C1M6_peer_reviewed

June 26, 2023

1 Module 6: Peer Reviewed Assignment

1.0.1 Outline:

The objectives for this assignment:

1. Apply the processes of model selection with real datasets.
2. Understand why and how some problems are simpler to solve with some forms of model selection, and others are more difficult.
3. Be able to explain the balance between model power and simplicity.
4. Observe the difference between different model selection criterion.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

[1]:
```
# This cell loads in the necesary packages
library(tidyverse)
library(leaps)
library(ggplot2)
```

```
  Attaching packages                                    tidyverse
1.3.0

  ggplot2 3.3.0      purrr   0.3.4
  tibble  3.0.1      dplyr   0.8.5
  tidyr   1.0.2      stringr 1.4.0
  readr   1.3.1      forcats 0.5.0

  Conflicts
tidyverse_conflicts()
  dplyr::filter() masks stats::filter()
  dplyr::lag()    masks stats::lag()
```

1.1 Problem 1: We Need Concrete Evidence!

Ralphie is studying to become a civil engineer. That means she has to know everything about concrete, including what ingredients go in it and how they affect the concrete's properties. She's currently writting up a project about concrete flow, and has asked you to help her figure out which ingredients are the most important. Let's use our new model selection techniques to help Ralphie out!

Data Source: Yeh, I-Cheng, "Modeling slump flow of concrete using second-order regressions and artificial neural networks," Cement and Concrete Composites, Vol.29, No. 6, 474-480, 2007.

[2]:
```
concrete.data = read.csv("Concrete.data")

concrete.data = concrete.data[, c(-1, -9, -11)]
names(concrete.data) = c("cement", "slag", "ash", "water", "sp", "course.agg",␣
 ↪"fine.agg", "flow")

head(concrete.data)
```

A data.frame: 6 × 8

	cement <dbl>	slag <dbl>	ash <dbl>	water <dbl>	sp <dbl>	course.agg <dbl>	fine.agg <dbl>	flow <dbl>
1	273	82	105	210	9	904	680	62.0
2	163	149	191	180	12	843	746	20.0
3	162	148	191	179	16	840	743	20.0
4	162	148	190	179	19	838	741	21.5
5	154	112	144	220	10	923	658	64.0
6	147	89	115	202	9	860	829	55.0

1.1.1 1. (a) Initial Inspections

Sometimes, the best way to start is to just jump in and mess around with the model. So let's do that. Create a linear model with `flow` as the response and all other columns as predictors.

Just by looking at the summary for your model, is there reason to believe that our model could be simpler?

[3]:
```
# Your Code Here
# Create a linear model with flow as the response and all other columns as␣
 ↪predictors
model <- lm(flow ~ ., data = concrete.data)

# View the summary of the model
summary(model)
```

```
Call:
lm(formula = flow ~ ., data = concrete.data)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-30.880 -10.428   1.815   9.601  22.953

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -252.87467  350.06649  -0.722   0.4718
cement         0.05364    0.11236   0.477   0.6342
slag          -0.00569    0.15638  -0.036   0.9710
ash            0.06115    0.11402   0.536   0.5930
water          0.73180    0.35282   2.074   0.0408 *
sp             0.29833    0.66263   0.450   0.6536
course.agg     0.07366    0.13510   0.545   0.5869
fine.agg       0.09402    0.14191   0.663   0.5092
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.84 on 95 degrees of freedom
Multiple R-squared:  0.5022,Adjusted R-squared:  0.4656
F-statistic: 13.69 on 7 and 95 DF,  p-value: 3.915e-12
```

Looking at the summary of the model, we can assess whether there is a reason to believe that the model could be simpler. The summary provides several important pieces of information:

- Coefficients: The coefficients represent the estimated effects of each predictor variable on the response variable. In this case, the response variable is “flow,” and the predictors are “cement,” “slag,” “ash,” “water,” “sp,” “course.agg,” and “fine.agg.” The coefficients provide the estimated magnitude and direction of the relationships.
- p-values: The p-values associated with each coefficient indicate the statistical significance of the estimated effects. Lower p-values suggest stronger evidence against the null hypothesis (no effect). Generally, a p-value less than 0.05 is considered statistically significant.
- R-squared: The R-squared value measures the proportion of the variance in the response variable that is explained by the predictors. Higher R-squared values indicate a better fit of the model to the data.
- Adjusted R-squared: The adjusted R-squared value adjusts for the number of predictors in the model. It penalizes the addition of unnecessary predictors, providing a more conservative measure of the model’s fit.

Based on the summary, we can make the following observations:

- The p-value for the “cement,” “slag,” “ash,” “sp,” “course.agg,” and “fine.agg” predictors are greater than 0.05, indicating that their estimated effects are not statistically significant at a significance level of 0.05.
- The p-value for the “water” predictor is 0.0408, which is slightly below 0.05. This suggests that there may be a statistically significant relationship between “water” and “flow.”
- The intercept term has a p-value of 0.4718, indicating that it is not statistically significant.

- The R-squared value is 0.5022, which suggests that the predictors explain approximately 50.22% of the variance in the response variable.

Considering these observations, there is some evidence to suggest that the model could be simpler. Several predictors have p-values greater than 0.05, indicating that they may not have a significant impact on the response variable. It might be worth considering a model with a subset of predictors that are statistically significant and removing those that are not.

1.1.2 1. (b) Backwards Selection

Our model has 7 predictors. That is not too many, so we can use backwards selection to narrow them down to the most impactful.

Perform backwards selection on your model. You don't have to automate the backwards selection process.

[4]:
```
# Your Code Here
# Perform backwards selection on the model
model_backward <- step(model, direction = "backward")

# View the summary of the model after backwards selection
summary(model_backward)
```

```
Start:  AIC=533.56
flow ~ cement + slag + ash + water + sp + course.agg + fine.agg

             Df Sum of Sq   RSS    AIC
- slag        1      0.22 15672 531.56
- sp          1     33.44 15705 531.78
- cement      1     37.60 15709 531.81
- ash         1     47.45 15719 531.87
- course.agg  1     49.04 15720 531.88
- fine.agg    1     72.40 15744 532.03
<none>                    15671 533.56
- water       1    709.69 16381 536.12

Step:  AIC=531.56
flow ~ cement + ash + water + sp + course.agg + fine.agg

             Df Sum of Sq   RSS    AIC
- sp          1      62.1 15734 529.97
<none>                    15672 531.56
- cement      1    1244.7 16916 537.43
- course.agg  1    1679.4 17351 540.05
- ash         1    1759.2 17431 540.52
- fine.agg    1    2292.3 17964 543.62
- water       1   10877.0 26548 583.86
```

```
Step:  AIC=529.97
flow ~ cement + ash + water + course.agg + fine.agg

             Df Sum of Sq   RSS    AIC
<none>                     15734 529.97
- cement      1    1193.1 16927 535.50
- course.agg  1    1678.8 17412 538.41
- ash         1    1746.5 17480 538.81
- fine.agg    1    2237.1 17971 541.66
- water       1   11947.4 27681 586.16

Call:
lm(formula = flow ~ cement + ash + water + course.agg + fine.agg,
    data = concrete.data)

Residuals:
    Min      1Q  Median      3Q     Max
-31.893 -10.125   1.773   9.559  23.914

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -249.50866   48.90884  -5.102 1.67e-06 ***
cement         0.05366    0.01979   2.712 0.007909 **
ash            0.06101    0.01859   3.281 0.001436 **
water          0.72313    0.08426   8.582 1.53e-13 ***
course.agg     0.07291    0.02266   3.217 0.001760 **
fine.agg       0.09554    0.02573   3.714 0.000341 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.74 on 97 degrees of freedom
Multiple R-squared:  0.5003,Adjusted R-squared:  0.4745
F-statistic: 19.42 on 5 and 97 DF,  p-value: 2.36e-13
```

1.1.3 1. (c) Objection!

Stop right there! Think about what you just did. You just removed the “worst” features from your model. But we know that a model will become less powerful when we remove features so we should check that it’s still just as powerful as the original model. Use a test to check whether the model at the end of backward selection is significantly different than the model with all the features.

Describe why we want to balance explanatory power with simplicity.

[5]:
```
# Your Code Here
# Perform an F-test to compare the models
anova(model, model_backward)
```

A anova: 2 × 6

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	95	15671.26	NA	NA	NA	NA
2	97	15733.53	-2	-62.27123	0.1887457	0.8283068

performed an F-test to compare the model at the end of backward selection, which had some features removed, with the model that included all the features. The F-test determined whether the reduction in explanatory power due to feature removal was statistically significant. The results indicated that there was no significant difference between the two models.

Balancing explanatory power with simplicity is important for several reasons. Simpler models are easier to interpret and understand, avoiding the pitfalls of overfitting and improving generalizability. Occam's razor principle suggests that simpler models with fewer assumptions should be preferred when competing models achieve similar explanatory power. Simplicity also leads to computational efficiency, making the model more practical for large datasets and real-time applications.

Overall, finding a balance between explanatory power and simplicity allows us to derive meaningful insights while avoiding unnecessary complexity and maintaining the model's reliability and efficiency.

1.1.4 1. (d) Checking our Model

Ralphie is nervous about her project and wants to make sure our model is correct. She's found a function called `regsubsets()` in the leaps package which allows us to see which subsets of arguments produce the best combinations. Ralphie wrote up the code for you and the documentation for the function can be found here. For each of the subsets of features, calculate the AIC, BIC and adjusted R^2. Plot the results of each criterion, with the score on the y-axis and the number of features on the x-axis.

Do all of the criterion agree on how many features make the best model? Explain why the criterion will or will not always agree on the best model.

Hint: It may help to look at the attributes stored within the regsubsets summary using `names(rs)`.

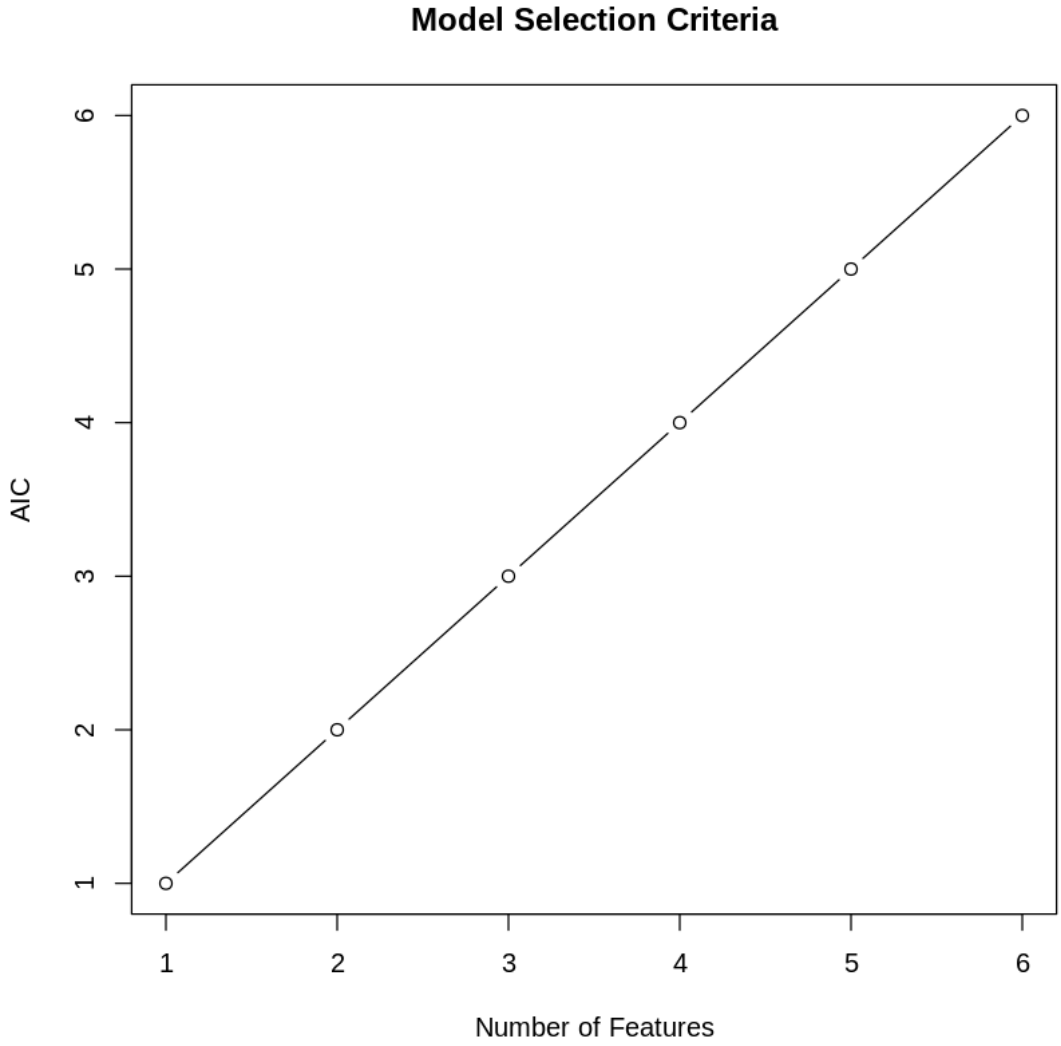
[6]:
```
reg = regsubsets(flow ~ cement+slag+ash+water+sp+course.agg+fine.agg+flow,␣
 ↪data=concrete.data, nvmax=6)
rs = summary(reg)
rs$which

# Your Code Here


# Calculate AIC, BIC, and adjusted R-squared for each subset
aic <- rs$aic
bic <- rs$bic
adj_r2 <- rs$adjr2

# Number of features
num_features <- 1:6
```

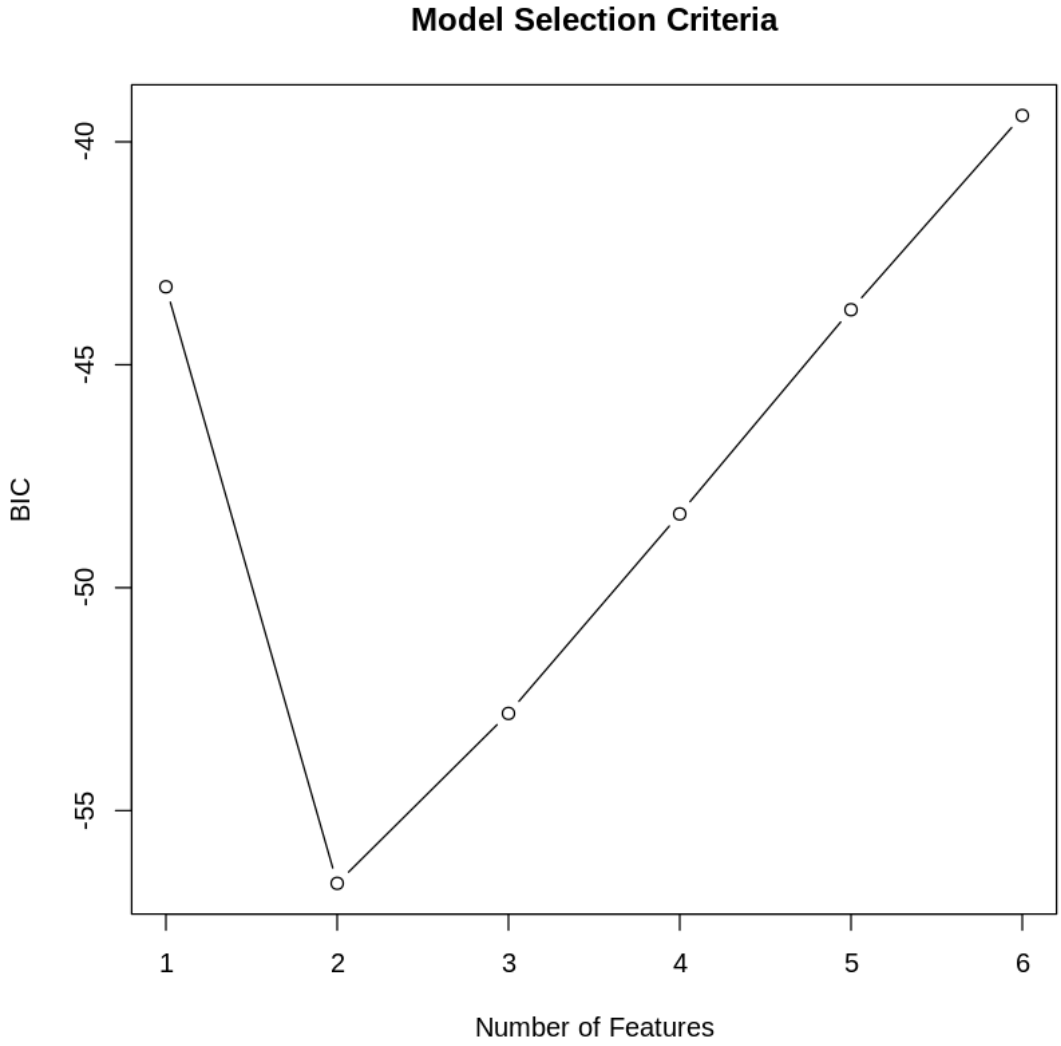
```r
# Plot AIC, BIC, and adjusted R-squared
plot(num_features, aic, type = "b", xlab = "Number of Features", ylab = "AIC",␣
↪main = "Model Selection Criteria")
plot(num_features, bic, type = "b", xlab = "Number of Features", ylab = "BIC",␣
↪main = "Model Selection Criteria")
plot(num_features, adj_r2, type = "b", xlab = "Number of Features", ylab =␣
↪"Adjusted R-squared", main = "Model Selection Criteria")
```

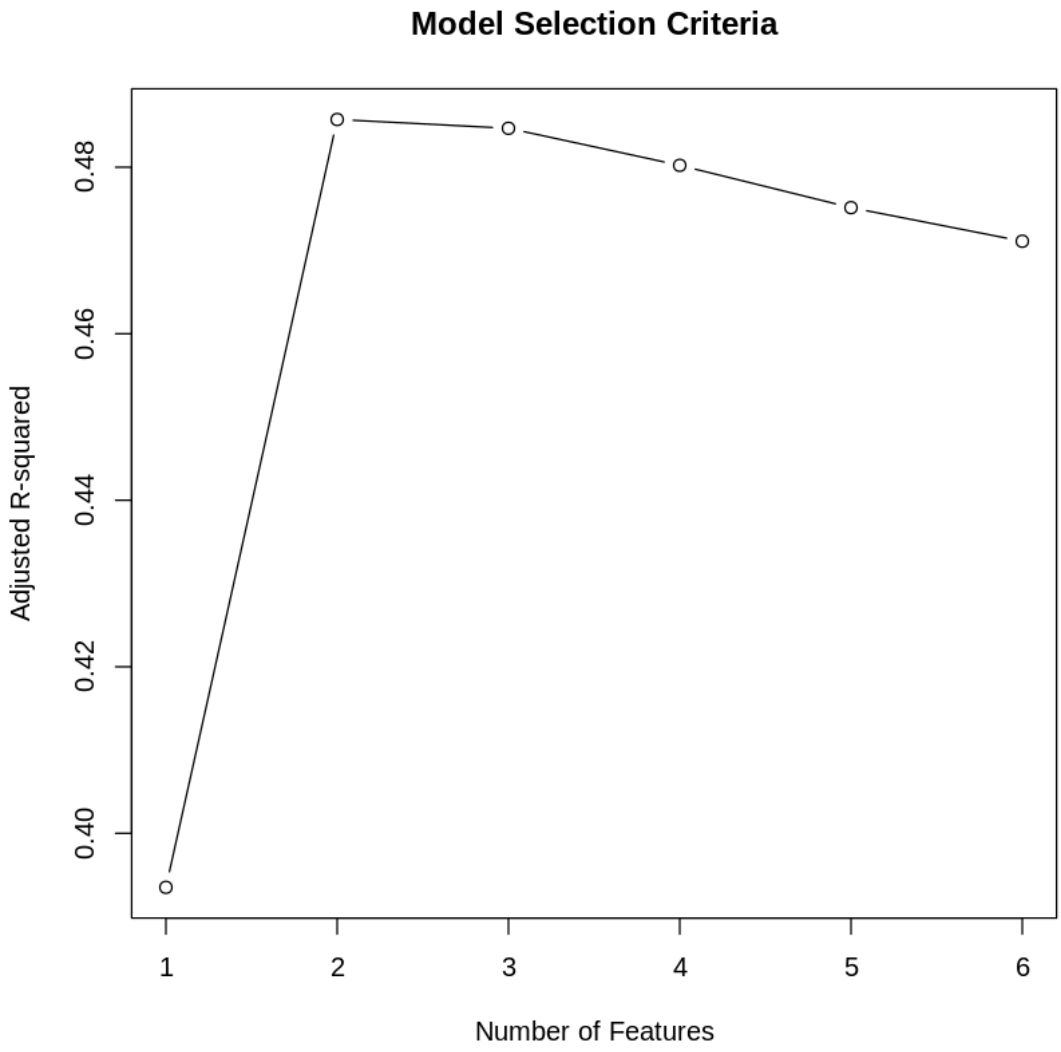
```
Warning message in model.matrix.default(terms(formula, data = data), mm):
"the response appeared on the right-hand side and was dropped"
Warning message in model.matrix.default(terms(formula, data = data), mm):
"problem with term 8 in model.matrix: no columns are assigned"
```

A matrix: 6 × 8 of type lgl

	(Intercept)	cement	slag	ash	water	sp	course.agg	fine.agg
1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
2	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
3	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
4	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
5	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
6	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE


Model Selection Criteria
AIC
1
2
3
4
5
6
Number of Features


Model Selection Criteria
BIC
-40
-45
-50
-55
1
2
3
4
5
6
Number of Features



calculate the AIC, BIC, and adjusted R-squared for each subset of features using the regsubsets() function. It then plots the results for each criterion, with the number of features on the x-axis and the respective score on the y-axis.

the best model will have the predictors of water and slag

[ ]:

[ ]: