P8106 Midterm Project

Leyang Rui, Jinghan Zhao

2025-03-28

Load Data

```
load("data/dat1.RData")
train_data = dat1 |>
  janitor::clean_names() |>
  mutate(
    gender = as.factor(gender),
    diabetes = as.factor(diabetes),
    hypertension = as.factor(hypertension),
    race = fct_recode(race,
                  White = "1",
                  Asian = "2",
                  Black = "3",
                  Hispanic = "4"),
    gender = fct_recode(gender,
                        Male = "1",
                        Female = "0"),
    smoking = fct_recode(smoking,
                         "Never smoked" = "0",
                         "Former smoker" = "1",
                         "Current smoker" = "2"))

load("data/dat2.RData")
test_data = dat2 |>
  janitor::clean_names() |>
  mutate(
    gender = as.factor(gender),
    diabetes = as.factor(diabetes),
    hypertension = as.factor(hypertension),
    race = fct_recode(race,
                  White = "1",
                  Asian = "2",
                  Black = "3",
                  Hispanic = "4"),
    gender = fct_recode(gender,
                        Male = "1",
                        Female = "0"),
    smoking = fct_recode(smoking,
                         "Never smoked" = "0",
                         "Former smoker" = "1",
```

```
                          "Current smoker" = "2")
  )
```

Modify Data

```
train_data1 =
  train_data %>%
  select(-id, -height, -weight, -hypertension)

x_train = model.matrix(log_antibody ~ ., train_data1)[, -1]
colnames(x_train) = make.names(colnames(x_train), unique = TRUE)
y_train = train_data1[, "log_antibody"]

test_data =
  test_data %>%
  select(-id, -height, -weight, -hypertension)

x_test = model.matrix(log_antibody ~ ., test_data)[, -1]
colnames(x_test) = make.names(colnames(x_test), unique = TRUE)
y_test = test_data[, "log_antibody"]

ctrl1 = trainControl(method = "cv", number = 10)
```

Descriptive Analysis

Numeric Variables

```
train_data |>
  pivot_longer(
    cols = c(age, height, weight, bmi),
    names_to = "variable",
    values_to = "value"
  ) |>
  ggplot(aes(x = value, y = log_antibody, color = variable)) +
  geom_point(alpha = 0.5, size = 0.6) +
  facet_wrap(variable ~ .,  scales = "free") +
    labs(title = "Distribution of the Demographic Continuous Variables",
         x = "Variables",
         y = "Log_Antibody")
```

```
train_data |>
  pivot_longer(
    cols = c(sbp, ldl, time),
    names_to = "variable",
    values_to = "value"
  ) |>
  ggplot(aes(x = value, y = log_antibody, color = variable)) +
  geom_point(alpha = 0.5, size = 0.6) +
```

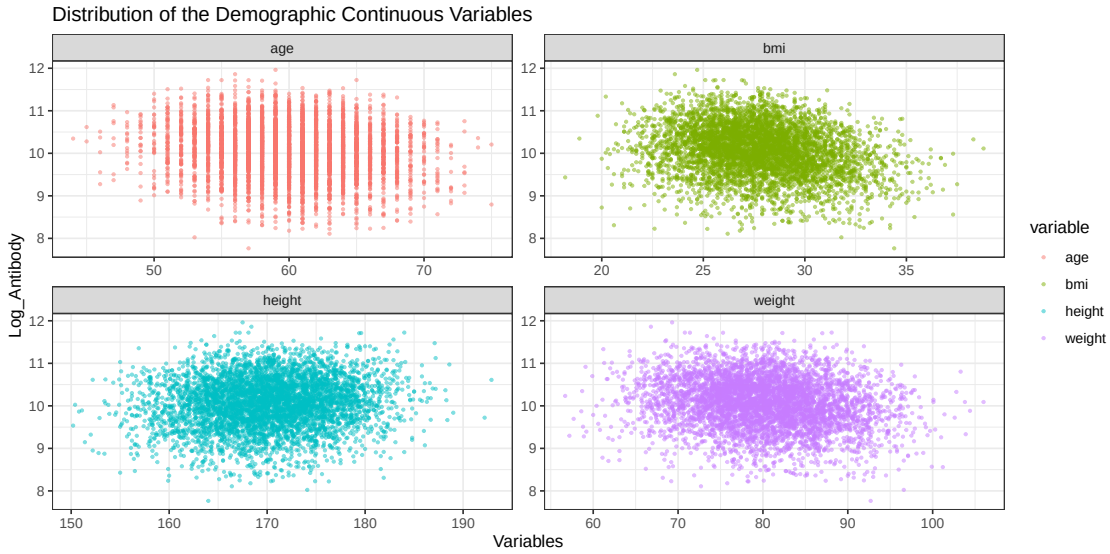

Figure 1: Distribution of the Demographic Continuous Variables

```
  facet_wrap(variable ~ .,  scales = "free") +
    labs(title = "Distribution of the Clinical Continuous Variables",
         x = "Variables",
         y = "Log_Antibody")
```

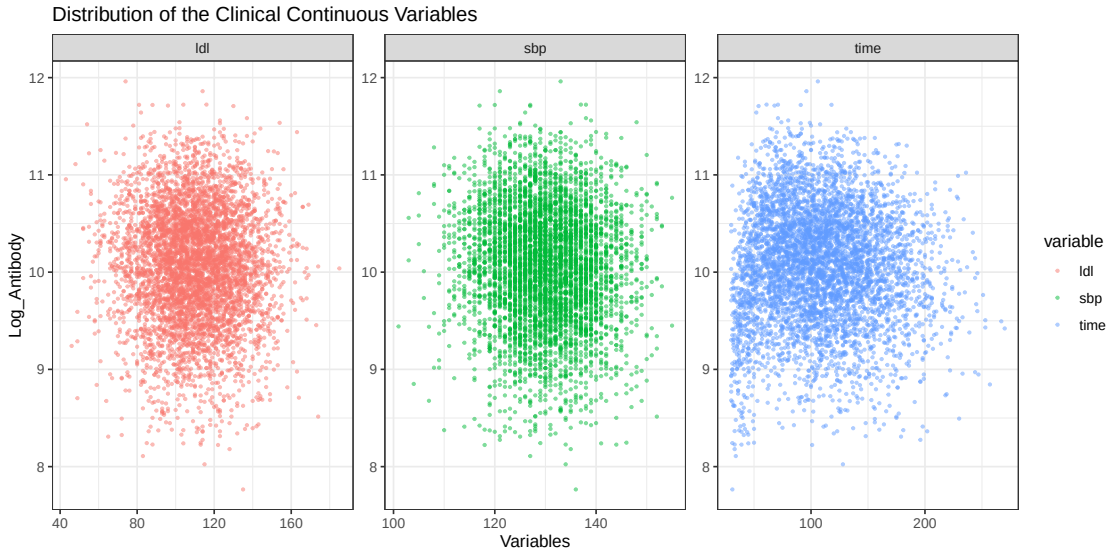

Figure 2: Distribution of the Clinical Continuous Variables

```
train_data %>%
  pivot_longer(
    cols = c(age, height, weight, bmi, sbp, ldl, time, log_antibody),
    names_to = "variable_name",
    values_to = "value"
```

```
  ) %>%
  group_by(variable_name) %>%
  summarize(
    mean = mean(value),
    median = median(value),
    min = min(value),
    first_quantile = quantile(value, probs = 0.25),
    third_quantile = quantile(value, probs = 0.75),
    max = max(value)
  ) %>%
  ungroup() %>%
  arrange(desc(variable_name == "log_antibody"), variable_name) %>%
  knitr::kable(digits = 3, caption = "Descriptive Statistics for the Continuous Variables")
```

Table 1: Descriptive Statistics for the Continuous Variables

variable_name	mean	median	min	first_quantile	third_quantile	max
log_antibody	10.064	10.089	7.765	9.682	10.478	11.961
age	59.968	60.000	44.000	57.000	63.000	75.000
bmi	27.740	27.600	18.200	25.800	29.500	38.800
height	170.126	170.100	150.200	166.100	174.225	192.900
ldl	109.909	110.000	43.000	96.000	124.000	185.000
sbp	129.900	130.000	101.000	124.000	135.000	155.000
time	108.863	106.000	30.000	76.000	138.000	270.000
weight	80.109	80.100	56.700	75.400	84.900	106.000

Categorical Variables

```
train_data |>
  pivot_longer(
    cols = c(gender, race, smoking, diabetes, hypertension),
    names_to = "variable",
    values_to = "value"
  ) |>
  mutate(
    variable = factor(variable, levels = c("gender", "race", "smoking", "diabetes", "hypertension"))
  ) |>
  ggplot(aes(x = value, y = log_antibody, fill = variable)) +
  geom_boxplot(alpha = 0.5) +
  facet_wrap(variable ~ .,  scales = "free") +
    labs(title = "Distribution of the Categorical Variables",
         x = "Variables",
         y = "Log_Antibody") +
  theme(axis.text.x = element_text(angle = 30, vjust = 1, hjust = 1))
```

Correlation Plot

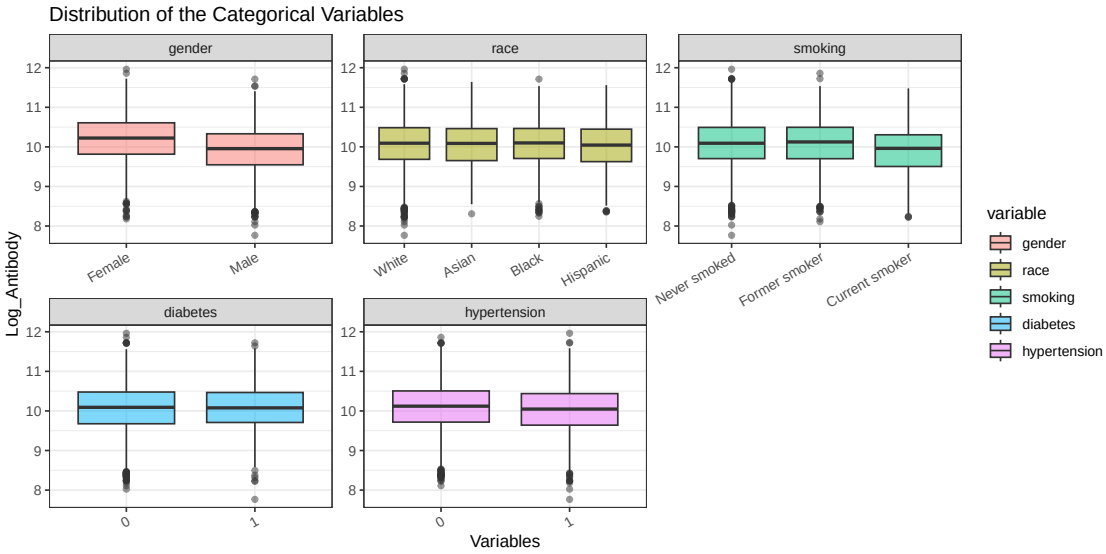

Figure 3: Distribution of the Categorical Variables

```
x_corr = model.matrix(log_antibody ~ ., train_data[, -1])[, -1]
corrplot(cor(x_corr), method = "circle", type = "full")
```

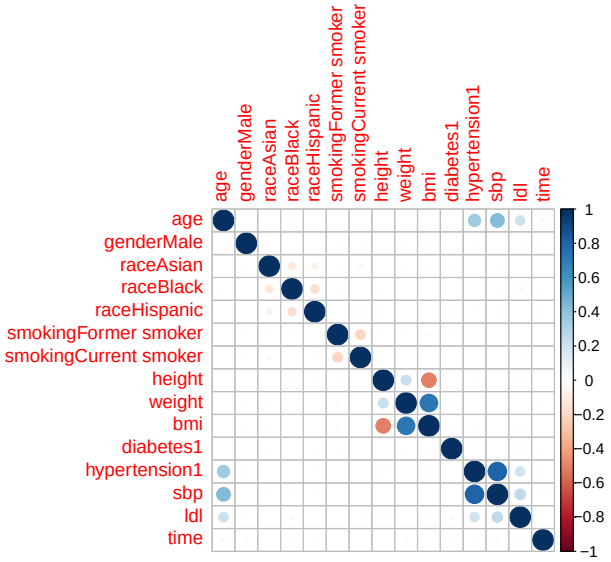

Figure 4: Correlation Plot

Regression

Elastic Net

```
set.seed(37)
enet_fit = train(log_antibody ~ .,
                 data = train_data1,
                 method = "glmnet",
                 tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                        lambda = exp(seq(-2, -8, length = 100))),
                 trControl = ctrl1)
enet_fit$bestTune
```

```
##      alpha      lambda
## 2044     1 0.004544037
```

```
coef(enet_fit$finalModel, enet_fit$bestTune$lambda)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                                    s1
## (Intercept)             1.272351e+01
## age                    -1.914947e-02
## genderMale             -2.859704e-01
## raceAsian               .
## raceBlack               .
## raceHispanic           -2.469963e-02
## smokingFormer smoker    1.592525e-02
## smokingCurrent smoker -1.770936e-01
## bmi                    -4.820177e-02
## diabetes1               4.263422e-05
## sbp                     .
## ldl                     .
## time                   -1.850422e-04
```

```
mycol = rainbow(25)
mypar = list(superpose.symbol = list(col = mycol),
superpose.line = list(col = mycol))

plot(enet_fit, par.settings = mypar, xTrans = log)
```

PCR

```
set.seed(37)
pcr_fit = train(x_train, y_train,
                method = "pcr",
                tuneGrid = data.frame(ncomp = 1:12),
                trControl = ctrl1,
                preProcess = c("center", "scale"))
summary(pcr_fit)
```

```
## Data:    X dimension: 5000 12
##  Y dimension: 5000 1
```

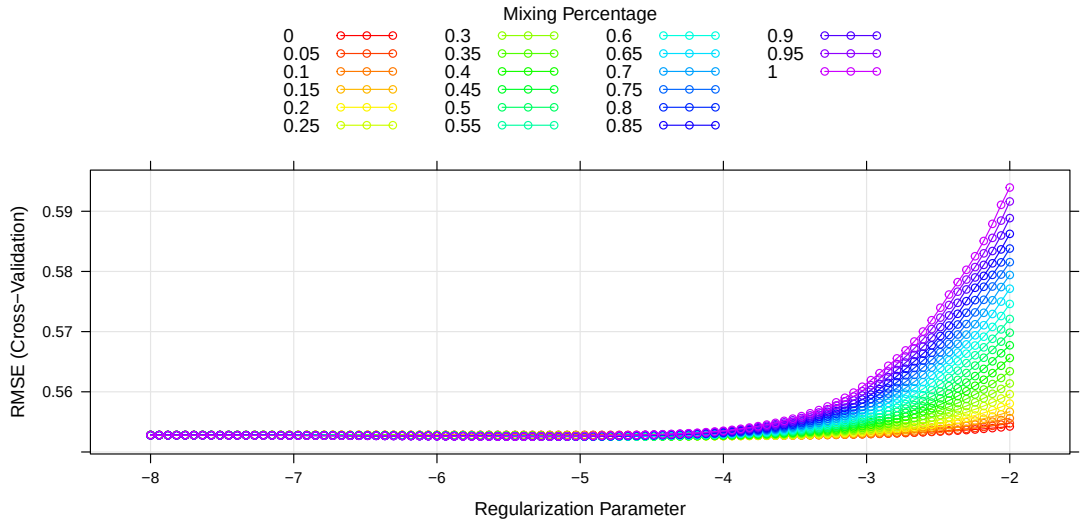

Figure 5: Effect of Tuning Parameters on Train Error (Elastic Net)

```
## Fit method: svdpc
## Number of components considered: 12
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X          13.522   23.760   33.489   42.534    51.17   59.563    67.70
## .outcome    1.296    1.493    1.512    1.512     1.54    2.032    13.44
##           8 comps  9 comps  10 comps  11 comps  12 comps
## X           75.76    82.63     89.32     95.36     100.0
## .outcome    13.48    13.68     13.78     13.84      14.5
```

```
ggplot(pcr_fit, highlight = TRUE)
```

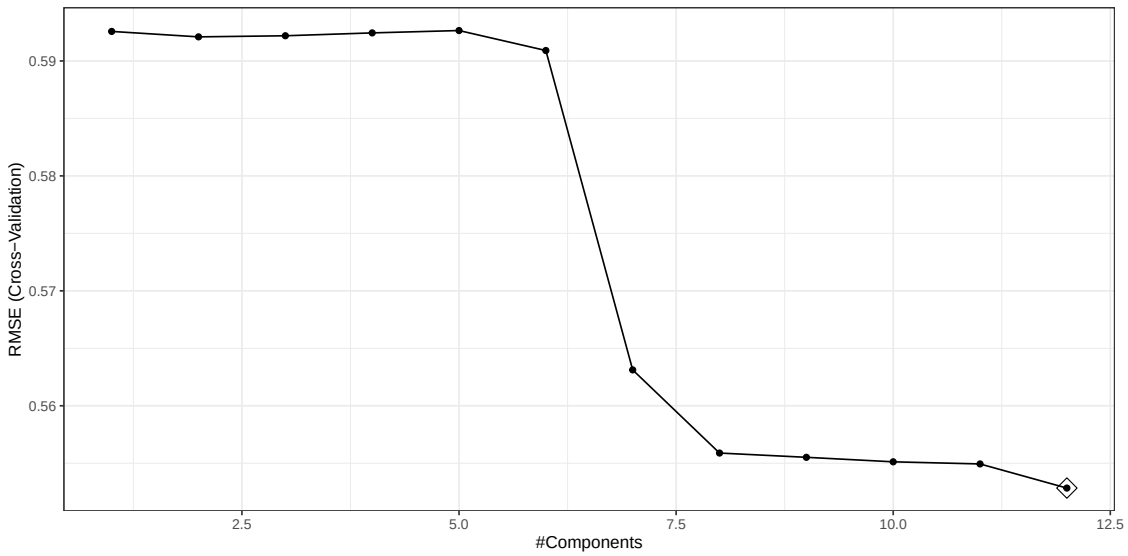

Figure 6: Component Selection (PCR)

PLS

```
set.seed(37)
pls_fit = train(x_train, y_train,
                method = "pls",
                tuneGrid = data.frame(ncomp = 1:12),
                trControl = ctrl1,
                preProcess = c("center", "scale"))
summary(pls_fit)
```

```
## Data:    X dimension: 5000 12
##  Y dimension: 5000 1
## Fit method: oscorespls
## Number of components considered: 3
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps
## X           9.295    21.21    26.88
## .outcome   13.885    14.45    14.50
```

```
ggplot(pls_fit, highlight = TRUE)
```

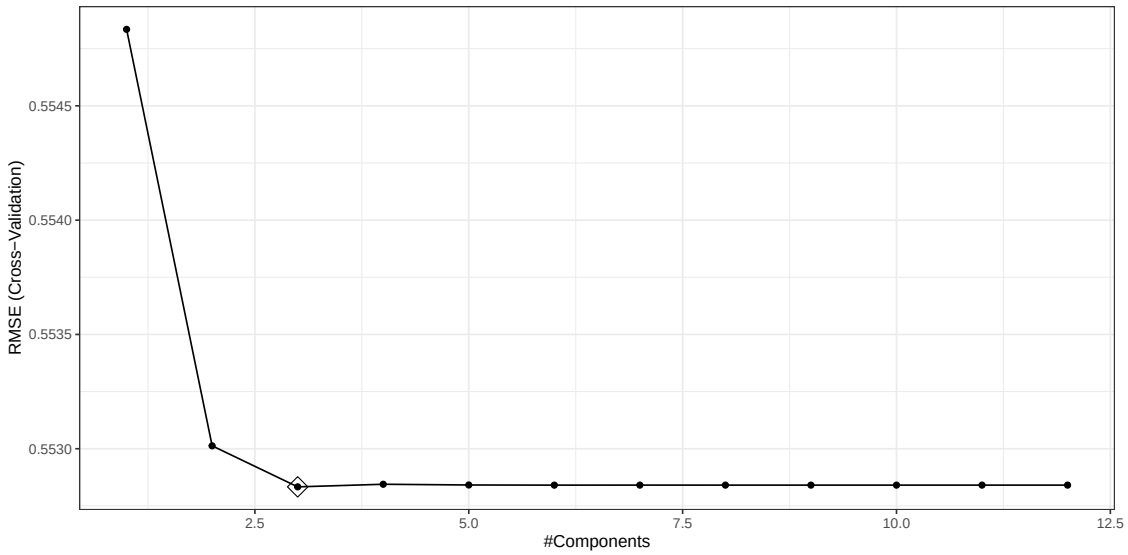

Figure 7: Component Selection (PLS)

GAM

```
set.seed(37)

gam.fit = train(x_train, y_train,
                method = "gam",
                trControl = ctrl1)
```

```
gam.fit$bestTune
```

```
##   select method
## 2   TRUE GCV.Cp
```

```
gam.fit$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ genderMale + raceAsian + raceBlack + raceHispanic +
##     smokingFormer.smoker + smokingCurrent.smoker + diabetes1 +
##     s(age) + s(sbp) + s(ldl) + s(bmi) + s(time)
##
## Estimated degrees of freedom:
## 0.992 0.000 0.000 4.179 7.915  total = 21.09
##
## GCV score: 0.2786375
```

```
par(mfrow = c(3, 2))
plot(gam.fit$finalModel)

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

MARS

```
set.seed(37)

mars_grid = expand.grid(degree = 1:3,
                        nprune = 2:12)

mars.fit = train(x_train, y_train,
                 method = "earth",
                 tuneGrid = mars_grid,
                 trControl = ctrl1)

ggplot(mars.fit)
```

```
mars.fit$bestTune
```

```
##   nprune degree
## 8      9      1
```

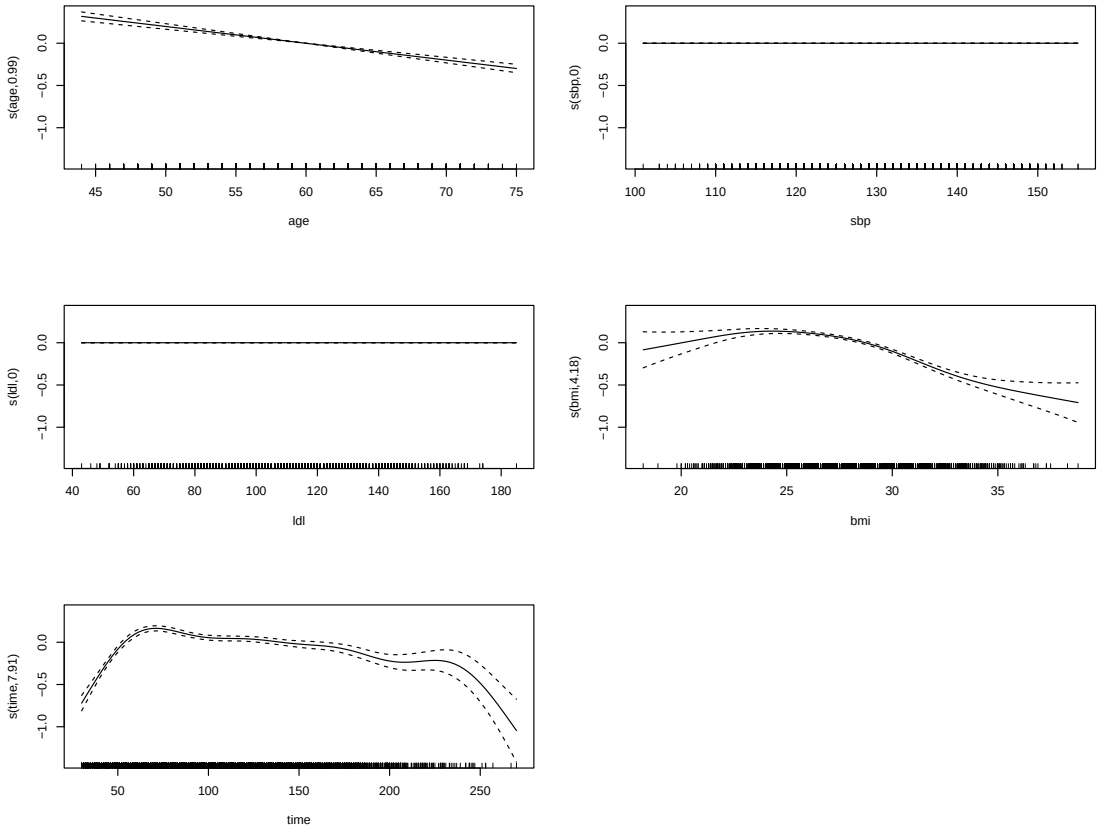

Figure 8: Degree of Predictors (GAM)

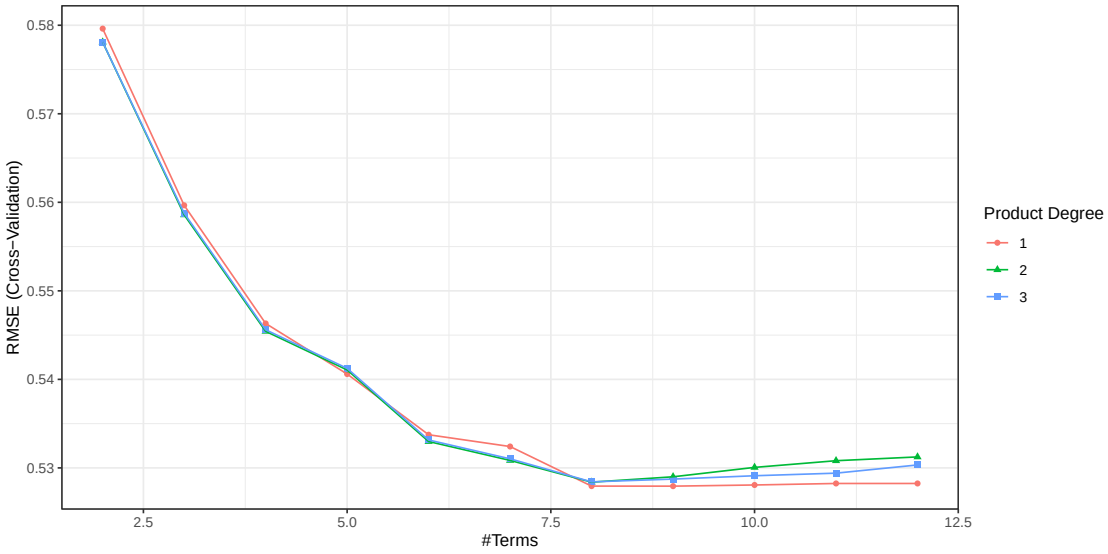

Figure 9: Term and Degree Selection (MARS)

```r
coef(mars.fit$finalModel)
```

```
##           (Intercept)           h(27.8-bmi)           h(time-57)
##          10.847446930          -0.061997354          -0.002254182
##           h(57-time)            genderMale            h(age-59)
##          -0.033529326          -0.296290451          -0.022957648
##            h(59-age) smokingCurrent.smoker           h(bmi-23.7)
##           0.016138468          -0.205126851          -0.084380175
```

Regression Trees

```r
set.seed(37)

tree.fit = train(x_train, y_train,
                 method = "rpart",
                 tuneGrid = data.frame(cp = exp(seq(-10, 0, length = 100,))),
                 trControl = ctrl1)

plot(tree.fit, xTrans = log)
```

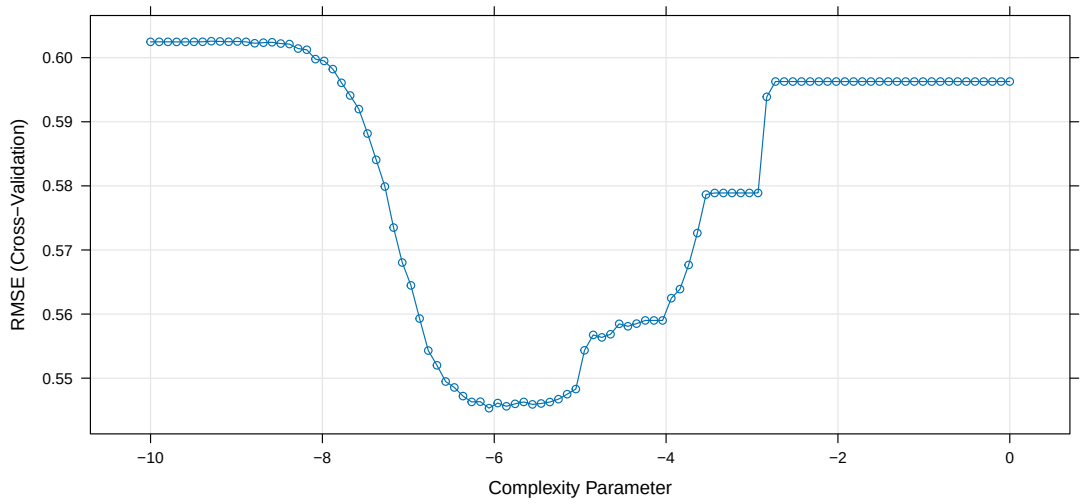

Figure 10: Tuning Parameter Selection (Regression Tree)

```r
rpart.plot(tree.fit$finalModel)
```

Comparison

```r
resamp = resamples(list(elastic_net = enet_fit,
                        pcr = pcr_fit,
```

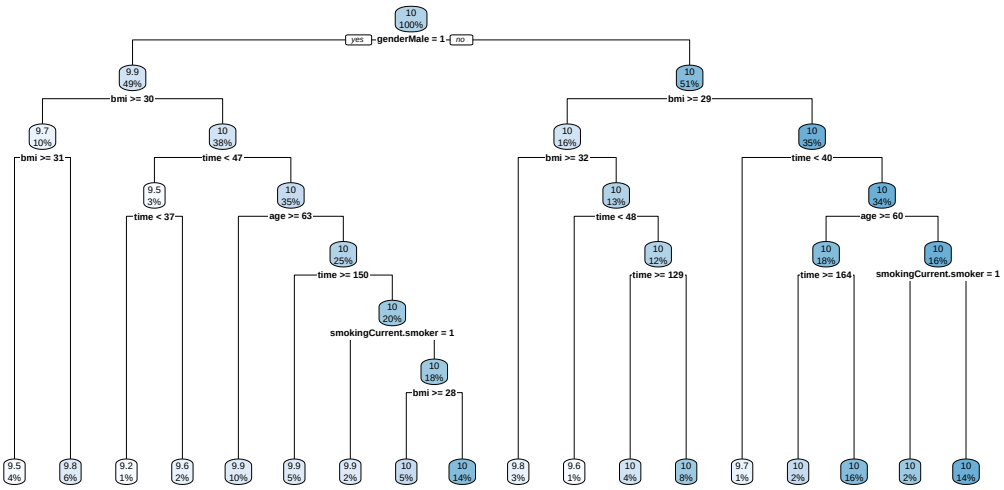

Figure 11: Final Regression Tree Model

```
                       pls = pls_fit,
                       gam = gam.fit,
                       mars = mars.fit,
                       tree = tree.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: elastic_net, pcr, pls, gam, mars, tree
## Number of resamples: 10
##
## MAE
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## elastic_net 0.4294666 0.4333146 0.4414904 0.4404188 0.4479709 0.4502491    0
## pcr         0.4283770 0.4331431 0.4418042 0.4406867 0.4484017 0.4516552    0
## pls         0.4283860 0.4331977 0.4418812 0.4407023 0.4483486 0.4517004    0
## gam         0.4038951 0.4165022 0.4250940 0.4229309 0.4317016 0.4337601    0
## mars        0.4068067 0.4144508 0.4248340 0.4221903 0.4299231 0.4327355    0
## tree        0.4229802 0.4279666 0.4363778 0.4353252 0.4432244 0.4468699    0
##
## RMSE
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## elastic_net 0.5346950 0.5401593 0.5540536 0.5525646 0.5599590 0.5733181    0
## pcr         0.5348357 0.5398323 0.5544163 0.5528409 0.5604510 0.5739030    0
## pls         0.5347609 0.5398986 0.5544173 0.5528333 0.5603197 0.5740022    0
## gam         0.5101573 0.5227377 0.5286720 0.5286310 0.5327635 0.5511329    0
## mars        0.5115426 0.5227742 0.5278403 0.5279391 0.5326822 0.5440279    0
## tree        0.5295247 0.5387966 0.5445051 0.5453183 0.5539628 0.5570269    0
##
## Rsquared
```

```
##                  Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## elastic_net 0.1039358 0.1362602 0.1422293 0.1424292 0.1471805 0.1856091    0
## pcr         0.1036311 0.1369615 0.1396188 0.1414866 0.1456933 0.1846009    0
## pls         0.1037262 0.1368722 0.1397305 0.1415124 0.1458437 0.1846014    0
## gam         0.1629446 0.2041048 0.2183934 0.2154693 0.2374062 0.2522148    0
## mars        0.1561416 0.2106884 0.2225692 0.2174307 0.2382161 0.2471323    0
## tree        0.1129441 0.1451478 0.1704858 0.1676913 0.1904570 0.2117980    0
```

```r
bwplot(resamp, metric = "RMSE")
```

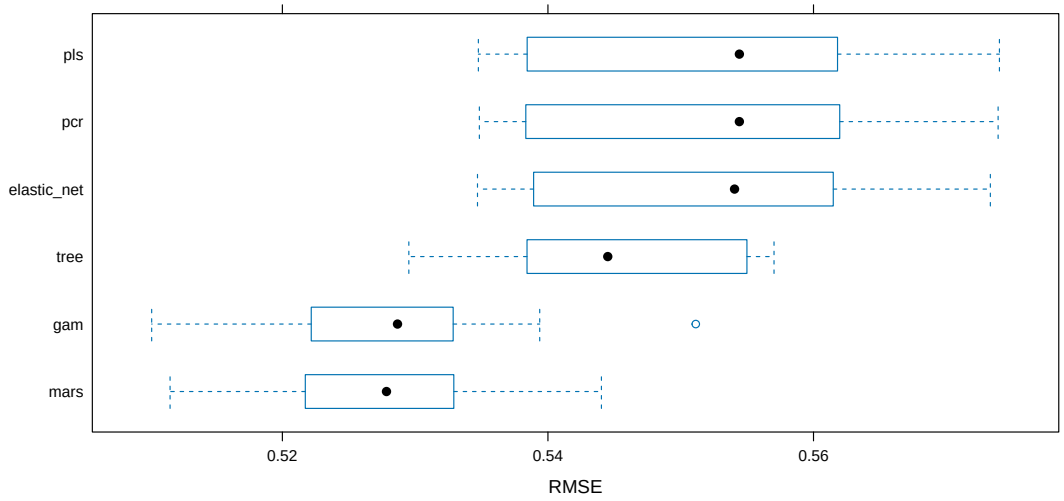

Figure 12: Model Selection

Model Performance

```r
predicted_values = predict(mars.fit$finalModel, newdata = x_test)

residuals = y_test - predicted_values

plot(predicted_values, residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residuals vs Fitted Values (MARS)",
     pch = 20, col = "mediumpurple1")
abline(h = 0, col = "blue", lwd = 2)
```

```r
plot(y_test, predicted_values,
     xlab = "Actual Values", ylab = "Predicted Values",
     main = "Prediction vs Actual (MARS)",
     pch = 20, col = "orange")
abline(0, 1, col = "mediumseagreen", lwd = 2)
```

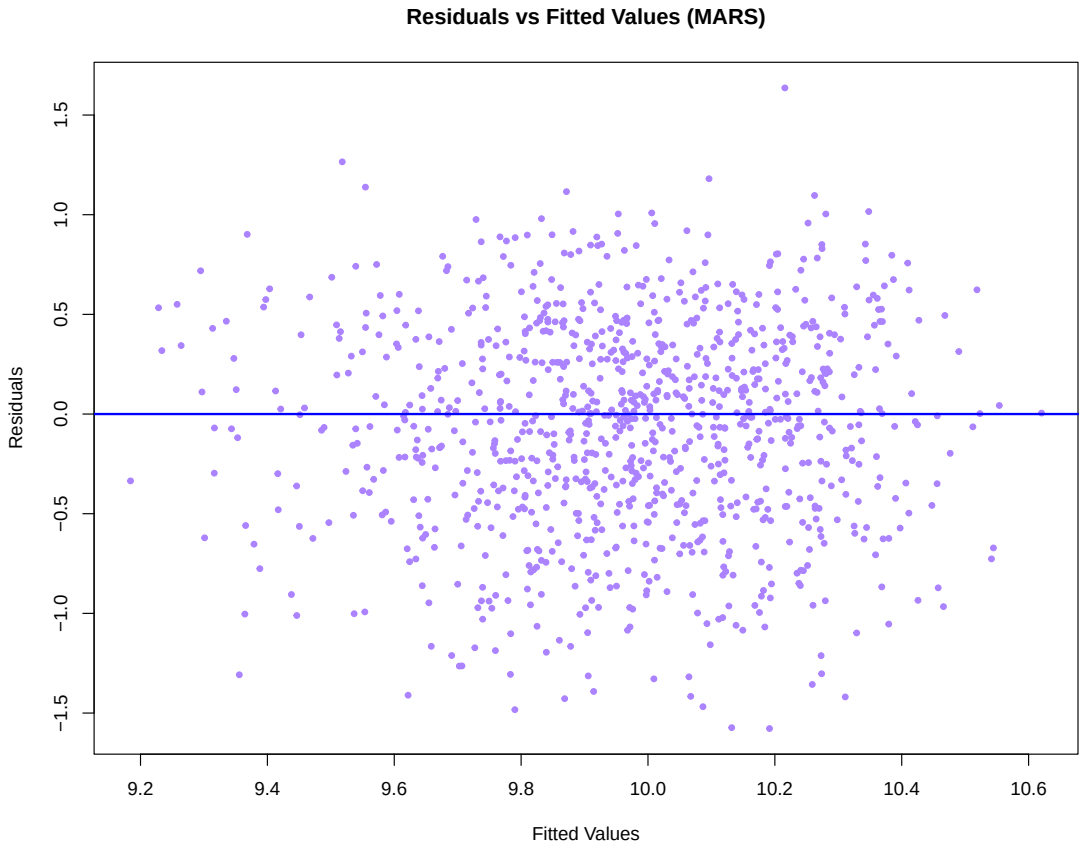

Figure 13: Residuals vs Fitted Values (MARS)

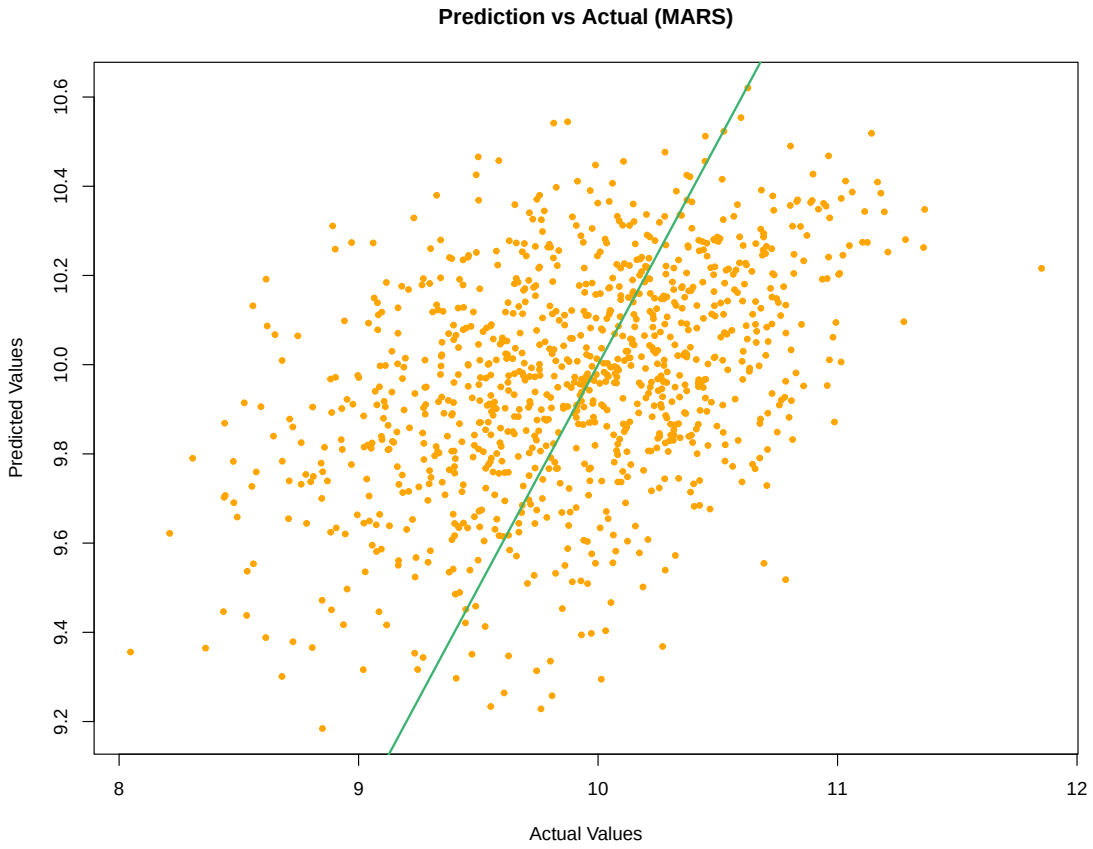

Figure 14: Prediction vs Actual (MARS)

```r
rmse = sqrt(mean((y_test - predicted_values)^2))
rmse
```

```
## [1] 0.5327718
```