

Practice Problems 6

Jordan Lian

3/14/2021

For all of the problems below, keep in mind that feature selection is one of the most difficult issues in model building, particularly regression. We have introduced several automatic feature selection approaches: forward and backward fitting using either p -value, Adjusted R -Squared, or AIC . In addition we also have Principal Component Analysis (PCA). However, in practice you may also need to choose from derived or combined features, *e.g.*, ratios or sums. This makes feature selection a combinatorially problem. In practice, you need to use domain expertise to choose features that you suspect or know contribute to the response variable. So, in this problem, use your domain knowledge and intuition.

R implements backward and forward fitting using AIC with the `step()` function. You may use it for the problems if you wish. Note that AIC will likely produce a model that includes coefficients with a p -value > 0.05 . That is because AIC -based selection is based on adding or eliminating features that reduce the information in the model – it is not based on statistical significance. Also note that elimination is a greedy algorithm and will not produce an optimal model. Like finding an optimal decision tree, finding an optimal set of features is an NP -Complete problem and thus is computationally intractable requiring suboptimal solutions that can be identified in polynomial time.

In addition, just because a model A has a higher Adjusted R -Squared or a lower AIC compared to model B doesn't mean that it is better. Model A may have a lower mean squared error (smaller mean residuals) but that difference in mean error could be due to sampling. Thus, data scientists confirm that model A is really better than model B by running a one-way $ANOVA$ or a t -test that compares mean residuals. In R you can simply use `a <- anova(modelA, modelB)` followed by `summary(a)` to determine if the difference in the model performance is statistically significant.

Compare your answer with those of your peers using the discussion forum.

Problem 1 (60 Points)

Download the [data set on student achievement](#) in secondary education math education of two Portuguese schools (use the data set *Students Math*). Using any packages you wish, complete the following tasks:

```
library(tidyverse)

origin_math <- read_csv('student-mat.csv')

## Parsed with column specification:
## cols(
##   .default = col_character(),
##   age = col_double(),
##   Medu = col_double(),
##   Fedu = col_double(),
##   traveltime = col_double(),
```

```

##   studytime = col_double(),
##   failures = col_double(),
##   famrel = col_double(),
##   freetime = col_double(),
##   goout = col_double(),
##   Dalc = col_double(),
##   Walc = col_double(),
##   health = col_double(),
##   absences = col_double(),
##   G1 = col_double(),
##   G2 = col_double(),
##   G3 = col_double()
## )

## See spec(...) for full column specifications.

math <- origin_math
head(math)

## # A tibble: 6 x 33
##   school sex      age address famsize Pstatus  Medu  Fedu Mjob Fjob reason
##   <chr>  <chr> <dbl> <chr>    <chr>    <chr>  <dbl> <dbl> <chr> <chr> <chr>
## 1 GP     F        18   U       GT3      A          4     4 at_h~ teac~ course
## 2 GP     F        17   U       GT3      T          1     1 at_h~ other course
## 3 GP     F        15   U       LE3      T          1     1 at_h~ other other
## 4 GP     F        15   U       GT3      T          4     2 heal~ serv~ home
## 5 GP     F        16   U       GT3      T          3     3 other other home
## 6 GP     M        16   U       LE3      T          4     3 serv~ other reput~
## # ... with 22 more variables: guardian <chr>, traveltime <dbl>,
## #   studytime <dbl>, failures <dbl>, schoolsup <chr>, famsup <chr>, paid <chr>,
## #   activities <chr>, nursery <chr>, higher <chr>, internet <chr>,
## #   romantic <chr>, famrel <dbl>, freetime <dbl>, goout <dbl>, Dalc <dbl>,
## #   Walc <dbl>, health <dbl>, absences <dbl>, G1 <dbl>, G2 <dbl>, G3 <dbl>

```

1. (10 pts) Create scatter plots and pairwise correlations between *age*, *absences*, *G1*, and *G2* and final grade (*G3*) using the *pairs.panels()* function in R.

```

library(psych)

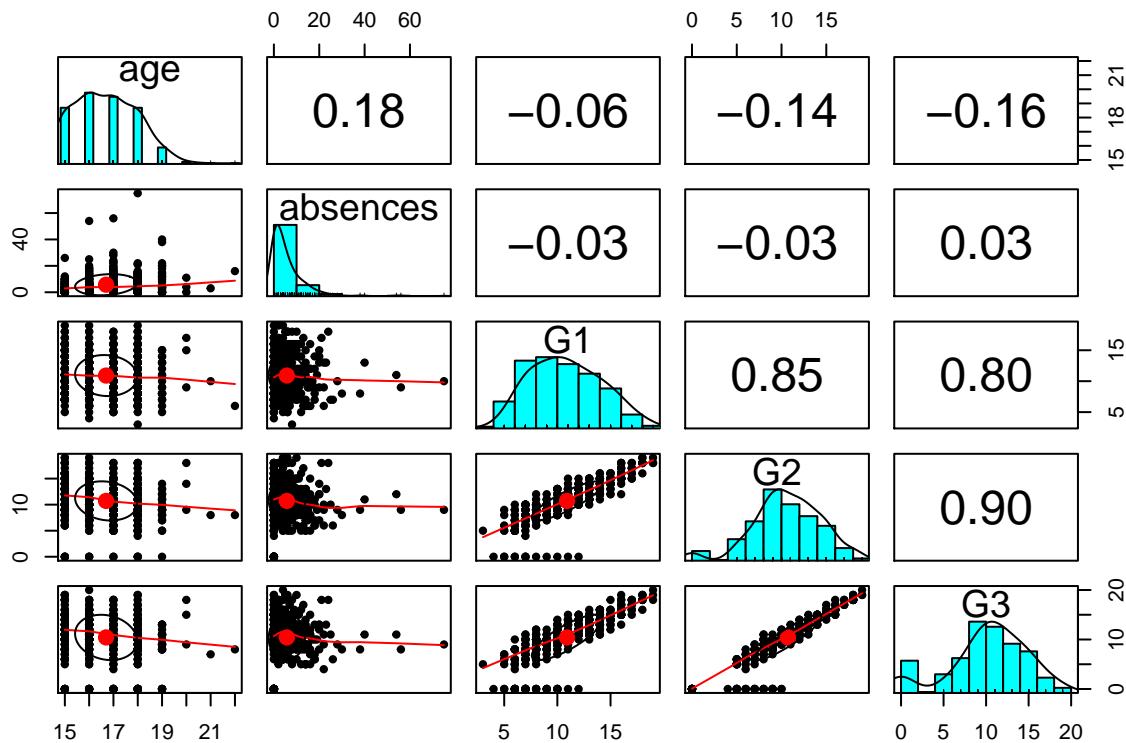
## Warning: package 'psych' was built under R version 4.0.4

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

pairs.panels(math[,c(3, 30:33)])

```



2. (10 pts) Build a multiple regression model predicting final math grade (G3) using as many features as you like but you must use at least four. Include at least one categorical variables and be sure to properly convert it to dummy codes. Select the features that you believe are useful – you do not have to include all features.

I used G1, G2, studytime, activities, absences, and health. I felt that students that did the most outside of school would demonstrate better time management skills, and I figured that previous test scores and study time all would have a strong factor with the final grades in addition to absences and health.

```
# Convert activities from yes/no to 1/0 respectively
math$activities[math$activities == 'yes'] <- 1
math$activities[math$activities == 'no'] <- 0

# Convert activities to factor to show that variable is categorical
math$activities <- factor(math$activities)

# Create model
multi_model <- lm(G3 ~ G1 + G2 + studytime + activities + absences + health, data = math)
multi_model
```

```
##
## Call:
## lm(formula = G3 ~ G1 + G2 + studytime + activities + absences +
##     health, data = math)
##
```

```

## Coefficients:
## (Intercept)          G1          G2    studytime activities1    absences
## -2.10875      0.16035      0.99287     -0.12517     -0.26982      0.03592
##   health
##   0.09120

```

3. (20 pts) Using the model from (2), use stepwise backward elimination to remove all non-significant variables and then state the final model as an equation. State the backward elimination measure you applied (p -value, AIC, Adjusted R²). This [tutorial shows how to use various feature elimination techniques](#).

```
library(MASS)
```

```

##
## Attaching package: 'MASS'

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

step_model <- stepAIC(multi_model, direction = "both",
                      trace = FALSE)
summary(step_model)

##
## Call:
## lm(formula = G3 ~ G1 + G2 + activities + absences, data = math)
##
## Residuals:
##     Min      1Q  Median      3Q      Max
## -9.2233 -0.3684  0.2795  0.9771  3.7877
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.95013   0.35018  -5.569 4.78e-08 ***
## G1          0.15666   0.05555   2.820  0.00504 **
## G2          0.98864   0.04900  20.176 < 2e-16 ***
## activities1 -0.27957   0.19303  -1.448  0.14834
## absences     0.03615   0.01206   2.997  0.00290 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.915 on 390 degrees of freedom
## Multiple R-squared:  0.8271, Adjusted R-squared:  0.8253
## F-statistic: 466.5 on 4 and 390 DF,  p-value: < 2.2e-16

```

4. (10 pts) Calculate the 95% confidence interval for a prediction – you may choose any data you wish for some new student.

```
# For G3 scores
t.test(math$G3)
```

```

## 
##  One Sample t-test
##
## data:  math$G3
## t = 45.182, df = 394, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   9.961992 10.868388
## sample estimates:
## mean of x
## 10.41519

```

5. (10 pts) What is the RMSE for this model – use the entire data set for both training and validation. You may find the [residuals\(\) function](#) useful. Alternatively, you can inspect the model object, e.g., if your model is in the variable *m*, then the residuals (errors) are in *m\$residuals* and your predicted values (fitted values) are in *m\$fitted.values*.

```

# Use regression and compare to actual data
predictions <- multi_model %>% predict(math)

# RMSE
library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

## 
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
## 
##     lift

RMSE(predictions, math$G3)

## [1] 1.895044

```

Problem 2 (40 Points)

For this problem, the following [short tutorial might be helpful in interpreting the logistic regression output](#).

1. (5 pts) Using the same data set as in Problem (1), add another column, PF – pass-fail. Mark any student whose final grade is less than 10 as F, otherwise as P and then build a dummy code variable for that new column. Use the new dummy variable column as the response variable.

```

math$PF[math$G3 < 10] <- 'F'
math$PF[math$G3 >= 10] <- 'P'

```

2. (10 pts) Build a binomial logistic regression model classifying a student as passing or failing. Eliminate any non-significant variable using an elimination approach of your choice. Use as many features as you like but you must use at least four – choose the ones you believe are most useful.

```
# Convert new column to factor
math$PF[math$PF == 'F'] <- 0
math$PF[math$PF == 'P'] <- 1
math$PF <- as.factor(math$PF)

# Regression model
mylogit <- glm(PF ~ G3 + G2 + G1 + studytime + absences + health, data = math, family = "binomial")
```

Warning: glm.fit: algorithm did not converge

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

3. (5 pts) State the regression equation.

```
mylogit
```

```
##
## Call: glm(formula = PF ~ G3 + G2 + G1 + studytime + absences + health,
##           family = "binomial", data = math)
##
## Coefficients:
## (Intercept)          G3          G2          G1      studytime      absences
## -407.45606    43.16837   0.12223   -0.20888   -0.24870   -0.05209
##             health
##            -0.19607
##
## Degrees of Freedom: 394 Total (i.e. Null); 388 Residual
## Null Deviance: 500.5
## Residual Deviance: 6.211e-08 AIC: 14
```

```
summary(mylogit)
```

```
##
## Call:
## glm(formula = PF ~ G3 + G2 + G1 + studytime + absences + health,
##       family = "binomial", data = math)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -5.550e-05 -2.100e-08  2.100e-08  2.100e-08  4.053e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.075e+02  6.694e+04 -0.006   0.995
## G3          4.317e+01  7.677e+03  0.006   0.996
## G2          1.222e-01  4.221e+03  0.000   1.000
## G1          -2.089e-01  3.093e+03  0.000   1.000
```

```

## studytime -2.487e-01 5.063e+03 0.000 1.000
## absences -5.209e-02 6.812e+02 0.000 1.000
## health -1.961e-01 2.724e+03 0.000 1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 5.0050e+02 on 394 degrees of freedom
## Residual deviance: 6.2113e-08 on 388 degrees of freedom
## AIC: 14
##
## Number of Fisher Scoring iterations: 25

```

4. (20 pts) What is the accuracy of your model? Use the entire data set for both training and validation.

```

# Test accuracy vs model
accuracy <- mylogit %>% predict(math)

# RMSE
RMSE(accuracy, as.integer(math$PF))

```

```
## [1] 200.6381
```

```
# R-squared
R2(accuracy, as.integer(math$PF))
```

```
## [1] 0.5932288
```

I wanted to use a confusion matrix, but I couldn't get that to work due to issues with levels/references. I used a table, but that was too much to print, and I also used a CrossTable, but again there was too much information to print out. The accuracy was around 50%, which wasn't too great. The RMSE and R^2 values indicate this as well.

Problem 3 (10 Points)

1. (8 pts) Implement the example from the textbook on pages 205 to 217 for the [data set on white wines](#).

Step 1 - collecting data

<http://archive.ics.uci.edu/ml>

Step 2 - exploring and preparing the data

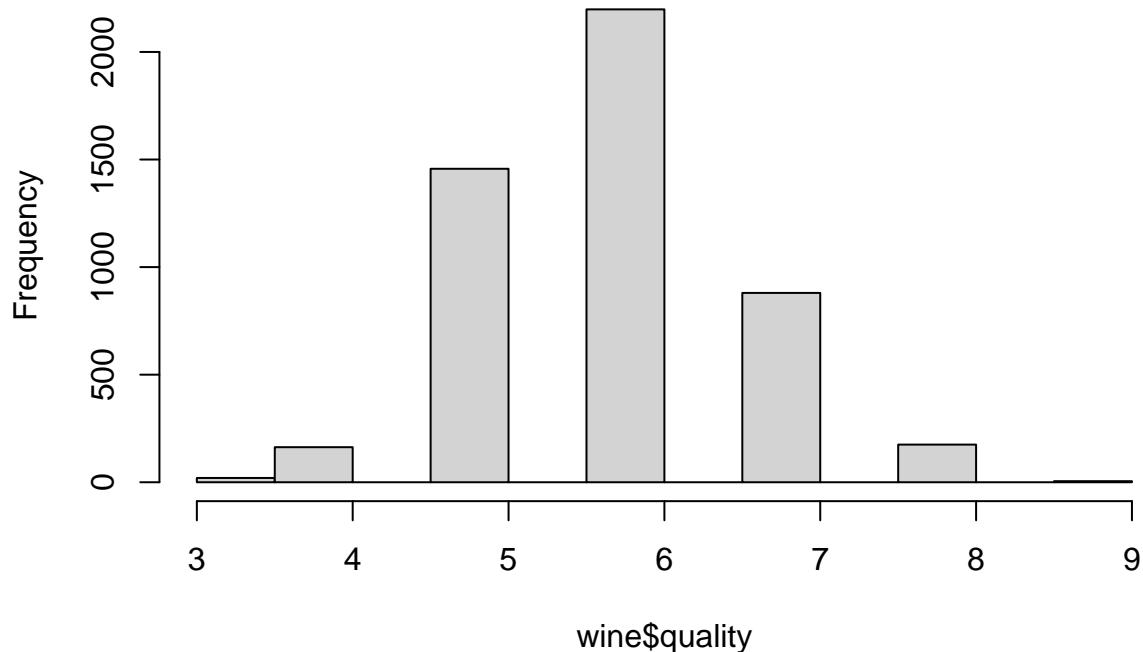
```

# Load dataset
wine <- read.csv("whitewines.csv")

# Histogram
hist(wine$quality)

```

Histogram of wine\$quality



```
# Divide into training/test datasets
wine_train <- wine[1:3750, ]
wine_test <- wine[3751:4898, ]
```

Step 3 - training a model on the data

```
library(rpart)
m.rpart <- rpart(quality ~ ., data = wine_train)
m.rpart

## n= 3750
##
## node), split, n, deviance, yval
##      * denotes terminal node
##
## 1) root 3750 3140.06000 5.886933
##    2) alcohol< 10.85 2473 1510.66200 5.609381
##      4) volatile.acidity>=0.2425 1406 740.15080 5.402560
##        8) volatile.acidity>=0.4225 182 92.99451 4.994505 *
##        9) volatile.acidity< 0.4225 1224 612.34560 5.463235 *
##      5) volatile.acidity< 0.2425 1067 631.12090 5.881912 *
##    3) alcohol>=10.85 1277 1069.95800 6.424432
##      6) free.sulfur.dioxide< 11.5 93 99.18280 5.473118 *
```

```

##      7) free.sulfur.dioxide>=11.5 1184  879.99920 6.499155
##      14) alcohol< 11.85 611  447.38130 6.296236 *
##      15) alcohol>=11.85 573  380.63180 6.715532 *

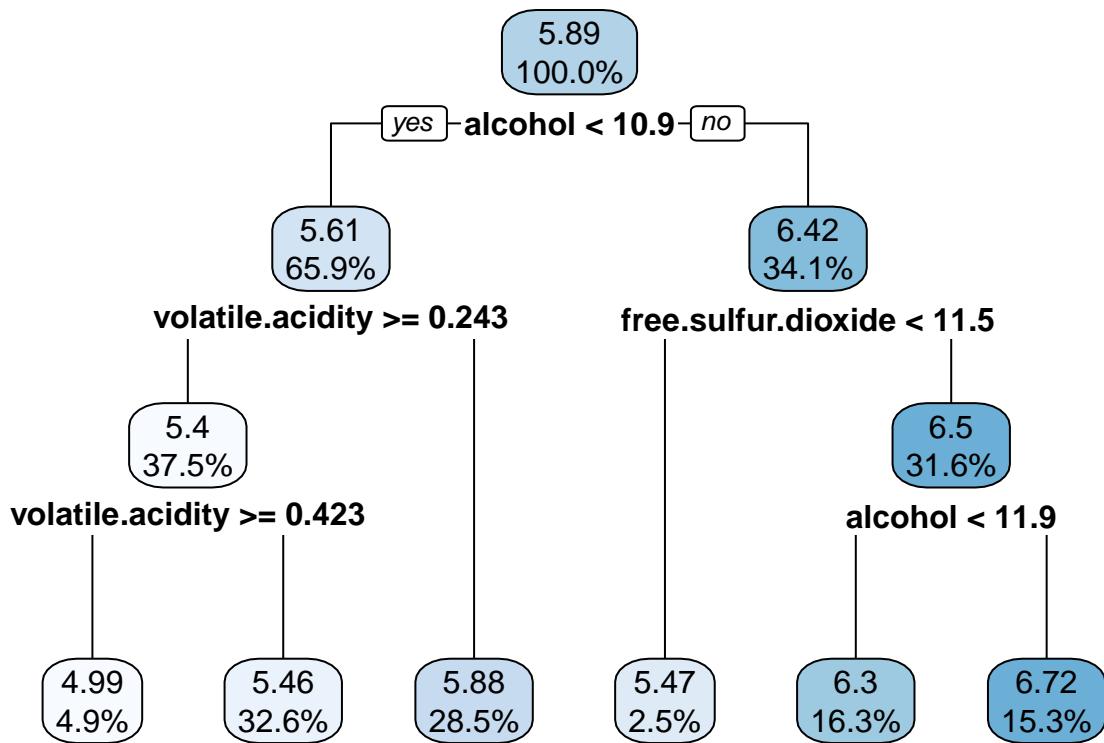
```

Visualizing decision trees

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.4
```

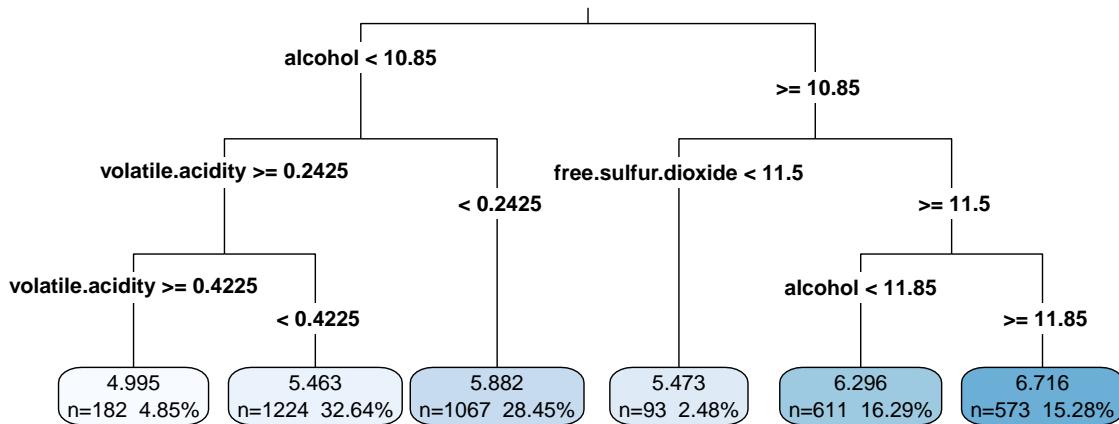
```
rpart.plot(m.rpart, digits = 3)
```



```

# change parameters
rpart.plot(m.rpart, digits = 4, fallen.leaves = TRUE,
           type = 3, extra = 101)

```



Step 4 - evaluating model performance

```
p.rpart <- predict(m.rpart, wine_test)
summary(p.rpart)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##    4.995   5.463   5.882   5.999   6.296   6.716
```

```
summary(wine_test$quality)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##    3.000   5.000   6.000   5.848   6.000   8.000
```

```
cor(p.rpart, wine_test$quality)
```

```
## [1] 0.4931608
```

Measuring performance with the mean absolute error

$$MAE = \frac{1}{n} \sum_{i=1}^n e_i$$

```
MAE <- function(actual, predicted) {  
  mean(abs(actual - predicted))  
}  
MAE(p.rpart, wine_test$quality)
```

```
## [1] 0.5732104
```

```
mean(wine_train$quality)
```

```
## [1] 5.886933
```

```
MAE(5.87, wine_test$quality)
```

```
## [1] 0.5815679
```

Step 5 - improving model performance

```
library(RWeka)
```

```
## Warning: package 'RWeka' was built under R version 4.0.4
```

```
m.m5p <- M5P(quality ~ ., data = wine_train)  
summary(m.m5p)
```

```
##  
## === Summary ===  
##  
## Correlation coefficient           -0.2414  
## Mean absolute error              102.3629  
## Root mean squared error         129.5719  
## Relative absolute error        14704.2234 %  
## Root relative squared error   14159.8116 %  
## Total Number of Instances      3750
```

```
p.m5p <- predict(m.m5p, wine_test)  
summary(p.m5p)
```

```
##      Min. 1st Qu. Median  Mean 3rd Qu.  Max.  
## -539.90 -165.65 -107.07 -112.27 -33.70  32.49
```

```
cor(p.m5p, wine_test$quality)
```

```
## [1] -0.2036594
```

```
MAE(wine_test$quality, p.m5p)
```

```
## [1] 118.6835
```

2. (2 pts) Calculate the RMSE for the model.

```
# get RMSE from summary of m.m5p
names(summary(m.m5p))
```

```
## [1] "string"  "details"
```

```
summary(m.m5p)$details
```

```
##   correlationCoefficient      meanAbsoluteError    rootMeanSquaredError
##             -0.2414045          102.3629346          129.5718906
##   relativeAbsoluteError rootRelativeSquaredError
##             14704.2233575          14159.8115851
```

```
summary(m.m5p)$details[[3]]
```

```
## [1] 129.5719
```