Math 5364

Data Mining 2

Homework 22

Mary Barker

1. Import the data set BIO120.txt, which contains data for 3146 Biol 120 students, including the following variables:

   - Grade: 1 = A, B, or C, 0 = all other grades.
   - Rank: Percentile rank represented as a decimal between 0 and 1, w values close to 1 corresponding to higher ranked students.
   - Math and Verbal: Math and Verbal SAT scores
   - Prev: 1 = student has taken Bio120 before, and 0 = student has not
   - Rdg: Status of student regarding the remedial course Reading 100. possible levels are Never Taken, Concurrently Enrolled, Passed, and Failed.
   - Father's and Mother's education levels
   - Gender

2. Build the best possible logistic regression model for predicting grade based on the other variables.

   (a) Divide the data set into two parts for the purpose of cross-validation.

   (b) Fit a univariate model regressing grade onto each of the other variables. For the quantitative variables, attempt to determine if higher order terms are needed using the groupplot function (see LogisticRegressionFunctions.txt for some helpeful functions). As with linear regression, you can use a likelihood ratio test to formally test whether these terms are needed (LRtest function) For the categorical variables, a univariate model can help to determine if some of the levels can be grouped together to create a variable with fewer levels. This is essential for the

father and mother variables which have 8 levels. It is likely that a stepwise regression will eliminate one of the parent's education variables, since they are highly correlated and have a large number of parameters.

```
basicmodel <- glm(grade~., data = bio.train, family=binomial)
summary(basicmodel)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1729  -0.9761   0.4209   0.9404   2.4381

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -6.279e+00  8.048e-01  -7.802 6.11e-15 ***
rank                             3.820e+00  2.920e-01  13.084  < 2e-16 ***
math                             4.418e-03  7.967e-04   5.545 2.93e-08 ***
verbal                           2.308e-03  7.003e-04   3.296 0.000981 ***
prevTRUE                        -7.196e-01  1.791e-01  -4.017 5.89e-05 ***
rdgFailed                       -1.157e+01  3.090e+02  -0.037 0.970135
rdgNever                         3.681e-01  4.341e-01   0.848 0.396496
rdgPassed                        2.315e-01  5.463e-01   0.424 0.671694
fatherBachelor Degree            1.359e-01  4.586e-01   0.296 0.767032
fatherGraduate/Professional degree  2.164e-01  4.828e-01   0.448 0.653935
fatherHigh School Diploma or GED    4.939e-02  4.541e-01   0.109 0.913382
fatherNo High School             4.840e-01  6.525e-01   0.742 0.458206
fatherNot Available             -2.809e-01  5.770e-01  -0.487 0.626424
fatherSome College               9.147e-02  4.529e-01   0.202 0.839952
fatherSome High School          -1.102e-01  5.213e-01  -0.211 0.832614
motherBachelor Degree            4.431e-01  4.287e-01   1.034 0.301333
motherGraduate/Professional degree  2.566e-01  4.499e-01   0.570 0.568527
motherHigh School Diploma or GED   -1.404e-01  4.249e-01  -0.330 0.741084
motherNo High School             1.808e-01  6.666e-01   0.271 0.786211
motherNot Available              6.378e-01  5.705e-01   1.118 0.263610
motherSome College               1.910e-01  4.203e-01   0.455 0.649449
motherSome High School          -8.138e-02  5.206e-01  -0.156 0.875778
genderMale                      -9.647e-02  1.025e-01  -0.941 0.346671
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3051.4  on 2201  degrees of freedom
Residual deviance: 2562.6  on 2179  degrees of freedom
AIC: 2608.6

Number of Fisher Scoring iterations: 12
```

- Rank

  ```
  rank.model1 = glm(grade~rank, data=bio.train, family=binomial)
  summary(rank.model1)
  ```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8418  -1.0489   0.6307   0.9697   2.2716

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.0440     0.1878  -16.20   <2e-16 ***
rank          4.6711     0.2727   17.13   <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: 3051.4  on 2201  degrees of freedom
Residual deviance: 2687.5  on 2200  degrees of freedom
AIC: 2691.5

Number of Fisher Scoring iterations: 4
```

```
LRtest(rank.model1, basicmodel)
```

The result for the LR test was 1.110223e-16.

In order to check whether high order terms are necessary, first we will generate logit plots to view the curvature.

Figure 1 Logit plot for rank and grade with degree 1

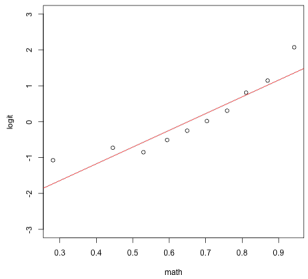
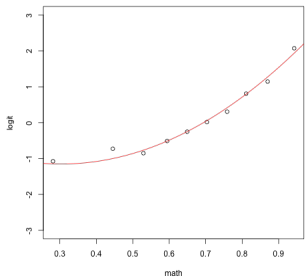


Figure 2 Logit plot for rank and grade with degree 2



```
rank.model2 = glm(grade~rank+I(rank^2), data=bio.train, family=binomial)
summary(rank.model2)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1572  -0.9541   0.4438   1.0005   1.6892

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4556     0.3833  -1.188 0.234666
rank         -4.5914     1.3071  -3.513 0.000444 ***
I(rank^2)     7.5659     1.0818   6.994 2.67e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3051.4  on 2201  degrees of freedom
Residual deviance: 2641.5  on 2199  degrees of freedom
AIC: 2647.5

Number of Fisher Scoring iterations: 4
```

```
LRtest(rank.model2, basicmodel)
```

The result for the LR test was 6.023138e-09.

```
rank.model3 = glm(grade~rank+I(rank^2)+I(rank^3), data=bio.train, family=bin
summary(rank.model3)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3306  -0.9562   0.3580   1.0410   1.8281

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.0310     0.7458  -2.723  0.00646 **
rank          6.3416     4.4184   1.435  0.15121
I(rank^2)   -13.9689     8.2310  -1.697  0.08967 .
I(rank^3)    12.7633     4.8036   2.657  0.00788 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3051.4  on 2201  degrees of freedom
Residual deviance: 2633.9  on 2198  degrees of freedom
AIC: 2641.9

Number of Fisher Scoring iterations: 4
```

```
LRtest(rank.model3, basicmodel)
```

The result for the LR test was 5.600436e-08.

```
rank.model4 = glm(grade~rank+I(rank^2)+I(rank^3)+I(rank^4), data=bio.train,
```

```
summary(rank.model4)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2586  -0.9405   0.3932   1.0283   1.9073

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.040      1.309  -2.323   0.0202 *
rank          16.980     11.695   1.452   0.1465
I(rank^2)    -49.089     36.027  -1.363   0.1730
I(rank^3)     58.656     45.631   1.285   0.1986
I(rank^4)    -20.698     20.335  -1.018   0.3087
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3051.4  on 2201  degrees of freedom
Residual deviance: 2632.9  on 2197  degrees of freedom
AIC: 2642.9

Number of Fisher Scoring iterations: 4
```

```
LRtest(rank.model4, basicmodel)
```

The result for the LR test was 4.13323e-08.

- Math

```
math.model1 = glm(grade~math, data=bio.train, family=binomial)
```

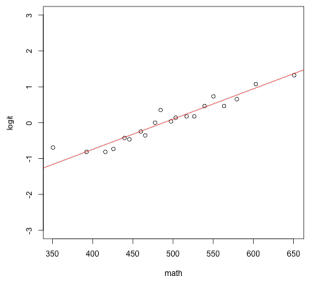
```
summary(math.model1)
```

```
Call:
glm(formula = grade ~ math, family = binomial, data = bio.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0446  -1.0783   0.6487   1.1001   1.8840

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.1227015  0.3279069  -12.57   <2e-16 ***
math         0.0084458  0.0006594   12.81   <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: 3051.4  on 2201  degrees of freedom
Residual deviance: 2865.9  on 2200  degrees of freedom
AIC: 2869.9

Number of Fisher Scoring iterations: 4
```

```
LRtest(math.model1, basicmodel)
```

The result for the LR test was 0.

Figure 3 Logit plot for math and grade with degree 1



doesn't look like HOT will help.

- Verbal

```
verbal.model1 = glm(grade~verbal, data=bio.train, family=binomial)
```

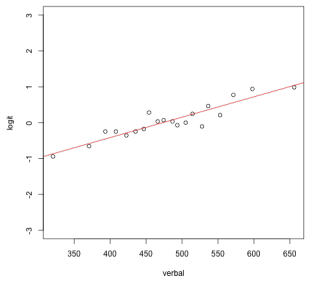
```
summary(verbal.model1)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9525  -1.1460   0.7704   1.1369   1.9182

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.6903524  0.2812212  -9.567   <2e-16 ***
verbal       0.0056867  0.0005779   9.840   <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: 3051.4  on 2201  degrees of freedom
Residual deviance: 2946.8  on 2200  degrees of freedom
AIC: 2950.8

Number of Fisher Scoring iterations: 4
```

```
LRtest(verbal.model1, basicmodel)
```

The result for the LR test was 0.

Figure 4 Logit plot for verbal and grade with degree 1



doesn't look like HOT will help.

- Prev

```
prevmodel = glm(grade~rank+math+verbal+rdg+father+mother+gender, data=bio.tra
LRtest(prevmodel, basicmodel)
```

The result for the LR test was 3.690684e-05.

- Gender

```
gendermodel = glm(grade~rank+math+verbal+prev+rdg+father+mother, data=bio.tra
LRtest(gendermodel, basicmodel)
```

The result for the LR test was 0.346872.

- rdg

```
rdgmodel = glm(grade~rank+math+verbal+prev+father+mother+gender, data=bio.tra
LRtest(rdgmodel, basicmodel)
```

The result for the LR test was 0.5859034.

- Father

```
fathermodel = glm(grade~rank+math+verbal+prev+rdg+mother+gender, data=bio.tra
LRtest(fathermodel, basicmodel)
```

The result for the LR test was 0.8947546.

```
father.recode = c('HighSchool/SomeCollege',
                  'Bachelor/Grad',
```

```
                    'Bachelor/Grad',
                    'HighSchool/SomeCollege',
                    'NoHighSchool/SomeHighSchool',
                    'NA',
                    'HighSchool/SomeCollege',
                    'NoHighSchool/SomeHighSchool')

new.father = father.recode[bio$father] %$
new.father.model = glm(grade~rank+math+verbal+prev+rdg+mother+gender+new.fat
                       data=bio.train, family=binomial)
summary(new.father.model)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1827  -0.9785   0.4198   0.9405   2.4355

Coefficients:
                                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                                 -6.140e+00  7.132e-01  -8.610  < 2e-16 ***
rank                                         3.823e+00  2.918e-01  13.102  < 2e-16 ***
math                                         4.428e-03  7.962e-04   5.561 2.68e-08 ***
verbal                                       2.314e-03  6.992e-04   3.310 0.000934 ***
prevTRUE                                    -7.258e-01  1.792e-01  -4.051 5.11e-05 ***
rdgFailed                                   -1.154e+01  3.090e+02  -0.037 0.970212
rdgNever                                     3.828e-01  4.338e-01   0.882 0.377539
rdgPassed                                    2.492e-01  5.467e-01   0.456 0.648555
motherBachelor Degree                        4.352e-01  4.258e-01   1.022 0.306761
motherGraduate/Professional degree           2.628e-01  4.463e-01   0.589 0.555959
motherHigh School Diploma or GED            -1.526e-01  4.213e-01  -0.362 0.717173
motherNo High School                         4.799e-01  6.162e-01   0.779 0.436087
motherNot Available                          6.252e-01  5.696e-01   1.098 0.272421
motherSome College                           1.848e-01  4.174e-01   0.443 0.657936
motherSome High School                      -1.010e-01  5.168e-01  -0.195 0.845121
genderMale                                  -9.466e-02  1.024e-01  -0.924 0.355386
new.father[train]HighSchool/SomeCollege     -8.698e-02  1.256e-01  -0.693 0.488506
new.father[train]NA                         -4.306e-01  4.035e-01  -1.067 0.285885
new.father[train]NoHighSchool/SomeHighSchool -1.303e-01  2.653e-01  -0.491 0.623247
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3051.4  on 2201  degrees of freedom
Residual deviance: 2564.1  on 2183  degrees of freedom
AIC: 2602.1
```

```
LRtest(new.father.model, basicmodel)
```

The result for the LR test was 0.8278164.

- Mother

```
mothermodel = glm(grade~rank+math+verbal+prev+rdg+father+gender, data=bio.tra
LRtest(mothermodel, basicmodel)
```

The result for the LR test was 0.02942975.

```
mother.recode = c('HighSchool/SomeCollege',
                  'Bachelor/Grad',
                  'Bachelor/Grad',
                  'HighSchool/SomeCollege',
                  'NoHighSchool/SomeHighSchool',
                  'NA',
                  'HighSchool/SomeCollege',
                  'NoHighSchool/SomeHighSchool')

new.mother = mother.recode[bio$mother] %$
new.mother.model = glm(grade~rank+math+verbal+prev+rdg+father+gender+new.moth
                       data=bio.train, family=binomial)
summary(new.mother.model)
LRtest(new.mother.model, basicmodel)
```

The result for the LR test was 0.1521139

```
mother.recode = c('NA',
                  'Bachelor/Grad',
                  'Bachelor/Grad',
                  'HighSchool/SomeCollege',
                  'NoHighSchool',
                  'NA',
                  'HighSchool/SomeCollege',
```

```
                    'SomeHighSchool')
new.mother = mother.recode[bio$mother] %$
new.mother.model = glm(grade~rank+math+verbal+prev+rdg+father+gender+new.moth
                       data=bio.train, family=binomial)
summary(new.mother.model)
LRtest(new.mother.model, basicmodel)
```

The result for the LR test was 0.05273772.

(c) Use stepwise and best subsets methods to narrow down the list of predictor variables. Given the small number of predictor variables, you can also adopt a manual selection approach to select the variables or to modify the results of the stepwise/best subsets procedures.

```
model = glm(grade~rank+math+verbal+prev+rdg+father+new.mother[train]+gender,
            data=bio.train, family=binomial)
step.model=step(model)

X = model.matrix(model)
X = X[,2:ncol(X)]
y = bio.train$grade %$
Xy = data.frame(X,y)

best.model = bestglm(Xy, family=binomial)
summary(best.model$BestModel) %$
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-0.9313 -0.3973  0.0403  0.3813  1.0926

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -0.6698868  0.0746661  -8.972  < 2e-16 ***
rank                             0.8153016  0.0528470  15.428  < 2e-16 ***
math                             0.0008846  0.0001515   5.840 6.01e-09 ***
verbal                           0.0004962  0.0001342   3.698 0.000223 ***
prevTRUE                        -0.1415409  0.0336836  -4.202 2.75e-05 ***
motherHigh.School.Diploma.or.GED -0.0916938  0.0236126  -3.883 0.000106 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4485 on 2196 degrees of freedom
Multiple R-squared:  0.197,   Adjusted R-squared:  0.1952
F-statistic: 107.8 on 5 and 2196 DF,  p-value: < 2.2e-16
```

Conclusion: keep rank, math, verbal, prev, new.mother

(d) Fit a tentative final model. The quantitative variables should be checked again for functional form and categorical variables should be checked for groupings. You can also consider adding interaction terms.

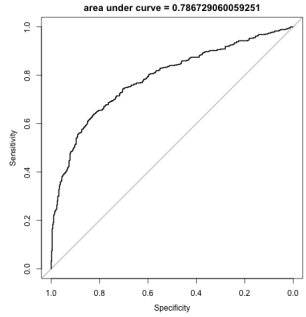
```
tentative.final = glm(grade~rank+I(rank^2)+I(rank^3)+I(rank^4)+math+verbal+prev+
                      data=new.bio[train,], family=binomial)
summary(tentative.final)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2690  -0.9487   0.3454   0.9617   2.2236

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -3.2948220  0.6181593  -5.330 9.82e-08 ***
rank                          -2.9792040  1.3933607  -2.138  0.03251 *
I(rank^2)                      5.6877054  1.1608730   4.900 9.61e-07 ***
math                           0.0038972  0.0007827   4.980 6.37e-07 ***
verbal                         0.0021858  0.0006889   3.173  0.00151 **
prevTRUE                      -0.7140206  0.1770225  -4.034 5.50e-05 ***
motherHighSchool/SomeCollege -0.3699392  0.1153981  -3.206  0.00135 **
motherNA                      -0.2146892  0.1467931  -1.463  0.14360
motherNoHighSchool             0.0005960  0.4421573   0.001  0.99892
motherSomeHighSchool          -0.6384259  0.3154586  -2.024  0.04299 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3051.4  on 2201  degrees of freedom
Residual deviance: 2553.1  on 2192  degrees of freedom
AIC: 2573.1

Number of Fisher Scoring iterations: 4
```

(e) Assess the performance of the model by determining its classification accuracy using a cutoff probability of 0.5 and finding the area under the ROC curve. Each of these metrics can be calculated from the training sample using leave-one-out or delete-d cross-validation, and they can be calculated using the validation sample.

```
results = predict(tentative.final, new.bio[-train,], type='response')
predicted.grade = (results >= 0.5) * 1
table(predicted.grade)
myacc <- confmatrix(bio$grade[-train], predicted.grade) %$
```

The result for the `LR` test was 0.7256356.



(f) Finally, assess the fit of the final model using the Hosmer-Lemeshow goodness-of-fit test.

```
Pihat = predict(tentative.final, type='response')
HLgof.test(fit=Pihat, obs=bio.train$grade) %$
```

```
        Hosmer-Lemeshow C statistic

data:  Pihat and bio.train$grade
X-squared = 8.1803, df = 8, p-value = 0.4161


$H

        Hosmer-Lemeshow H statistic

data:  Pihat and bio.train$grade
X-squared = 10.921, df = 8, p-value = 0.2062
```

```
#Data Mining hw 22
library(bestglm)
library(pROC)
library(MKmisc)

source('~/Dropbox/Tarleton/data_mining/generic_functions/dataset_ops.R')
source('~/Dropbox/Tarleton/data_mining/class_notes/outliers.R')
source('~/Dropbox/Tarleton/data_mining/class_notes/useful_logistic_ftns.R')
bio <- read.csv("~/Dropbox/Tarleton/data_mining/dfiles/BIOL120Data.csv", header=T,sep=',')

path = '~/Dropbox/Tarleton/data_mining/hw22/'

# 1. Import the data set BIO120.txt, which contains data for 3146 Biol 120 students,
#    including the following variables:
#    * Grade: 1 = A, B, or C, 0 = all other grades.
#    * Rank: Percentile rank represented as a decimal between 0 and 1, w
#      values close to 1 corresponding to higher ranked students.
#    * Math and Verbal: Math and Verbal SAT scores
#    * Prev: 1 = student has taken Bio120 before, and 0 = student has not
#    * Rdg: Status of student regarding the remedial course Reading 100.
#      possible levels are Never Taken, Concurrently Enrolled, Passed, and Failed.
#    * Father's and Mother's education levels
#    * Gender

# 2. Build the best possible logistic regression model for predicting grade based on
#     the other variables.
#
#    (a). Divide the data set into two parts for the purpose of cross-validation.

          splitset <- splitdata(bio, 0.7)
          train = splitset$train
          bio.train <- bio[train,]
          bio.test <- bio[-train,]

#    (b). Fit a univariate model regressing grade onto each of the other variables.
#         For the quantitative variables, attempt to determine if higher order terms
#         are needed using the groupplot function (see LogisticRegressionFunctions.txt
#         for some helpeful functions). As with linear regression, you can use a
#         likelihood ratio test to formally test whether these terms are needed
#         (LRtest function)
```

```
#         For the categorical variables, a univariate model can help to determine if
#         some of the levels can be grouped together to create a variable with fewer
#         levels. This is essential for the father and mother variables which have 8
#         levels. It is likely that a stepwise regression will eliminate one of the
#         parent's education variables, since they are highly correlated and have a
#         large number of parameters.

          basicmodel <- glm(grade~., data = bio.train, family=binomial)
          summary(basicmodel)

          #rank
          rank.model1 = glm(grade~rank, data=bio.train, family=binomial)
          summary(rank.model1)
          LRtest(rank.model1, basicmodel)
          #1.110223e-16
          # Conclusion: keep rank

          # rank higher order terms?
          quantlogitplot(bio.train$grade, bio.train$rank, 1, 'math', 'logit', 10, c(-3,3))
          dev.copy(png, paste0(path,'grade_rank_logit.png'))
          dev.off()
          quantlogitplot(bio.train$grade, bio.train$rank, 2, 'math', 'logit', 10, c(-3,3))
          dev.copy(png, paste0(path,'grade_rank_logit_deg2.png'))
          dev.off()

          rank.model2 = glm(grade~rank+I(rank^2), data=bio.train, family=binomial)
          summary(rank.model2)
          LRtest(rank.model2, basicmodel)
          # 6.023138e-09
          rank.model3 = glm(grade~rank+I(rank^2)+I(rank^3), data=bio.train, family=binomial)
          summary(rank.model3)
          LRtest(rank.model3, basicmodel)
          # 5.600436e-08
          rank.model4 = glm(grade~rank+I(rank^2)+I(rank^3)+I(rank^4), data=bio.train, family=binomial)
          summary(rank.model4)
          LRtest(rank.model4, basicmodel)
          # 4.13323e-08

          #math
          math.model1 = glm(grade~math, data=bio.train, family=binomial)
```

```
summary(math.model1)
LRtest(math.model1, basicmodel)
# 0
# Conclusion: keep math

quantlogitplot(bio.train$grade, bio.train$math, 1, 'math', 'logit', 20, c(-3,3))
dev.copy(png, paste0(path,'grade_math_logit.png'))
dev.off()
# doesn't look like HOT will help.

#verbal
verbal.model1 = glm(grade~verbal, data=bio.train, family=binomial)
summary(verbal.model1)
LRtest(verbal.model1, basicmodel)
# 0
# Conclusion: keep verbal

quantlogitplot(bio.train$grade, bio.train$verbal, 1, 'verbal', 'logit', 20, c(-3,3))
dev.copy(png, paste0(path,'grade_verbal_logit.png'))
dev.off()
# doesn't look like HOT will help.


#PREV
table(bio$prev)
prevmodel = glm(grade~rank+math+verbal+rdg+father+mother+gender, data=bio.train, family=binomial)
LRtest(prevmodel, basicmodel)
#3.690684e-05
# Conclusion: keep prev

#gender
table(bio$gender)
gendermodel = glm(grade~rank+math+verbal+prev+rdg+father+mother, data=bio.train, family=binomial)
LRtest(gendermodel, basicmodel)
#0.346872
# Conclusion: drop gender

#RDG
table(bio$rdg)
rdgmodel = glm(grade~rank+math+verbal+prev+father+mother+gender, data=bio.train, family=binomial)
```

```
LRtest(rdgmodel, basicmodel)
#0.5859034
# Conclusion: drop rdg

#FATHER
table(bio$father)
fathermodel = glm(grade~rank+math+verbal+prev+rdg+mother+gender, data=bio.train, family=binomial)
LRtest(fathermodel, basicmodel)
#0.8947546
# Conclusion: drop father
# re - leveling father
father.recode = c('HighSchool/SomeCollege',
                  'Bachelor/Grad',
                  'Bachelor/Grad',
                  'HighSchool/SomeCollege',
                  'NoHighSchool/SomeHighSchool',
                  'NA',
                  'HighSchool/SomeCollege',
                  'NoHighSchool/SomeHighSchool')

new.father = father.recode[bio$father]
new.father.model = glm(grade~rank+math+verbal+prev+rdg+mother+gender+new.father[train],
                       data=bio.train, family=binomial)
summary(new.father.model)
LRtest(new.father.model, basicmodel)
#0.8278164
# Conclusion--still drop father

#MOTHER
table(bio$mother)
mothermodel = glm(grade~rank+math+verbal+prev+rdg+father+gender, data=bio.train, family=binomial)
LRtest(mothermodel, basicmodel)
#0.02942975
# Conclusion: keep mother

# re - leveling mother
mother.recode = c('HighSchool/SomeCollege',
                  'Bachelor/Grad',
                  'Bachelor/Grad',
                  'HighSchool/SomeCollege',
```

```
                        'NoHighSchool/SomeHighSchool',
                        'NA',
                        'HighSchool/SomeCollege',
                        'NoHighSchool/SomeHighSchool')

        new.mother = mother.recode[bio$mother]
        new.mother.model = glm(grade~rank+math+verbal+prev+rdg+father+gender+new.mother[train],
                            data=bio.train, family=binomial)
        summary(new.mother.model)
        LRtest(new.mother.model, basicmodel)
        # 0.1521139
        # Conclusion: Bad recoding

        mother.recode = c('NA',
                        'Bachelor/Grad',
                        'Bachelor/Grad',
                        'HighSchool/SomeCollege',
                        'NoHighSchool',
                        'NA',
                        'HighSchool/SomeCollege',
                        'SomeHighSchool')
        new.mother = mother.recode[bio$mother]
        new.mother.model = glm(grade~rank+math+verbal+prev+rdg+father+gender+new.mother[train],
                            data=bio.train, family=binomial)
        summary(new.mother.model)
        LRtest(new.mother.model, basicmodel)
        # 0.05273772
        # Conclusion: better recoding

#    (c). Use stepwise and best subsets methods to narrow down the list of predictor
#         variables. Given the small number of predictor variables, you can also adopt
#         a manual selection approach to select the variables or to modify the results
#         of the stepwise/best subsets procedures.

        model = glm(grade~rank+math+verbal+prev+rdg+father+new.mother[train]+gender,
                    data=bio.train, family=binomial)
        step.model=step(model)

        X = model.matrix(model)
        X = X[,2:ncol(X)]
```

```
          y = bio.train$grade
          Xy = data.frame(X,y)

#          best.model = bestglm(Xy, family=binomial)
#          summary(best.model$BestModel)
          # Conclusion: keep rank, math, verbal, prev, new.mother


#    (d). Fit a tentative final model. The quantitative variables should be checked again
#         for functional form and categorical variables should be checked for groupings.
#         You can also consider adding interaction terms.
          new.bio <- bio
          new.bio$mother <- new.mother

          tentative.final = glm(grade~rank+I(rank^2)+I(rank^3)+I(rank^4)+math+verbal+prev+mother,
                                data=new.bio[train,], family=binomial)
          summary(tentative.final)

#    (e). Assess the performance of the model by determining its classification accuracy
#         using a cutoff probability of 0.5 and finding the area under the ROC curve.
#         Each of these metrics can be calculated from the training sample using
#         leave-one-out or delete-d cross-validation, and they can be calculated using
#         the validation sample.

          results = predict(tentative.final, new.bio[-train,], type='response')
          predicted.grade = (results >= 0.5) * 1
          table(predicted.grade)
          myacc <- confmatrix(bio$grade[-train], predicted.grade)
          #0.7256356

          rc <- roc(bio$grade[-train], results)
          plot(rc, main=paste0('area␣under␣curve␣=␣', rc$auc))

#    (f). Finally, assess the fit of the final model using the Hosmer-Lemeshow goodness-
#         of-fit test.

          Pihat = predict(tentative.final, type='response')
          HLgof.test(fit=Pihat, obs=bio.train$grade)
```