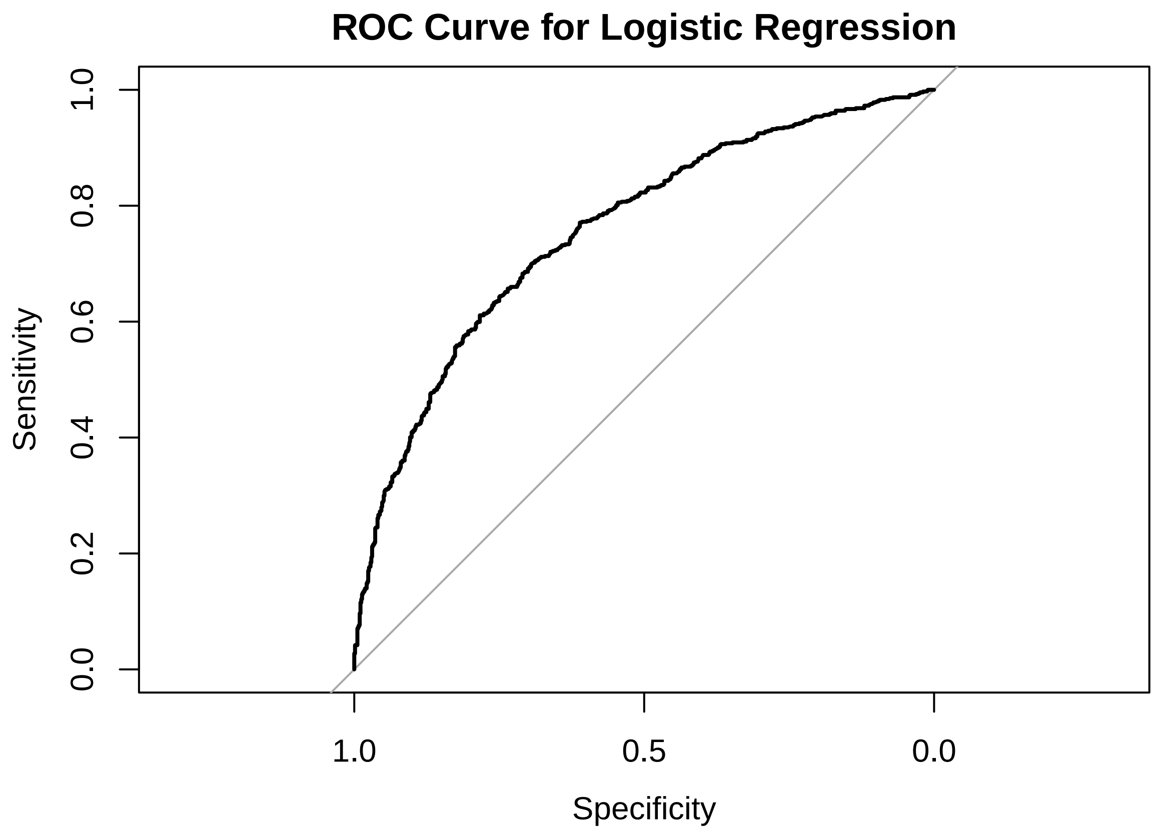
```r
# Split data into training and testing sets
set.seed(999)
trainIndex <- createDataPartition(compas_filtered$is_recid, p = 0.8, list = FALSE)
train_data <- compas_filtered[trainIndex, ]
test_data <- compas_filtered[-trainIndex, ]

# Fit the simple model on training data
simple_mod <- glm(is_recid ~ age + sex + race + priors_count + decile_score,
                  data = train_data,
                  family = binomial)

# Predict probabilities on test data
simple_pred_probs <- predict(simple_mod, newdata = test_data, type = "response")

# Apply a threshold to classify probabilities into binary outcomes
simple_pred_classes <- ifelse(simple_pred_probs > 0.5, 1, 0)

# Confusion Matrix
confusionMatrix(as.factor(simple_pred_classes), test_data$is_recid)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 564 253
##          1 184 441
```

```
##
##                Accuracy : 0.6969
##                  95% CI : (0.6725, 0.7206)
##     No Information Rate : 0.5187
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3909
##
##  Mcnemar's Test P-Value : 0.001142
##
##             Sensitivity : 0.7540
##             Specificity : 0.6354
##          Pos Pred Value : 0.6903
##          Neg Pred Value : 0.7056
##              Prevalence : 0.5187
##          Detection Rate : 0.3911
##    Detection Prevalence : 0.5666
##       Balanced Accuracy : 0.6947
##
##        'Positive' Class : 0
##
```

ROC Analysis and AUC - Simple Model

```r
# Compute ROC curve and AUC
roc_obj <- roc(test_data$is_recid, simple_pred_probs)
plot(roc_obj, main = "ROC Curve for Logistic Regression")
```




```
auc(roc_obj)
```

```
## Area under the curve: 0.757
```

Evaluating Final Model Performance

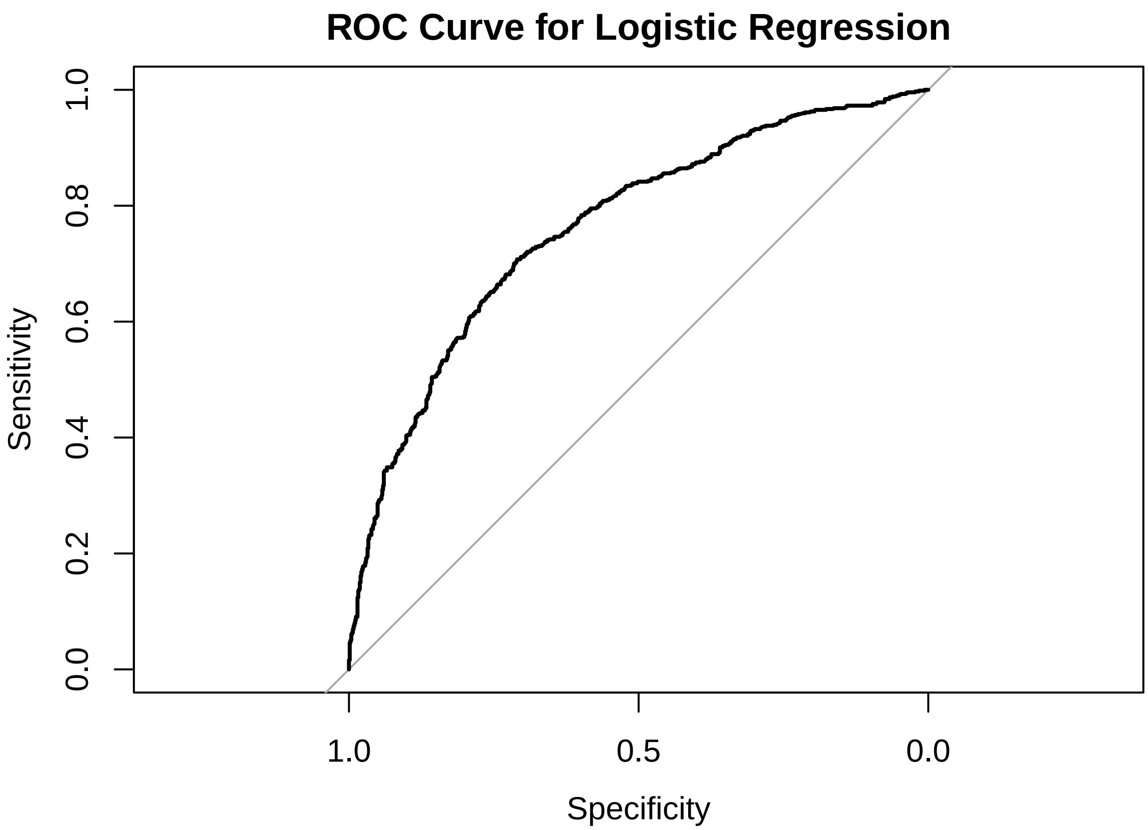
```
# Split data into training and testing sets
set.seed(999)
trainIndex <- createDataPartition(compas_filtered$is_recid, p = 0.8, list = FALSE)
train_data <- compas_filtered[trainIndex, ]
test_data <- compas_filtered[-trainIndex, ]

# Fit the final model on training data
final_mod <- glm(as.formula(paste("is_recid ~", paste(attr(final_model$terms, "term.labels"), collapse =
                 data = train_data,
                 family = binomial)

# Predict probabilities on test data
pred_probs <- predict(final_mod, newdata = test_data, type = "response")

# Apply a threshold to classify probabilities into binary outcomes
pred_classes <- ifelse(pred_probs > 0.5, 1, 0)
```

```r
# Confusion Matrix
confusionMatrix(as.factor(pred_classes), test_data$is_recid)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 561 240
##          1 187 454
##
##                Accuracy : 0.7039
##                  95% CI : (0.6796, 0.7274)
##     No Information Rate : 0.5187
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.4053
##
##  Mcnemar's Test P-Value : 0.01185
##
##             Sensitivity : 0.7500
##             Specificity : 0.6542
##          Pos Pred Value : 0.7004
##          Neg Pred Value : 0.7083
##              Prevalence : 0.5187
##          Detection Rate : 0.3890
##    Detection Prevalence : 0.5555
##       Balanced Accuracy : 0.7021
##
##        'Positive' Class : 0
##
```

ROC Analysis and AUC - Final Model

```r
# Compute ROC curve and AUC
roc_obj <- roc(test_data$is_recid, pred_probs)
plot(roc_obj, main = "ROC Curve for Logistic Regression")
```




```
auc(roc_obj)
```

```
## Area under the curve: 0.7629
```

Group-Level and Individual-Level Analysis

```
# Confidence intervals for group-level predictions
group_summary <- train_data %>%
  group_by(is_recid) %>%
  summarise(mean_decile_score = mean(decile_score),
            ci_lower = mean_decile_score - 1.96 * sd(decile_score) / sqrt(n()),
            ci_upper = mean_decile_score + 1.96 * sd(decile_score) / sqrt(n()))

group_summary
```

```
## # A tibble: 2 x 4
##   is_recid mean_decile_score ci_lower ci_upper
##   <fct>                <dbl>    <dbl>    <dbl>
## 1 0                     3.56     3.47     3.65
## 2 1                     5.54     5.43     5.64
```

```r
# Prediction intervals for individual predictions
prediction_intervals <- test_data %>%
  mutate(predicted_prob = pred_probs,
         lower_bound = predicted_prob - 1.96 * sqrt(predicted_prob * (1 - predicted_prob)),
         upper_bound = predicted_prob + 1.96 * sqrt(predicted_prob * (1 - predicted_prob)))

head(prediction_intervals, n=10)
```

```
##    age    sex             race priors_count decile_score is_recid
## 2   34   Male African-American            0            3        1
## 5   43   Male            Other            2            1        0
## 21  64   Male African-American           13            6        1
## 31  33   Male African-American            0           10        1
## 34  55   Male        Caucasian            0            1        0
## 48  49   Male        Caucasian            0            1        0
## 49  34   Male African-American            4            3        1
## 53  48   Male        Caucasian           20            6        1
## 54  63 Female         Hispanic            1            1        0
## 58  49   Male African-American            4            1        0
##    predicted_prob lower_bound upper_bound
## 2       0.3478665 -0.58566854   1.2814016
## 5       0.2908019 -0.59929743   1.1809013
## 21      0.4213833 -0.54642697   1.3891935
## 31      0.6562205 -0.27471787   1.5871589
## 34      0.1936776 -0.58087360   0.9682288
## 48      0.2102086 -0.58840650   1.0088238
## 49      0.5116694 -0.46806371   1.4914024
## 53      0.8267377  0.08492882   1.5685465
## 54      0.1683565 -0.56504131   0.9017543
## 58      0.3100061 -0.59648556   1.2164978
```