Lab 13: Variable Selection and Logistic Regression

Stat 131A

April 25, 2022

Welcome to the Lab 13! In the first part, we will apply variable selection techniques to find the best subset of covariates to predict the red wine quality using physicochemical tests scores such as citric acid, pH, etc.

The dataset we will be using is related to red variants of the Portuguese *vinho verde* wine. There are 1599 samples available in the dataset. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

The explanatory variables are all continuous variables and based on physicochemical tests:

- **fixed acidity**
- **volatile acidity**
- **citric acid**
- **residual sugar**
- **chlorides**
- **free sulfur dioxide**
- **total sulfur dioxide**
- **density**
- **pH**
- **sulphates**
- **alcohol**

The response variable is the **quality** score between 0 and 10 (based on sensory data).

We randomly split the data into two parts-the `wine` dataset with 1199 samples and the `wine.test` dataset with 400 samples. Splitting the dataset is a common technique when we want to evaluate the model performance. There are training set, validation set, and test set. The validation set is used for model selection. That is, to estimate the performance of the different model in order to choose the best one. The test set is used for estimating the performance of our final model.

```
set.seed(20170413)
wine.dataset <- read.csv("winequality-red.csv", sep = ";")
test.samples <- sample(1:nrow(wine.dataset), 400)
wine <- wine.dataset[-test.samples, ]
wine.test <- wine.dataset[test.samples, ]
```

We now fit a linear regression using all of the explanatory variables:

```
wine.fit <- lm(quality ~. ,data = na.omit(wine))
summary(wine.fit)
```

```
##
```

```
## Call:
## lm(formula = quality ~ ., data = na.omit(wine))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69755 -0.35429 -0.03872  0.42375  1.99847
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.093e+01  2.416e+01   0.867   0.3864
## fixed.acidity         3.130e-02  2.980e-02   1.050   0.2938
## volatile.acidity     -1.163e+00  1.373e-01  -8.465  < 2e-16 ***
## citric.acid          -3.944e-01  1.649e-01  -2.392   0.0169 *
## residual.sugar        2.289e-02  1.693e-02   1.351   0.1768
## chlorides            -2.191e+00  4.821e-01  -4.544 6.09e-06 ***
## free.sulfur.dioxide   6.202e-03  2.407e-03   2.576   0.0101 *
## total.sulfur.dioxide -3.471e-03  8.193e-04  -4.237 2.44e-05 ***
## density              -1.699e+01  2.468e+01  -0.688   0.4913
## pH                   -4.129e-01  2.176e-01  -1.898   0.0580 .
## sulphates             1.022e+00  1.311e-01   7.802 1.33e-14 ***
## alcohol               2.860e-01  2.991e-02   9.562  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6314 on 1187 degrees of freedom
## Multiple R-squared:  0.3891, Adjusted R-squared:  0.3834
## F-statistic: 68.72 on 11 and 1187 DF,  p-value: < 2.2e-16
```

Exercise 1: Backward elimination based on p-values

We start with our full model `wine.fit`.

(a) Remove the term corresponding to the coefficient estimate with the highest p-value in the full model. Print the summary of your updated model.

```
# Insert your code here and save your updated model as `wine.backward`
# wine.backward <-
# summary(wine.backward)
```

(b) In `R`, there are functions which automatically perform variable selection. The `step()` function uses AIC, which is very similar to RSS but also takes the number of explanatory variables into account. For example, to do backward elimination starting with our full model:

```
step(wine.fit, direction = "backward")
```

```
## Start:  AIC=-1090.65
## quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
```

```
## - density               1     0.189 473.43 -1092.2
## - fixed.acidity         1     0.440 473.68 -1091.5
## - residual.sugar        1     0.728 473.97 -1090.8
## <none>                              473.24 -1090.7
## - pH                    1     1.436 474.67 -1089.0
## - citric.acid           1     2.281 475.52 -1086.9
## - free.sulfur.dioxide   1     2.646 475.88 -1086.0
## - total.sulfur.dioxide  1     7.156 480.39 -1074.7
## - chlorides             1     8.231 481.47 -1072.0
## - sulphates             1    24.266 497.50 -1032.7
## - volatile.acidity      1    28.567 501.80 -1022.4
## - alcohol               1    36.456 509.69 -1003.7
##
## Step:  AIC=-1092.17
## quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS      AIC
## - fixed.acidity         1     0.275 473.70 -1093.47
## - residual.sugar        1     0.553 473.98 -1092.77
## <none>                              473.43 -1092.17
## - citric.acid           1     2.280 475.71 -1088.41
## - free.sulfur.dioxide   1     2.806 476.23 -1087.08
## - pH                    1     3.257 476.68 -1085.95
## - total.sulfur.dioxide  1     7.456 480.88 -1075.43
## - chlorides             1     8.540 481.97 -1072.73
## - sulphates             1    24.885 498.31 -1032.74
## - volatile.acidity      1    29.700 503.13 -1021.21
## - alcohol               1    94.097 567.52  -876.81
##
## Step:  AIC=-1093.47
## quality ~ volatile.acidity + citric.acid + residual.sugar + chlorides +
##     free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +
##     alcohol
##
##                        Df Sum of Sq    RSS      AIC
## - residual.sugar        1     0.590 474.29 -1093.98
## <none>                              473.70 -1093.47
## - citric.acid           1     2.140 475.84 -1090.07
## - free.sulfur.dioxide   1     2.940 476.64 -1088.05
## - pH                    1     5.910 479.61 -1080.60
## - total.sulfur.dioxide  1     8.930 482.63 -1073.08
## - chlorides             1     9.930 483.63 -1070.60
## - sulphates             1    25.248 498.95 -1033.21
## - volatile.acidity      1    30.044 503.75 -1021.74
## - alcohol               1    94.495 568.20  -877.39
##
## Step:  AIC=-1093.98
## quality ~ volatile.acidity + citric.acid + chlorides + free.sulfur.dioxide +
##     total.sulfur.dioxide + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS      AIC
## <none>                              474.29 -1093.98
```

```
## - citric.acid           1     1.867 476.16 -1091.27
## - free.sulfur.dioxide   1     3.331 477.62 -1087.59
## - pH                    1     5.975 480.27 -1080.97
## - total.sulfur.dioxide  1     8.638 482.93 -1074.34
## - chlorides             1     9.691 483.98 -1071.73
## - sulphates             1    24.867 499.16 -1034.71
## - volatile.acidity      1    29.500 503.79 -1023.63
## - alcohol               1    96.007 570.30  -874.96
```

```
##
## Call:
## lm(formula = quality ~ volatile.acidity + citric.acid + chlorides +
##     free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +
##     alcohol, data = na.omit(wine))
##
## Coefficients:
##          (Intercept)      volatile.acidity           citric.acid
##             4.697601             -1.131930             -0.294894
##            chlorides   free.sulfur.dioxide  total.sulfur.dioxide
##            -2.288882              0.006858             -0.003640
##                   pH             sulphates               alcohol
##            -0.580238              0.994885              0.300961
```

Now try to understand the output of `step()` function. Which variables were omitted from the final model? Provide a list of those variables in order of their elimination, and write the final model.

Variables eliminated (in order):

Final model:

(c) Start from the model with only intercept term. Use the `step()` function to perform forward selection. Write the variables added in order of their addition and the final model.
Hint. (a) Use the `scope` argument in step function. (b) Use `formula()` function to get the formula of your full model.

```
# Insert your code here
```

Variables added (in order):

Final model:

Exercise 2: Regression on all subsets of variables

To find the optimal subset of a certain number of variables for a regression, and to compare between different numbers of variables, use the `regsubsets()` function in the `leaps` package.

```
require(leaps)
```

```
## Loading required package: leaps
```

```
## Warning: package 'leaps' was built under R version 4.1.3
```

```r
regsub_out <- regsubsets(x = wine[, -12] , y = wine[, 12])
```

The default maximum subset size is `nvmax = 8`.

```r
coef(regsub_out, 7)
```

```
##          (Intercept)     volatile.acidity            chlorides
##          4.152458457         -0.992176453         -2.456920286
##  free.sulfur.dioxide total.sulfur.dioxide                   pH
##          0.007546816         -0.003900354         -0.428126506
##            sulphates              alcohol
##          0.975879324          0.292841170
```

Optimal subsets of each size are chosen by RSS. To compare models with different subset sizes, use AIC.

```r
coef(regsub_out, 1:3)
```

```
## [[1]]
## (Intercept)     alcohol
##   1.7135526   0.3758375
##
## [[2]]
##      (Intercept) volatile.acidity          alcohol
##        2.9291785       -1.3732552        0.3285815
##
## [[3]]
##      (Intercept) volatile.acidity        sulphates          alcohol
##        2.4393960       -1.1985154        0.7178874        0.3214194
```

What is the best model using 1 variable? Using 7? Is the optimal model of 7 covariates the same as that found in exercise 1?

Answer here

Exercise 3: Compare performance using test set

Use the test set to assess the performance of the models resulting from forward stepwise selection and the full model. What is the test set root mean square error for the two models?

Answer here

Logistic regression: customer retention

A telecommunications company is concerned about the number of customers leaving their landline business for cable competitors. They need to understand who is leaving. Imagine that you're an analyst at this company and you have to find out who is leaving and why.

We will use data from IBM Watson Analytics to predict customer retention. Analysis of relevant customer data can lead to the design of focused customer retention programs.

The data includes:

- Customers who left within the last month (column `Churn`)
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

```
set.seed(131)
retention <- read.csv("customer_retention.csv", stringsAsFactors = FALSE)
retention$SeniorCitizen <- factor(retention$SeniorCitizen, 0:1, c("No", "Yes"))
retention <- retention[with(
    retention, MultipleLines != "No phone service" &
        OnlineSecurity != "No internet service"), ]
retention$Churn <- as.numeric(factor(retention$Churn, c("No", "Yes"))) - 1
retention$PhoneService = NULL
retention$PaymentMethod = factor(retention$PaymentMethod)
retention <- retention[, -which(names(retention) %in% c("customerID", "PhoneService"))]

test.set <- sample(nrow(retention), 500)
retention.test <- retention[test.set, ]
retention <- retention[-test.set, ]
```

Exercise 4

(a) Fit a logistic regression for `Churn` given all other variables in the dataset.

```
# insert your code here to fit a logistic regression.
```

(b) There are four payment methods available for customers.

```
levels(retention$PaymentMethod)
```

```
## [1] "Bank transfer (automatic)" "Credit card (automatic)"
## [3] "Electronic check"          "Mailed check"
```

While holding other predictors in the model constant, which payment method category is associated with the largest retention probability? Uncomment your answer below (ctrl-shift-c/cmd-shift-c).

Which payment method category is associated with the smallest retention probability? Uncomment your answer below (ctrl-shift-c/cmd-shift-c).

What is the probability difference comparing the payment method category with largest retention probability to that with the smallest? Uncomment your answer below (ctrl-shift-c/cmd-shift-c).

(c) Using your fitted model, generate predictions on the test set `retention.test`. What is the test set prediction accuracy (ie the proportion you got right)?
Hint. Use the `predict()` function with argument `type = "response"` to get the predicted probabilities. When the probability is larger than 0.5, our prediction is 1.

```
# Insert your code here
```

Answer here