Logistic_model

Chloe Florence and Tiffany Le

2025-09-26

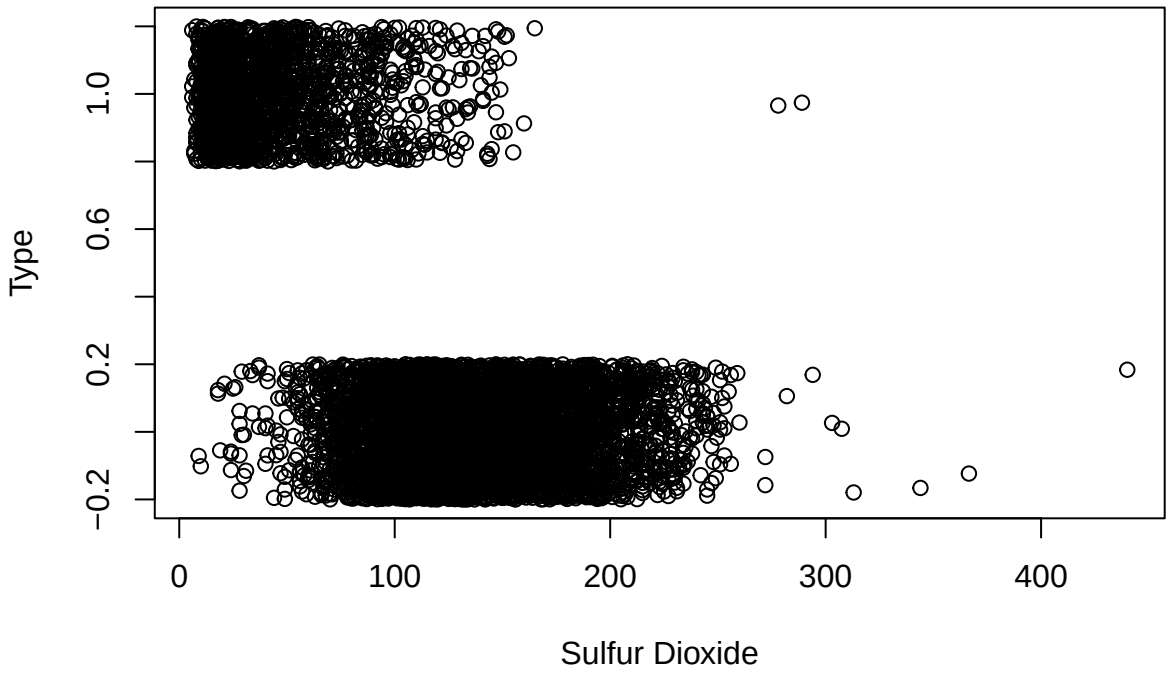
Importing Data

```
library(dplyr)
library(class)
library(leaps)
# creating binomial predictor for type of wine
red_wine <- read.csv("winequality-red.csv", header = TRUE, sep = ";")
red_wine["type"] = 1
white_wine <- read.csv("winequality-white.csv", header = TRUE, sep = ";")
white_wine["type"] = 0
# creating binomial predictor for if a wine is good or not.
wine <- bind_rows(red_wine, white_wine)
wine$good <- ifelse(wine$quality > 5, 1, 0)
```

Data Exploration

Plots

```
plot(wine$total.sulfur.dioxide, jitter(wine$type), xlab = "Sulfur Dioxide", ylab = "Type")
```

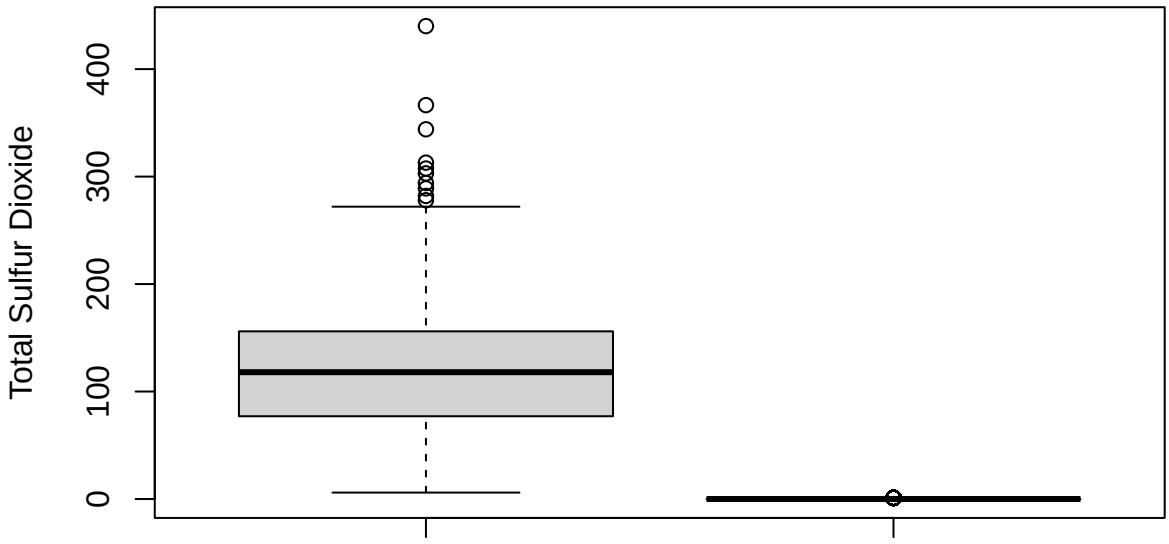


```r
# looking at correlation of various predictors
cor(wine)
```

```
##                       fixed.acidity volatile.acidity citric.acid residual.sugar
## fixed.acidity           1.00000000       0.21900826  0.32443573    -0.11198128
## volatile.acidity        0.21900826       1.00000000 -0.37798132    -0.19601117
## citric.acid             0.32443573      -0.37798132  1.00000000     0.14245123
## residual.sugar         -0.11198128      -0.19601117  0.14245123     1.00000000
## chlorides               0.29819477       0.37712428  0.03899801    -0.12894050
## free.sulfur.dioxide    -0.28273543      -0.35255731  0.13312581     0.40287064
## total.sulfur.dioxide   -0.32905390      -0.41447619  0.19524198     0.49548159
## density                 0.45890998       0.27129565  0.09615393     0.55251695
## pH                     -0.25270047       0.26145440 -0.32980819    -0.26731984
## sulphates               0.29956774       0.22598368  0.05619730    -0.18592741
## alcohol                -0.09545152      -0.03764039 -0.01049349    -0.35941477
## quality                -0.07674321      -0.26569948  0.08553172    -0.03698048
## type                    0.48673983       0.65303559 -0.18739650    -0.34882101
## good                   -0.06735375      -0.26704633  0.07573859    -0.03248435
##                        chlorides free.sulfur.dioxide total.sulfur.dioxide
## fixed.acidity         0.29819477         -0.28273543          -0.32905390
## volatile.acidity      0.37712428         -0.35255731          -0.41447619
## citric.acid           0.03899801          0.13312581           0.19524198
## residual.sugar       -0.12894050          0.40287064           0.49548159
## chlorides             1.00000000         -0.19504479          -0.27963045
## free.sulfur.dioxide  -0.19504479          1.00000000           0.72093408
## total.sulfur.dioxide -0.27963045          0.72093408           1.00000000
## density               0.36261466          0.02571684           0.03239451
## pH                    0.04470798         -0.14585390          -0.23841310
## sulphates             0.39559331         -0.18845725          -0.27572682
## alcohol              -0.25691558         -0.17983843          -0.26573964
## quality              -0.20066550          0.05546306          -0.04138545
## type                  0.51267825         -0.47164366          -0.70035716
## good                 -0.18190812          0.04481948          -0.04758506
##                           density          pH    sulphates      alcohol
## fixed.acidity         0.45890998 -0.25270047  0.299567744 -0.095451523
## volatile.acidity      0.27129565  0.26145440  0.225983680 -0.037640386
## citric.acid           0.09615393 -0.32980819  0.056197300 -0.010493492
## residual.sugar        0.55251695 -0.26731984 -0.185927405 -0.359414771
## chlorides             0.36261466  0.04470798  0.395593307 -0.256915580
## free.sulfur.dioxide   0.02571684 -0.14585390 -0.188457249 -0.179838435
## total.sulfur.dioxide  0.03239451 -0.23841310 -0.275726820 -0.265739639
## density               1.00000000  0.01168608  0.259478495 -0.686745422
## pH                    0.01168608  1.00000000  0.192123407  0.121248467
## sulphates             0.25947850  0.19212341  1.000000000 -0.003029195
## alcohol              -0.68674542  0.12124847 -0.003029195  1.000000000
## quality              -0.30585791  0.01950570  0.038485446  0.444318520
## type                  0.39064532  0.32912865  0.487217970 -0.032969551
## good                 -0.26887620  0.01884228  0.035807167  0.394675617
##                         quality        type        good
## fixed.acidity        -0.07674321  0.48673983 -0.06735375
## volatile.acidity     -0.26569948  0.65303559 -0.26704633
## citric.acid           0.08553172 -0.18739650  0.07573859
## residual.sugar       -0.03698048 -0.34882101 -0.03248435
## chlorides            -0.20066550  0.51267825 -0.18190812
```

```
## free.sulfur.dioxide   0.05546306 -0.47164366  0.04481948
## total.sulfur.dioxide -0.04138545 -0.70035716 -0.04758506
## density              -0.30585791  0.39064532 -0.26887620
## pH                    0.01950570  0.32912865  0.01884228
## sulphates             0.03848545  0.48721797  0.03580717
## alcohol               0.44431852 -0.03296955  0.39467562
## quality               1.00000000 -0.11932328  0.81448366
## type                 -0.11932328  1.00000000 -0.11659486
## good                  0.81448366 -0.11659486  1.00000000
```

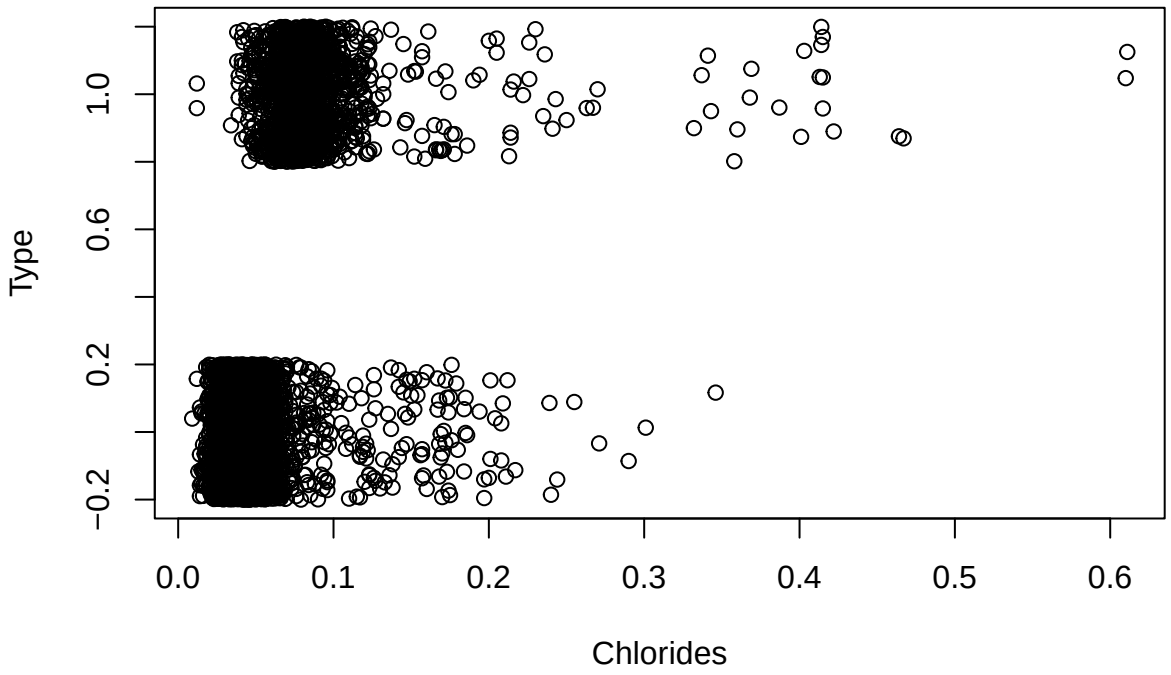
```
boxplot(wine$total.sulfur.dioxide, wine$type, xlab = "Type of Wine", ylab = "Total Sulfur Dioxide")
```




```
# looking for possible outliers
wine[wine[,"total.sulfur.dioxide"] > 400,]
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 6345           6.1             0.26        0.25            2.9     0.047
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 6345                 289                  440 0.99314 3.44      0.64    10.5
##      quality type good
## 6345       3    0    0
```

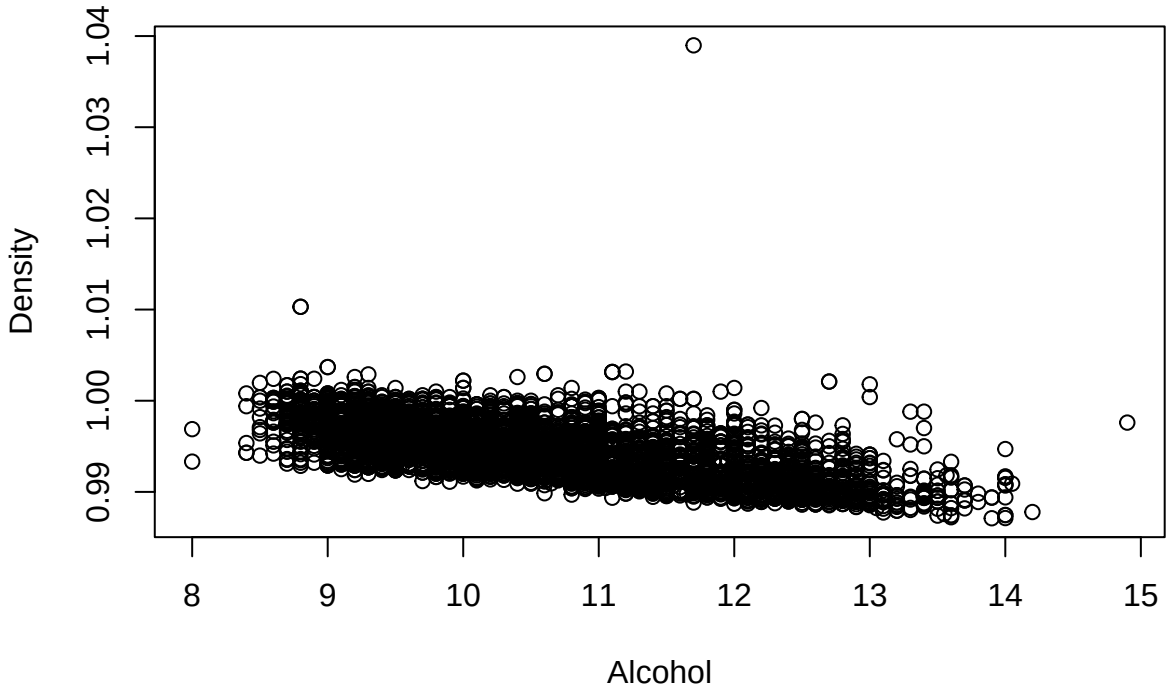
```
plot(wine$chlorides, jitter(wine$type), xlab = "Chlorides", ylab = "Type")
```



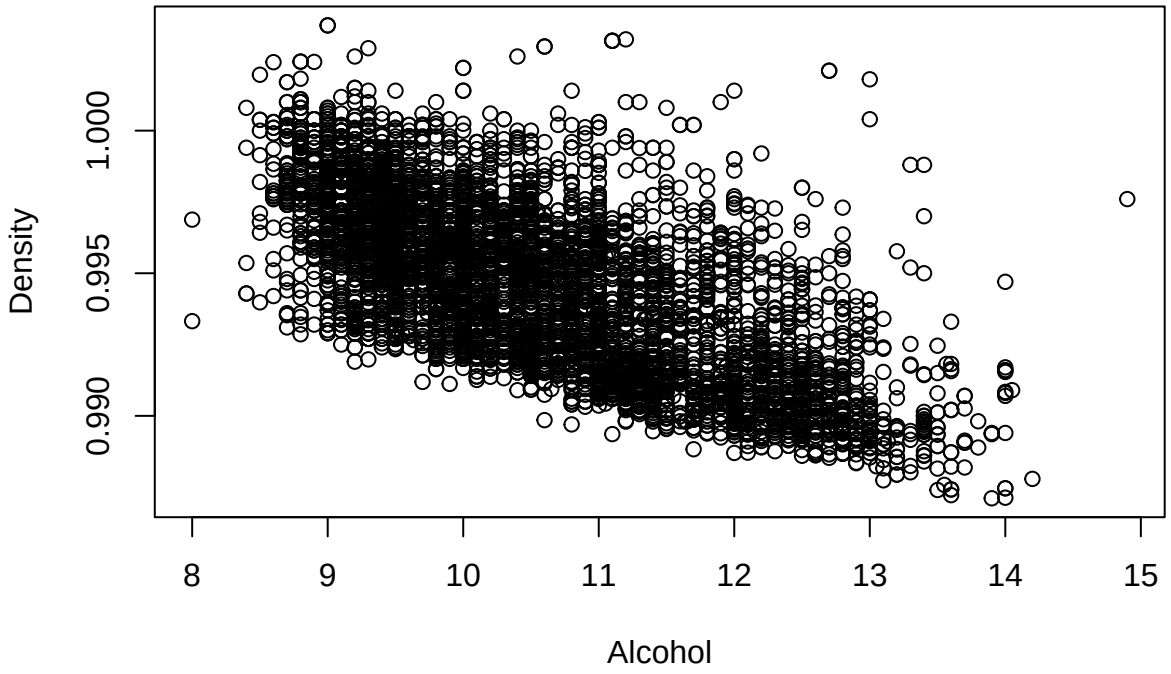

```
# possible outliers
wine[wine[, "chlorides"] > 0.6,]
```

```
##     fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 152           9.2             0.52        1.00            3.4     0.610
## 259           7.7             0.41        0.76            1.8     0.611
##     free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 152                  32                   69  0.9996 2.74      2.00     9.4
## 259                   8                   45  0.9968 3.06      1.26     9.4
##     quality type good
## 152       4    1    0
## 259       5    1    0
```

```
# evaluating effect of outliers
plot(wine$alcohol, wine$density, xlab = "Alcohol", ylab = "Density")
```




```
wo_outlier <- wine[wine["density"] < 1.01,]
plot(wo_outlier$alcohol, wo_outlier$density, xlab = "Alcohol", ylab = "Density")
```




```
wine[wine["density"] > 1.01,]
```

```
##      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 3253           7.9            0.330        0.28           31.6     0.053
## 3263           7.9            0.330        0.28           31.6     0.053
## 4381           7.8            0.965        0.60           65.8     0.074
##      free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 3253                  35                  176 1.01030 3.15      0.38     8.8
```

```
## 3263                  35                  176 1.01030 3.15      0.38     8.8
## 4381                   8                  160 1.03898 3.39      0.69    11.7
##      quality type good
## 3253       6    0    1
## 3263       6    0    1
## 4381       6    0    1
```

Summary Statistics

```
summary(wine)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
##       type             good
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :1.0000
##  Mean   :0.2461   Mean   :0.6331
##  3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

Logistic Model

```
set.seed(1)
train <- sample(6497, 5198)
# Logistic full model
log_mod <- glm(good ~ . - quality, data = wine, family = binomial, subset = train)
summary(log_mod)
```

```
##
## Call:
## glm(formula = good ~ . - quality, family = binomial, data = wine,
##     subset = train)
##
```

```
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.443e+02  4.907e+01   2.941  0.00327 **
## fixed.acidity         1.193e-01  5.587e-02   2.136  0.03272 *
## volatile.acidity     -4.618e+00  3.263e-01 -14.152  < 2e-16 ***
## citric.acid          -3.759e-01  2.863e-01  -1.313  0.18920
## residual.sugar        1.284e-01  2.106e-02   6.098 1.07e-09 ***
## chlorides            -1.199e+00  1.146e+00  -1.047  0.29527
## free.sulfur.dioxide   1.201e-02  2.818e-03   4.262 2.03e-05 ***
## total.sulfur.dioxide -5.621e-03  1.172e-03  -4.795 1.62e-06 ***
## density              -1.561e+02  4.990e+01  -3.128  0.00176 **
## pH                    8.970e-01  3.307e-01   2.712  0.00669 **
## sulphates             2.058e+00  2.956e-01   6.961 3.38e-12 ***
## alcohol               7.705e-01  6.638e-02  11.607  < 2e-16 ***
## type                  5.750e-01  2.097e-01   2.742  0.00611 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6825.3  on 5197  degrees of freedom
## Residual deviance: 5371.1  on 5185  degrees of freedom
## AIC: 5397.1
##
## Number of Fisher Scoring iterations: 4
```

```r
# best predictors for each number of predictors

regfit.fwd = regsubsets(good ~ . - quality,data = wine, subset = train)
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(good ~ . - quality, data = wine, subset = train)
## 12 Variables  (and intercept)
##                       Forced in Forced out
## fixed.acidity             FALSE      FALSE
## volatile.acidity          FALSE      FALSE
## citric.acid               FALSE      FALSE
## residual.sugar            FALSE      FALSE
## chlorides                 FALSE      FALSE
## free.sulfur.dioxide       FALSE      FALSE
## total.sulfur.dioxide      FALSE      FALSE
## density                   FALSE      FALSE
## pH                        FALSE      FALSE
## sulphates                 FALSE      FALSE
## alcohol                   FALSE      FALSE
## type                      FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1  ( 1 ) " "           " "              " "         " "            " "
## 2  ( 1 ) " "           "*"              " "         " "            " "
## 3  ( 1 ) " "           "*"              " "         " "            " "
## 4  ( 1 ) " "           "*"              " "         "*"            " "
## 5  ( 1 ) " "           "*"              " "         "*"            " "
```

```
## 6  ( 1 ) " "           "*"              " "          "*"            " "
## 7  ( 1 ) " "           "*"              " "          "*"            " "
## 8  ( 1 ) " "           "*"              " "          "*"            "*"
##          free.sulfur.dioxide total.sulfur.dioxide density pH  sulphates alcohol
## 1  ( 1 ) " "                 " "                  " "     " " " "       "*"
## 2  ( 1 ) " "                 " "                  " "     " " " "       "*"
## 3  ( 1 ) " "                 " "                  " "     " " "*"       "*"
## 4  ( 1 ) " "                 " "                  " "     " " "*"       "*"
## 5  ( 1 ) " "                 "*"                  " "     " " "*"       "*"
## 6  ( 1 ) "*"                 "*"                  " "     " " "*"       "*"
## 7  ( 1 ) "*"                 "*"                  " "     "*" "*"       "*"
## 8  ( 1 ) "*"                 "*"                  " "     "*" "*"       "*"
##          type
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) " "
## 6  ( 1 ) " "
## 7  ( 1 ) " "
## 8  ( 1 ) " "
```

```
# Best model through Best Subset Selection with BIC criteria
library(bestglm)
model <- bestglm(wine[train,-12], IC="BIC")
model
```

```
## BIC
## BICq equivalent for q in (5.45321160237977e-05, 0.635151358655417)
## Best Model:
##                          Estimate   Std. Error    t value      Pr(>|t|)
## (Intercept)          -0.913098797 0.0718819304 -12.702758  1.980067e-36
## volatile.acidity     -0.829760730 0.0395801022 -20.964088  9.725677e-94
## residual.sugar        0.011733475 0.0014560232   8.058576  9.499706e-16
## free.sulfur.dioxide   0.002507038 0.0004718694   5.312991  1.123437e-07
## total.sulfur.dioxide -0.001324350 0.0001655762  -7.998434  1.541554e-15
## sulphates             0.305566994 0.0407209526   7.503925  7.246010e-14
## alcohol               0.160061285 0.0053200272  30.086554 1.865151e-183
```

finding the test error for logistic regression

```
test<- wine[-train,]
glm.probs <- predict(log_mod, test , type = "response")

glm.pred <- rep(0, 1299)
glm.pred[glm.probs > .5] <- 1

table(glm.pred, test$good)
```

```
##
## glm.pred   0   1
##        0 278 120
##        1 206 695
```

```
mean(glm.pred != test$good)
```

```
## [1] 0.2509623
```

The test error in this case was 25.096%

Cross Validation with the logistic model

We will be training the data on 4/5ths of the data and then using the remaining 1/5 to test the data. The cross validation error will be the mean of the 5 different test errors we will get.

```
# getting the 5 different samples
set.seed(1)
a <- sample(1:6497, 1299)
b <- sample(setdiff(1:6497, a), 1299)
c <- sample(setdiff(1:6497, c(a, b)), 1299)
d <- sample (setdiff(1:6497, c(a, b, c)), 1300)
e <- setdiff(1:6497, c(a, b, c, d))
```

```
library(pROC)
par(mfrow=c(2,3))
samples <- list(a = a,b = b,c = c,d = d,e = e)
error <- c()
for (i in 1:5){
  # model for this training data set
  test <- samples[[i]]
  train <- setdiff(1:nrow(wine), test)
  log_mod <- glm(good ~ volatile.acidity + residual.sugar + free.sulfur.dioxide + total.sulfur.dioxide +
  glm.probs <- predict(log_mod, wine[test,], type = "response")

  glm.pred <- ifelse(glm.probs > 0.5, 1, 0)

  actual <- wine$good[test]
  new_err <- mean(glm.pred != actual)

  error <- c(error, new_err)

  roc_obj <- roc(actual, glm.pred)
  plot(roc_obj, main = "ROC Curve",xlab = "False Positive Rate (1 - Specificity)",
       ylab = "True Positive Rate (Sensitivity)")
  print(auc(roc_obj))
}
```

```
## Area under the curve: 0.7271
```

```
## Area under the curve: 0.6968
```

```
## Area under the curve: 0.7015
```

```
## Area under the curve: 0.6851
```

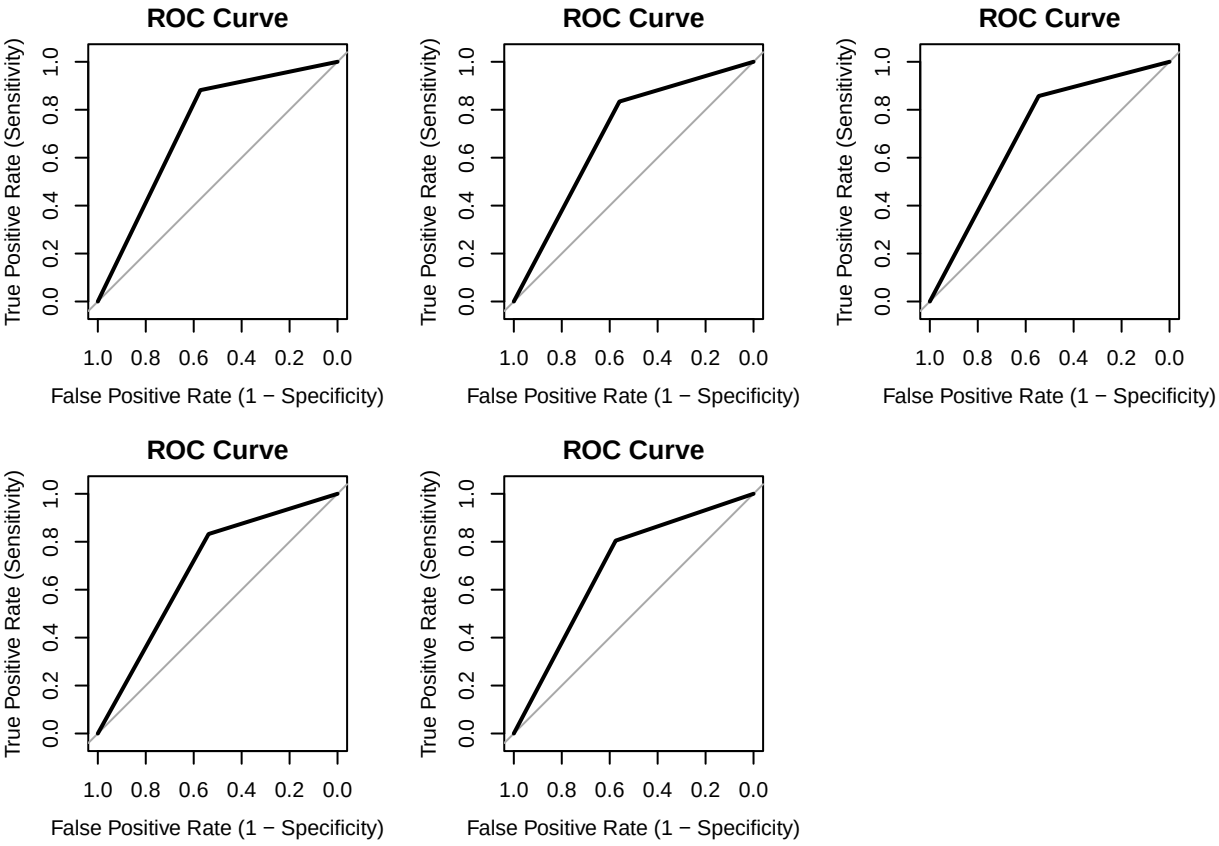
```
## Area under the curve: 0.6898
```

```
error
```

```
## [1] 0.2301771 0.2671286 0.2578907 0.2784615 0.2776923
```

```
mean(error)
```

```
## [1] 0.26227
```



accuracy = 1 - 0.26227 = 0.73773 ~ 73.77% accurate

Precision Average = 0.766588708513

This means on average 76.65% of the true positive are correct out of the total predicted positive.

Recall: 0.841987326462

This means on average 84.12% of the true positive are correct out of the total true positive.

Full Logistic Model

```
full_mod <- log_mod <- glm(good ~ volatile.acidity + residual.sugar + free.sulfur.dioxide + total.sulfu
summary(full_mod)
```

```
##
## Call:
## glm(formula = good ~ volatile.acidity + residual.sugar + free.sulfur.dioxide +
##     total.sulfur.dioxide + sulphates + alcohol, family = binomial,
##     data = wine)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -8.7440138  0.4201198 -20.813  < 2e-16 ***
## volatile.acidity     -4.3169708  0.2212239 -19.514  < 2e-16 ***
## residual.sugar        0.0633281  0.0074512   8.499  < 2e-16 ***
## free.sulfur.dioxide   0.0168408  0.0024883   6.768 1.31e-11 ***
## total.sulfur.dioxide -0.0076431  0.0008395  -9.105  < 2e-16 ***
## sulphates             1.8751919  0.2263165   8.286  < 2e-16 ***
```

```
## alcohol               0.9518199  0.0339175  28.063  < 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: 8541.0  on 6496  degrees of freedom
## Residual deviance: 6723.1  on 6490  degrees of freedom
## AIC: 6737.1
##
## Number of Fisher Scoring iterations: 4
```

KNN Algorithm

```
# knn error with a as test set
train <- c(b, c, d, e)

knn_a <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residual

table(knn_a, wine$good[a])
```

```
##
## knn_a   0   1
##     0 284 160
##     1 186 669
```

```
error_a <- mean(knn_a != wine$good[a]); error_a
```

```
## [1] 0.2663587
```

```
# knn error with b as test set
train <- c(a, c, d, e)

knn_b <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residual

table(knn_b, wine$good[b])
```

```
##
## knn_b   0   1
##     0 289 161
##     1 190 659
```

```
error_b <- mean(knn_b != wine$good[b]); error_b
```

```
## [1] 0.2702079
```

```
# knn error with c as test set
train <- c(a, b, d, e)

knn_c <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residual

table(knn_c, wine$good[c])
```

```
##
## knn_c   0   1
##     0 301 170
```

```
##     1 179 649
```

```
error_c <- mean(knn_c != wine$good[c]); error_c
```

```
## [1] 0.2686682
```

```
# knn error with d as test set
train <- c(a, b, c, e)

knn_d <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residual

table(knn_d, wine$good[d])
```

```
##
## knn_d   0   1
##     0 321 164
##     1 168 647
```

```
error_d <- mean(knn_d != wine$good[d]); error_d
```

```
## [1] 0.2553846
```

```
# knn error with e as test set
train <- c(a, b, c, d)

knn_e <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residual

table(knn_e, wine$good[e])
```

```
##
## knn_e   0   1
##     0 295 165
##     1 171 669
```

```
error_e <- mean(knn_e != wine$good[e]); error_e
```

```
## [1] 0.2584615
```

```
# mean of all 5 misclassification rates
mean(error_a, error_b, error_c, error_d, error_e)
```

```
## [1] 0.2663587
```

```
wine_knn_cv <- function(k) {
  train <- c(b, c, d, e)

  knn_a <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residua

  error_a <- mean(knn_a != wine$good[a])

  train <- c(a, c, d, e)

  knn_b <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residua

  error_b <- mean(knn_b != wine$good[b])

  train <- c(a, b, d, e)

  knn_c <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residua
```

```
  error_c <- mean(knn_c != wine$good[c])

  train <- c(a, b, c, e)

  knn_d <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residua

  error_d <- mean(knn_d != wine$good[d])

  train <- c(a, b, c, d)

  knn_e <- knn(data.frame(volatile.acidity = wine$volatile.acidity[train], residual.sugar = wine$residua

  error_e <- mean(knn_e != wine$good[e])

  mean(error_a, error_b, error_c, error_d, error_e)
  }
```

KNN error rate for different values of k

```
# misclassification rate when k = 2
wine_knn_cv(2)
```

```
## [1] 0.3156274
```

```
wine_knn_cv(3)
```

```
## [1] 0.3071594
```

```
wine_knn_cv(4)
```

```
## [1] 0.3063895
```

```
wine_knn_cv(5)
```

```
## [1] 0.3010008
```

```
wine_knn_cv(6)
```

```
## [1] 0.3033102
```

```
wine_knn_cv(7)
```

```
## [1] 0.2994611
```

```
wine_knn_cv(8)
```

```
## [1] 0.3110085
```

```
wine_knn_cv(9)
```

```
## [1] 0.3125481
```

```
wine_knn_cv(10)
```

```
## [1] 0.321786
```

```
cv_error <- numeric(100)
for (i in 1:100) {
  cv_error[i] <- wine_knn_cv(i)
```

```
}
min(cv_error)
```

```
## [1] 0.2663587
```

This minimum error rate comes when k = 1

```
par(mfrow = c(2, 3))

plot(roc(wine$good[a], as.numeric(knn_a)), main = "ROC Curve", xlab = "False Positive Rate", ylab =  "T

plot(roc(wine$good[b], as.numeric(knn_b)), main = "ROC Curve", xlab = "False Positive Rate", ylab =  "T

plot(roc(wine$good[c], as.numeric(knn_c)), main = "ROC Curve", xlab = "False Positive Rate", ylab =  "T

plot(roc(wine$good[d], as.numeric(knn_d)), main = "ROC Curve", xlab = "False Positive Rate", ylab =  "T

plot(roc(wine$good[e], as.numeric(knn_e)), main = "ROC Curve", xlab = "False Positive Rate", ylab =  "T

#Finding the area under the curve for the ROC curve.
auc(roc(wine$good[a], as.numeric(knn_a)))
```

```
## Area under the curve: 0.7056
```

```
auc(roc(wine$good[b], as.numeric(knn_b)))
```

```
## Area under the curve: 0.7035
```

```
auc(roc(wine$good[c], as.numeric(knn_c)))
```

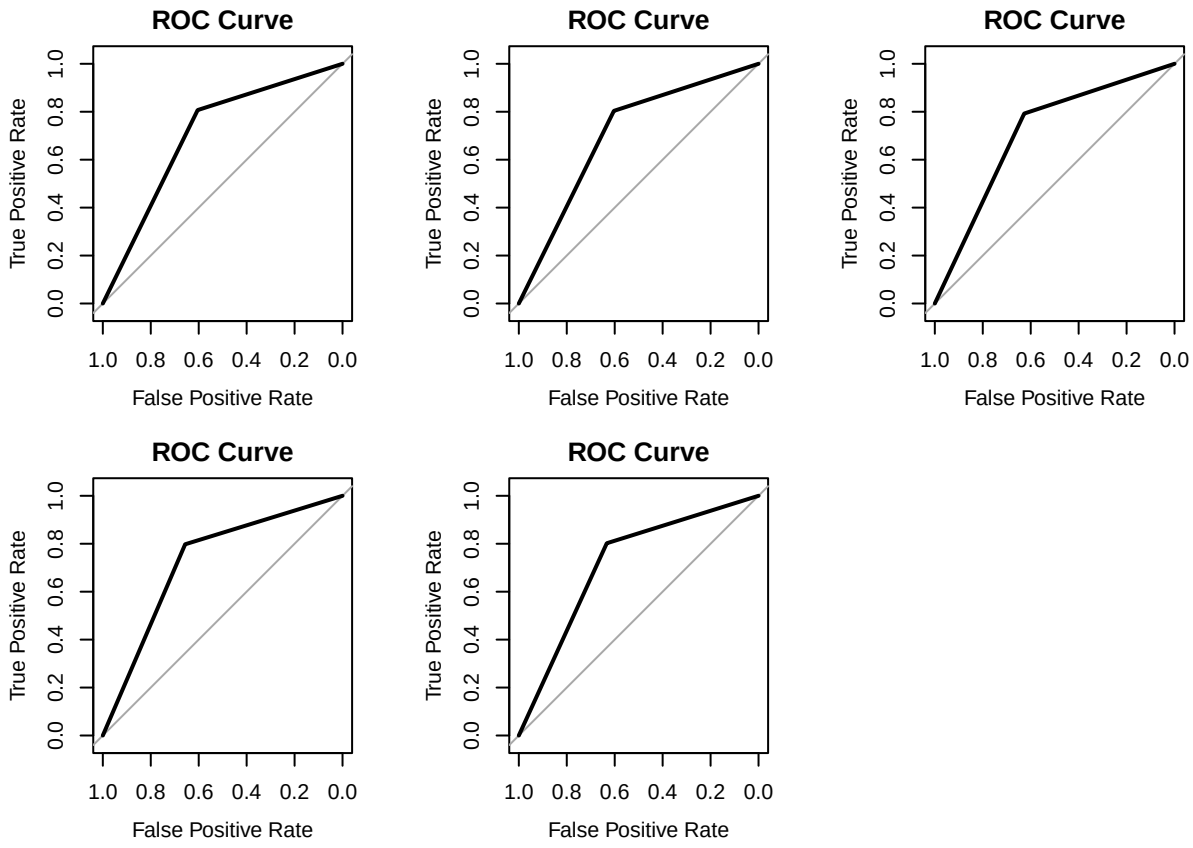
```
## Area under the curve: 0.7098
```

```
auc(roc(wine$good[d], as.numeric(knn_d)))
```

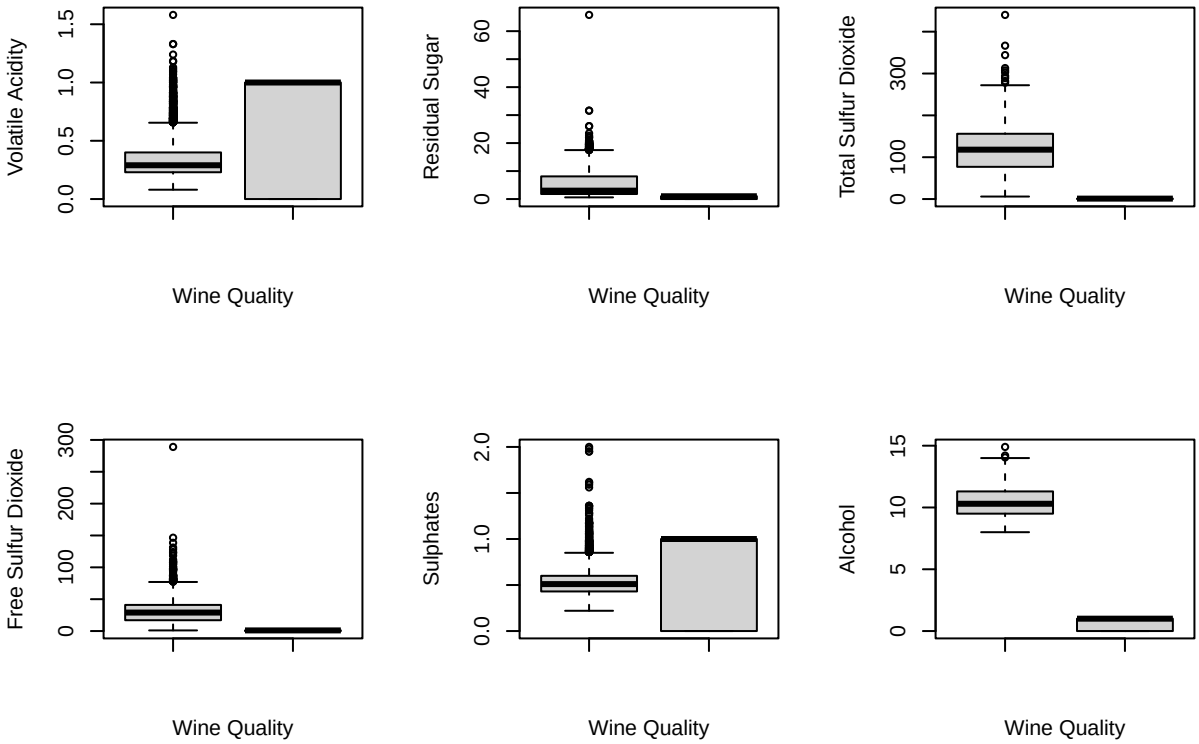
```
## Area under the curve: 0.7271
```

```
auc(roc(wine$good[e], as.numeric(knn_e)))
```

```
## Area under the curve: 0.7176
```




```
par(mfrow = c(2, 3))

boxplot(wine$volatile.acidity, wine$good, ylab = "Volatile Acidity", xlab = "Wine Quality")

boxplot(wine$residual.sugar, wine$good, ylab = "Residual Sugar", xlab = "Wine Quality")

boxplot(wine$total.sulfur.dioxide, wine$good, ylab = "Total Sulfur Dioxide", xlab = "Wine Quality")

boxplot(wine$free.sulfur.dioxide, wine$good, ylab = "Free Sulfur Dioxide", xlab = "Wine Quality")

boxplot(wine$sulphates, wine$good, ylab = "Sulphates", xlab = "Wine Quality")

boxplot(wine$alcohol, wine$good, ylab = "Alcohol", xlab = "Wine Quality")
```


Volatile Acidity
0.0
0.5
1.0
1.5
Wine Quality
Residual Sugar
0
20
40
60
Wine Quality
Total Sulfur Dioxide
0
100
300
Wine Quality
Free Sulfur Dioxide
0
100
200
300
Wine Quality
Sulphates
0.0
1.0
2.0
Wine Quality
Alcohol
0
5
10
15
Wine Quality