GR5291 Advanced Data Analysis Problem Set 4

Francis Zhang

October 9, 2024

Question

1.Perform a multiple linear regression model of 'bwt' birth weight in grams on the explanatory variables:

- 'age' mother's age in years
- 'lwt' mother's weight in pounds at last menstrual period
- 'race' mother's race ('0' = white, '1' = other)
- 'smoke' smoking status during pregnancy
- 'ptl' number of previous premature labours
- 'ht' history of hypertension
- 'ui' presence of uterine irritability
- 'ftv' number of physician visits during the first trimeste

i) Investigate whether there is any multicollinearity

ii) Run a ridge regression analysis and compare the results with the OLS results

2.Compare models selected using LASSO and a stepwise procedure to predict 'bwt' birth weight in grams using the above set of predictors.

3.For the procedures listed in Table 1 next page, give appropriate ranks with respect to the listed attributes: 1 = Good, 2 = Fair, 3= Poor. Given supporting reference from the literature, if you wish.

Solution

Question 1

Overview of the Dataset

```
library(MASS)
names(birthwt)
```

```
##  [1] "low"   "age"   "lwt"   "race"  "smoke" "ptl"   "ht"    "ui"    "ftv"
## [10] "bwt"
```

```
head(birthwt)
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
## 88   0  21 108    1     1   0  0  1   2 2594
## 89   0  18 107    1     1   0  0  1   0 2600
```

```
## 91    0  21 124    3      0   0  0  0    0 2622
```

```r
summary(birthwt)
```

```
##       low              age             lwt             race      
##  Min.   :0.0000   Min.   :14.00   Min.   : 80.0   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:19.00   1st Qu.:110.0   1st Qu.:1.000  
##  Median :0.0000   Median :23.00   Median :121.0   Median :1.000  
##  Mean   :0.3122   Mean   :23.24   Mean   :129.8   Mean   :1.847  
##  3rd Qu.:1.0000   3rd Qu.:26.00   3rd Qu.:140.0   3rd Qu.:3.000  
##  Max.   :1.0000   Max.   :45.00   Max.   :250.0   Max.   :3.000  
##      smoke             ptl               ht                ui        
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000  
##  Mean   :0.3915   Mean   :0.1958   Mean   :0.06349   Mean   :0.1481  
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000  
##  Max.   :1.0000   Max.   :3.0000   Max.   :1.00000   Max.   :1.0000  
##       ftv              bwt      
##  Min.   :0.0000   Min.   : 709  
##  1st Qu.:0.0000   1st Qu.:2414  
##  Median :0.0000   Median :2977  
##  Mean   :0.7937   Mean   :2945  
##  3rd Qu.:1.0000   3rd Qu.:3487  
##  Max.   :6.0000   Max.   :4990
```

```r
dim(birthwt)
```

```
## [1] 189  10
```

```r
# normalize 'race' column
birthwt$race <- ifelse(birthwt$race == 1, 0, 1)
head(birthwt)
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    1     0   0  0  1   0 2523
## 86   0  33 155    1     0   0  0  0   3 2551
## 87   0  20 105    0     1   0  0  0   1 2557
## 88   0  21 108    0     1   0  0  1   2 2594
## 89   0  18 107    0     1   0  0  1   0 2600
## 91   0  21 124    1     0   0  0  0   0 2622
```

```r
summary(birthwt)
```

```
##       low              age             lwt             race       
##  Min.   :0.0000   Min.   :14.00   Min.   : 80.0   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:19.00   1st Qu.:110.0   1st Qu.:0.0000  
##  Median :0.0000   Median :23.00   Median :121.0   Median :0.0000  
##  Mean   :0.3122   Mean   :23.24   Mean   :129.8   Mean   :0.4921  
##  3rd Qu.:1.0000   3rd Qu.:26.00   3rd Qu.:140.0   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :45.00   Max.   :250.0   Max.   :1.0000  
##      smoke             ptl               ht                ui        
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000  
##  Mean   :0.3915   Mean   :0.1958   Mean   :0.06349   Mean   :0.1481  
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000
```

```
##  Max.   :1.0000   Max.   :3.0000   Max.   :1.00000   Max.   :1.0000
##       ftv              bwt
##  Min.   :0.0000   Min.   : 709
##  1st Qu.:0.0000   1st Qu.:2414
##  Median :0.0000   Median :2977
##  Mean   :0.7937   Mean   :2945
##  3rd Qu.:1.0000   3rd Qu.:3487
##  Max.   :6.0000   Max.   :4990
```

```r
dim(birthwt)
```

```
## [1] 189  10
```

Develop models

```r
lm_model <- lm(bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv,
               data = birthwt)
summary(lm_model)
```

```
##
## Call:
## lm(formula = bwt ~ age + lwt + race + smoke + ptl + ht + ui +
##     ftv, data = birthwt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1838.40  -443.44    12.04   467.77  1675.38
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2971.999    308.188   9.643  < 2e-16 ***
## age           -2.833      9.572  -0.296 0.767613
## lwt            3.946      1.664   2.371 0.018795 *
## race        -397.275    102.988  -3.857 0.000159 ***
## smoke       -366.181    105.039  -3.486 0.000616 ***
## ptl          -47.525    101.881  -0.466 0.641437
## ht          -591.885    202.149  -2.928 0.003853 **
## ui          -512.896    138.717  -3.697 0.000289 ***
## ftv          -15.276     46.406  -0.329 0.742405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 649.8 on 180 degrees of freedom
## Multiple R-squared:  0.2398, Adjusted R-squared:  0.206
## F-statistic: 7.097 on 8 and 180 DF,  p-value: 3.828e-08
```

i) Multicollinearity Check

- Multicollinearity can inflate variance, making coefficients unstable and difficult to interpret. To investigate multicollinearity, we use the Variance Inflation Factor (VIF). VIF values above 5-10 may indicate problematic multicollinearity.

```r
lm_model <- lm(bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv,
               data = birthwt)
summary(lm_model)
```

```
##
## Call:
## lm(formula = bwt ~ age + lwt + race + smoke + ptl + ht + ui +
##     ftv, data = birthwt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1838.40  -443.44    12.04   467.77  1675.38
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2971.999    308.188   9.643  < 2e-16 ***
## age           -2.833      9.572  -0.296 0.767613
## lwt            3.946      1.664   2.371 0.018795 *
## race        -397.275    102.988  -3.857 0.000159 ***
## smoke       -366.181    105.039  -3.486 0.000616 ***
## ptl          -47.525    101.881  -0.466 0.641437
## ht          -591.885    202.149  -2.928 0.003853 **
## ui          -512.896    138.717  -3.697 0.000289 ***
## ftv          -15.276     46.406  -0.329 0.742405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 649.8 on 180 degrees of freedom
## Multiple R-squared:  0.2398, Adjusted R-squared:  0.206
## F-statistic: 7.097 on 8 and 180 DF,  p-value: 3.828e-08
```

```r
library(car)
```

```
## Loading required package: carData
```

```r
vif_values <- vif(lm_model)
vif_values
```

```
##      age      lwt     race    smoke      ptl       ht       ui      ftv
## 1.145378 1.153326 1.186686 1.176633 1.124891 1.087691 1.087057 1.075996
```

The VIF values obtained are all below 2, suggesting that multicollinearity is not a serious issue in this model.

ii) Ridge Regression Analysis

- Ridge regression adds a penalty to the regression model based on the sum of the squared coefficients, which helps manage multicollinearity.
- We can compare the coefficients and interpret how Ridge Regression handles multicollinearity differently from OLS.

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
x <- model.matrix(bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv,
                  data = birthwt)[, -1]
y <- birthwt$bwt

ridge_model <- glmnet(x, y, alpha = 0)
```

```r
cv_ridge <- cv.glmnet(x, y, alpha = 0)
best_lambda <- cv_ridge$lambda.min
ridge_coefs <- predict(ridge_model, s = best_lambda, type = "coefficients")

print(ridge_coefs)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) 2894.5366978
## age            0.3800058
## lwt            3.1480135
## race        -303.5504955
## smoke       -276.1707449
## ptl          -70.8654858
## ht          -460.6239472
## ui          -420.8876743
## ftv           -5.6315740
```

```r
print(coef(lm_model))
```

```
## (Intercept)         age         lwt        race       smoke         ptl
## 2971.998538   -2.832732    3.946122 -397.274691 -366.180997  -47.525354
##          ht          ui         ftv
## -591.884795 -512.895777  -15.275928
```

Ridge regression coefficients are generally closer to zero than those from OLS, which is expected due to the penalty applied on large coefficients to handle multicollinearity. Notably, the absolute values for smoke and race are reduced in Ridge, which suggests that Ridge regression is dampening their influence due to their correlation with other predictors.

In summary, Ridge regression helps stabilize the coefficients by shrinking them towards zero, particularly useful in the presence of mild multicollinearity, as it reduces the variance of coefficients without performing variable selection.

Question 2

LASSO

- LASSO regression is helpful for variable selection as it forces some coefficients to exactly zero, effectively selecting a subset of predictors.

```r
lasso_model <- glmnet(x, y, alpha = 1)  # alpha = 1 for lasso

cv_lasso <- cv.glmnet(x, y, alpha = 1)
best_lambda_lasso <- cv_lasso$lambda.min
lasso_coefs <- predict(lasso_model, s = best_lambda_lasso,
                       type = "coefficients")

print(lasso_coefs)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) 2933.698144
## age                   .
## lwt            3.076997
## race        -334.222774
```

```
## smoke       -307.420277
## ptl          -29.024826
## ht          -468.563726
## ui          -460.984450
## ftv            .
```

Stepwise Selection

- We can use stepwise selection based on AIC (Akaike Information Criterion) for feature selection.

```r
stepwise_model <- step(lm_model, direction = "both", trace = FALSE)

summary(stepwise_model)
```

```
##
## Call:
## lm(formula = bwt ~ lwt + race + smoke + ht + ui, data = birthwt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1853.02  -460.22    26.17   450.76  1620.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2894.000    233.044  12.418  < 2e-16 ***
## lwt            3.866      1.608   2.405 0.017180 *
## race        -389.695     99.721  -3.908 0.000131 ***
## smoke       -370.289    101.881  -3.635 0.000362 ***
## ht          -584.427    199.450  -2.930 0.003819 **
## ui          -522.540    134.495  -3.885 0.000143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 645.3 on 183 degrees of freedom
## Multiple R-squared:  0.2377, Adjusted R-squared:  0.2169
## F-statistic: 11.41 on 5 and 183 DF,  p-value: 1.345e-09
```

Comparison of Models (LASSO vs Stepwise)

LASSO and Stepwise selection both produce models that simplify prediction, but they do so differently. LASSO promotes model sparsity by penalizing and often eliminating weaker predictors, which can improve generalization to new data. Stepwise selection, driven by AIC, tends to retain a larger number of predictors to optimize model fit, sometimes including variables with marginal effects. Thus, LASSO's model may be more robust to variability, while Stepwise may provide slightly better in-sample fit due to its inclusion of additional predictors.

Overall, LASSO's tendency to favor simpler models may improve generalizability to new data, while Stepwise selection's focus on AIC may provide a slightly better fit for the training data by retaining more predictors.

Question 3

Attribute	OLS	Ridge	LASSO	Elastic Net
Performance when p » n	3	2	1	1
Performance under multicollinearity	3	1	2	1
Unbiased estimation	1	3	3	3

Attribute	OLS	Ridge	LASSO	Elastic Net
Model selection	3	3	1	1
Simplicity: Computation, Inference, Interpretation	1	2	2	2

Explanation:

1.Performance when p » n (High-dimensional Data):

- OLS (3): OLS performs poorly when the number of predictors (p) is greater than the number of observations (n) as it overfits and leads to unstable estimates.
- Ridge (2): Ridge regression handles high-dimensional data better than OLS by shrinking coefficients, but it does not perform variable selection.
- LASSO (1) and Elastic Net (1): Both are better suited for high-dimensional data. LASSO performs automatic variable selection, setting some coefficients to zero, and Elastic Net combines Ridge and LASSO properties, making it versatile in high dimensions.

2.Performance under Multicollinearity:

- OLS (3): OLS is sensitive to multicollinearity, which can inflate variance and destabilize coefficients.
- Ridge (1): Ridge regression is highly effective for handling multicollinearity as it applies a penalty that reduces coefficient variance.
- LASSO (2): LASSO can handle multicollinearity by selecting a subset of predictors, but it is less effective than Ridge at reducing the influence of multicollinear predictors.
- Elastic Net (1): Elastic Net performs well with multicollinear data because it combines LASSO's variable selection with Ridge's penalty for correlated predictors.

3.Unbiased Estimation:

- OLS (1): OLS provides unbiased estimates under the assumption that there is no multicollinearity and no omitted variable bias.
- Ridge (3), LASSO (3), Elastic Net (3): These methods introduce bias by shrinking coefficients. This bias-variance trade-off can improve prediction accuracy but sacrifices unbiased estimation.

4.Model Selection:

- OLS (3): OLS includes all predictors without any form of selection, which can lead to overfitting.
- Ridge (3): While Ridge regression shrinks coefficients, it does not perform variable selection (does not set coefficients to zero).
- LASSO (1) and Elastic Net (1): LASSO performs automatic variable selection by setting some coefficients to zero, and Elastic Net combines this with Ridge's stability, making both effective for model selection.

5.Simplicity: Computation, Inference, Interpretation:

- OLS (1): OLS is straightforward to compute and interpret as it does not involve any penalty terms.
- Ridge (2), LASSO (2), Elastic Net (2): These methods add complexity due to the need for cross-validation to choose the optimal penalty parameters. Interpretation can also be more challenging because of shrinkage and variable selection effects.