Red Wines - upper

Katie, Rita, and Chang

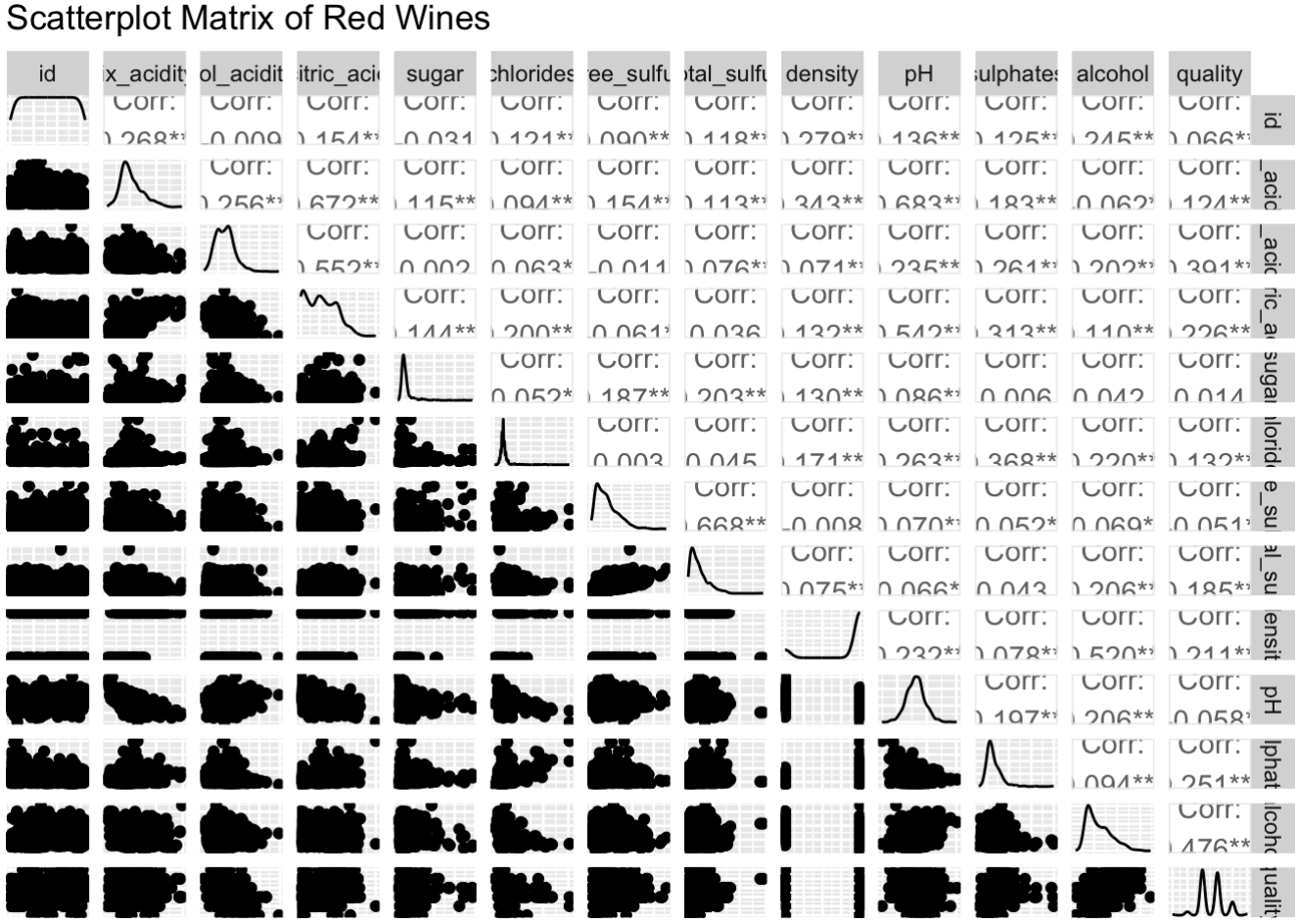
2023-11-01

Scatterplot Matrix

```
library("GGally")
ggpairs(red, axisLabels = "none",
        title = "Scatterplot Matrix of Red Wines")
```

```
# corr codes
```

Scatterplot Matrix



Create Binary Dependent Variable

```
red$highquality = factor((red$quality >= 6))
red$highquality <- as.integer(as.logical(red$highquality))
```

Descriptive Statistics

```
library("Rmisc")
```

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

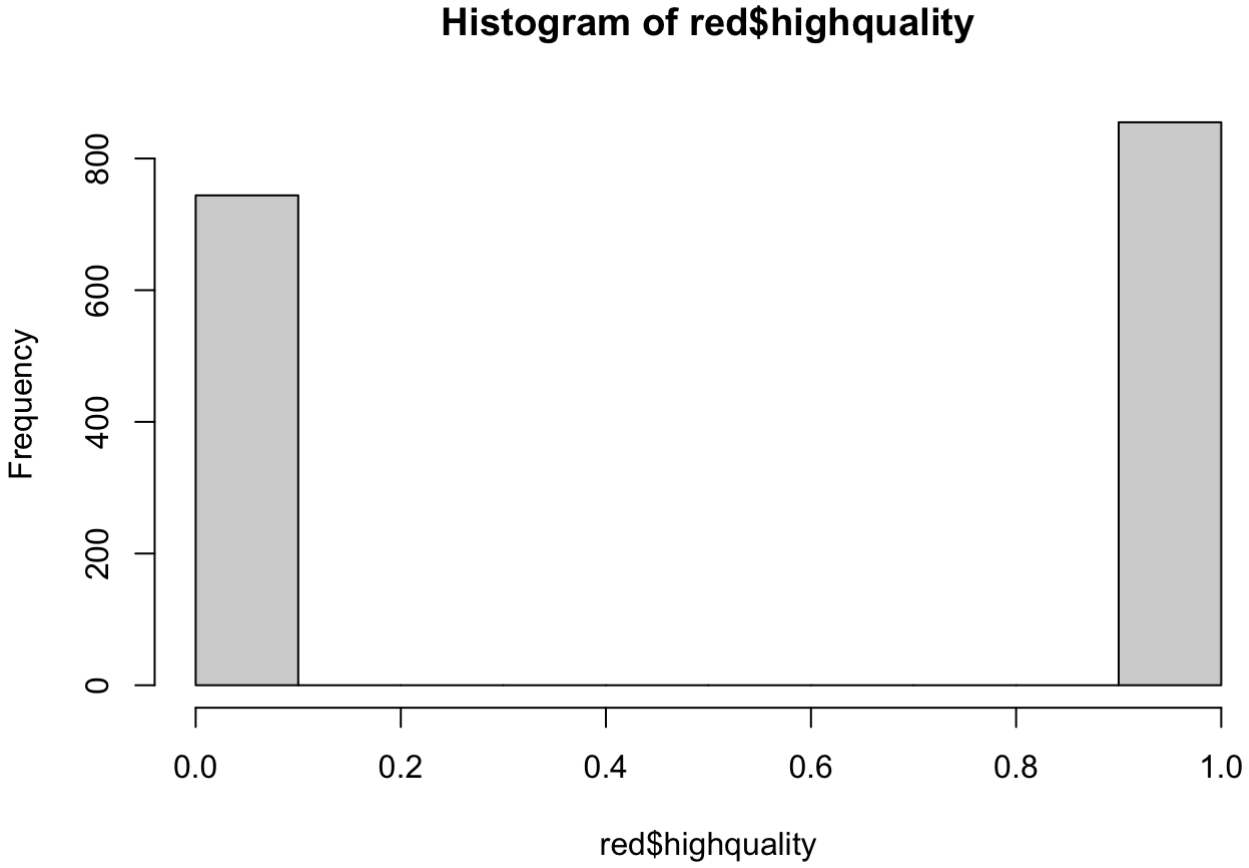
```
## Loading required package: plyr
```

```
sum = summary(red)
sum
```

```
##        id          fix_acidity     vol_acidity      citric_acid
##  Min.   :   1.0   Min.   : 4.60   Min.   :0.1200   Min.   :0.000
##  1st Qu.: 400.5   1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090
##  Median : 800.0   Median : 7.90   Median :0.5200   Median :0.260
##  Mean   : 800.0   Mean   : 8.32   Mean   :0.5284   Mean   :0.271
##  3rd Qu.:1199.5   3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420
##  Max.   :1599.0   Max.   :15.90   Max.   :1.5800   Max.   :1.000
##      sugar          chlorides        free_sulfur     total_sulfur
##  Min.   : 0.900   Min.   :0.01000   Min.   : 1.00   Min.   :  6.00
##  1st Qu.: 1.900   1st Qu.:0.07000   1st Qu.: 7.00   1st Qu.: 22.00
##  Median : 2.200   Median :0.08000   Median :14.00   Median : 38.00
##  Mean   : 2.539   Mean   :0.08787   Mean   :15.87   Mean   : 46.47
##  3rd Qu.: 2.600   3rd Qu.:0.09000   3rd Qu.:21.00   3rd Qu.: 62.00
##  Max.   :15.500   Max.   :0.61000   Max.   :72.00   Max.   :289.00
##     density             pH          sulphates         alcohol
##  Min.   :0.9900   Min.   :2.740   Min.   :0.3300   Min.   : 8.40
##  1st Qu.:1.0000   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50
##  Median :1.0000   Median :3.310   Median :0.6200   Median :10.20
##  Mean   :0.9985   Mean   :3.311   Mean   :0.6581   Mean   :10.42
##  3rd Qu.:1.0000   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
##  Max.   :1.0000   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##     quality       highquality
##  Min.   :3.000   Min.   :0.0000
##  1st Qu.:5.000   1st Qu.:0.0000
##  Median :6.000   Median :1.0000
##  Mean   :5.636   Mean   :0.5347
##  3rd Qu.:6.000   3rd Qu.:1.0000
##  Max.   :8.000   Max.   :1.0000
```

Plot high quality vs low quality distribution

```
hist (red$highquality)
```



Random Forest

```
library("randomForest")
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library("caret")
library("e1071")
library("rpart")

rf <- randomForest(highquality ~ . - quality, data = red, mtry = 4, importance = TRUE, n
tree = 50, na.action = na.omit)
```

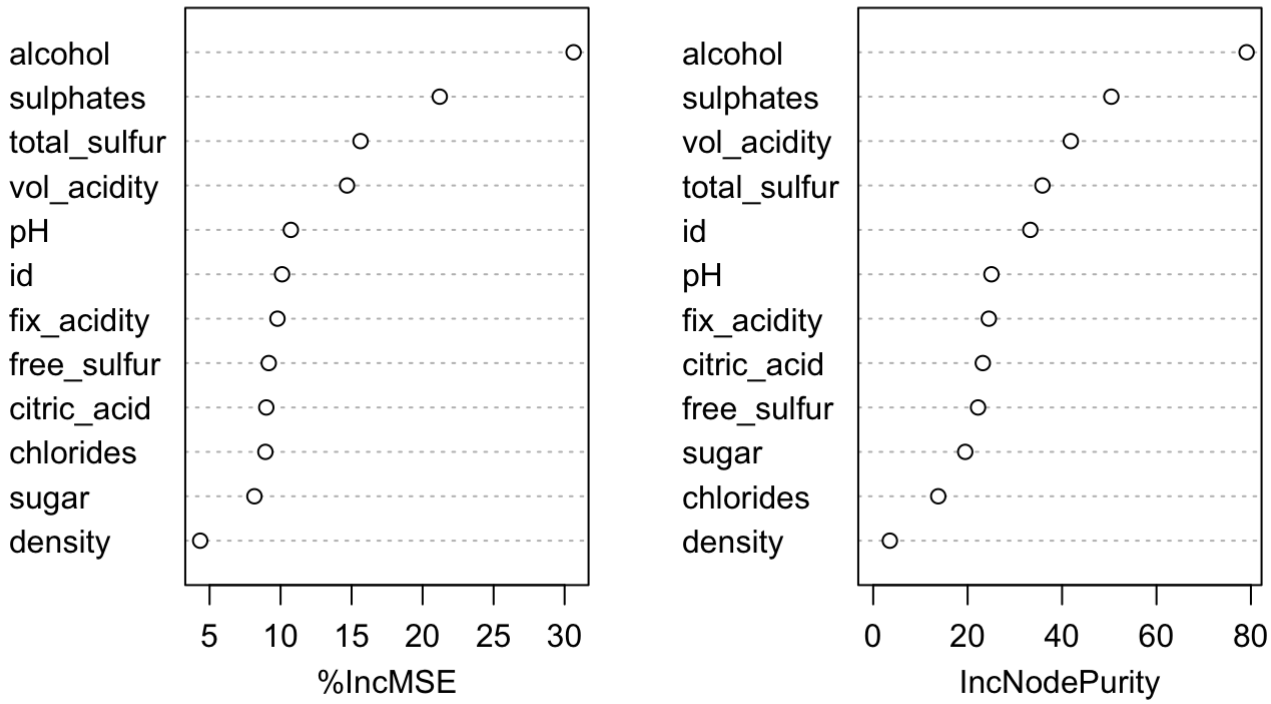
```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
print(rf)
```

```
##
## Call:
##  randomForest(formula = highquality ~ . - quality, data = red,      mtry = 4, importa
nce = TRUE, ntree = 50, na.action = na.omit)
##                Type of random forest: regression
##                      Number of trees: 50
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 0.136926
##                     % Var explained: 44.96
```

```
varImpPlot(rf)
```

rf



Random Forest Model

```
# Logit
randomforestmodlogit <- glm(highquality ~ alcohol + sulphates + total_sulfur + vol_acidi
ty, data = red, family = "binomial"(link = "logit"))
summary(randomforestmodlogit)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     vol_acidity, family = binomial(link = "logit"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1638  -0.8675   0.3076   0.8629   2.3262
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.588813   0.795118 -10.802  < 2e-16 ***
## alcohol       0.927362   0.069268  13.388  < 2e-16 ***
## sulphates     2.059047   0.365976   5.626 1.84e-08 ***
## total_sulfur -0.011976   0.001924  -6.225 4.83e-10 ***
## vol_acidity  -3.083277   0.364832  -8.451  < 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: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1684.2  on 1594  degrees of freedom
## AIC: 1694.2
##
## Number of Fisher Scoring iterations: 4
```

```
# Cloglog
randomforestmodcloglog <- glm(highquality ~ alcohol + sulphates + total_sulfur + vol_aci
dity, data = red, family = "binomial"(link = "cloglog"))
summary(randomforestmodcloglog)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     vol_acidity, family = binomial(link = "cloglog"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5006  -0.9020   0.2185   0.9295   2.0506
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.958517   0.478252 -10.368  < 2e-16 ***
## alcohol       0.505807   0.038543  13.123  < 2e-16 ***
## sulphates     1.324184   0.221318   5.983 2.19e-09 ***
## total_sulfur -0.009109   0.001364  -6.679 2.41e-11 ***
## vol_acidity  -2.022997   0.238813  -8.471  < 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: 2209  on 1598  degrees of freedom
## Residual deviance: 1701  on 1594  degrees of freedom
## AIC: 1711
##
## Number of Fisher Scoring iterations: 7
```

```
# The logit model performed better with a lower AIC value
```

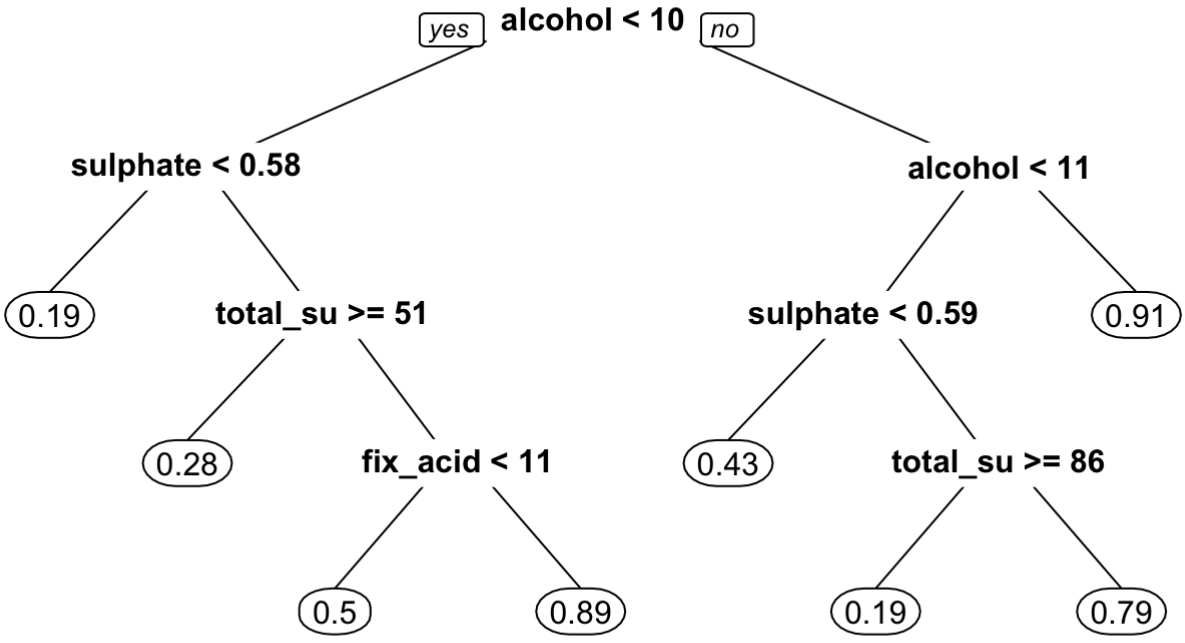
Cart

```
library("randomForest")
library("caret")
library("e1071")
library("rpart")
library("rpart.plot")

cartmodel = rpart(highquality ~ . - quality, data = red)
print(cartmodel)
```

```
## n= 1599
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 1599 397.823600 0.5347092
##    2) alcohol< 10.25 842 189.174600 0.3408551
##      4) sulphates< 0.575 353  54.900850 0.1926346 *
##      5) sulphates>=0.575 489 120.920200 0.4478528
##       10) total_sulfur>=50.5 204  41.509800 0.2843137 *
##       11) total_sulfur< 50.5 285  70.049120 0.5649123
##         22) fix_acidity< 10.75 239  59.748950 0.5020921 *
##         23) fix_acidity>=10.75 46   4.456522 0.8913043 *
##    3) alcohol>=10.25 757 141.812400 0.7503303
##      6) alcohol< 11.45 477 107.299800 0.6582809
##       12) sulphates< 0.585 134  32.753730 0.4253731 *
##       13) sulphates>=0.585 343  64.437320 0.7492711
##         26) total_sulfur>=85.5 21   3.238095 0.1904762 *
##         27) total_sulfur< 85.5 322  54.214290 0.7857143 *
##      7) alcohol>=11.45 280  23.585710 0.9071429 *
```

```
prp(cartmodel)
```



Cart Model

```
# Logit
cartmodlogit <- glm(highquality ~ alcohol + sulphates + total_sulfur + fix_acidity, data
= red, family = "binomial"(link = "logit"))
summary(cartmodlogit)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     fix_acidity, family = binomial(link = "logit"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3737  -0.9154   0.3562   0.8762   2.0206
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -12.172146   0.828396 -14.694  < 2e-16 ***
## alcohol        0.989178   0.068276  14.488  < 2e-16 ***
## sulphates      2.587844   0.370028   6.994 2.68e-12 ***
## total_sulfur  -0.011171   0.001895  -5.895 3.75e-09 ***
## fix_acidity    0.109461   0.035511   3.082  0.00205 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1752.5  on 1594  degrees of freedom
## AIC: 1762.5
##
## Number of Fisher Scoring iterations: 4
```

```
# Cloglog
cartmodcloglog <- glm(highquality ~ alcohol + sulphates + total_sulfur + fix_acidity, da
ta = red, family = "binomial"(link = "cloglog"))
summary(cartmodcloglog)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     fix_acidity, family = binomial(link = "cloglog"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7058  -0.9408   0.3075   0.9490   1.9387
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.835907   0.481268 -14.204  < 2e-16 ***
## alcohol       0.542953   0.037720  14.394  < 2e-16 ***
## sulphates     1.639060   0.217233   7.545 4.52e-14 ***
## total_sulfur -0.009315   0.001384  -6.732 1.67e-11 ***
## fix_acidity   0.027351   0.021284   1.285    0.199
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1777.5  on 1594  degrees of freedom
## AIC: 1787.5
##
## Number of Fisher Scoring iterations: 18
```

```
# The logit model performed better with the lower AIC value
```

Compare best logit model with AIC

```
library("AICcmodavg")
```

```
##
## Attaching package: 'AICcmodavg'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
models <- list(randomforestmodlogit, cartmodlogit)
mod.names <- c('RandomForest', 'Cart')
aictab(cand.set = models, modnames = mod.names)
```

```
##
## Model selection based on AICc:
##
##              K    AICc Delta_AICc AICcWt Cum.Wt      LL
## RandomForest 5 1694.21       0.00      1      1 -842.09
## Cart         5 1762.56      68.35      0      1 -876.26
```

```
# The random forest logit model performed the best
```

Compare best model with BIC

```
library("flexmix")
BIC(randomforestmodlogit)
```

```
## [1] 1721.058
```

```
BIC(randomforestmodcloglog)
```

```
## [1] 1737.845
```

```
BIC(cartmodlogit)
```

```
## [1] 1789.404
```

```
BIC(cartmodcloglog)
```

```
## [1] 1814.418
```

```
# The random forest logit model performed the best
```