IDS 702: Module 1.12

Bringing the MLR pieces together II (illustration)

Dr. Olanrewaju Michael Akande

IDS 702

Back to the diamonds data

Let's try model selection for our diamonds example. We will do this on the log scale - recall our analysis in the previous module.

First, forward selection using AIC

```
diamonds <- read.csv("data/diamonds.csv", header= T,
                     colClasses = c("numeric","factor","factor","factor","numeric"))
diamonds$CaratsCent <- diamonds$Carats - mean(diamonds$Carats)
diamonds$CaratsCent2 <- diamonds$CaratsCent^2
NullModel <- lm(log(Price)~1,data=diamonds)
FullModel <- lm(log(Price)~CaratsCent+CaratsCent2+
                  Color*Clarity+Color*Certification+
                  Clarity*Certification,
                data=diamonds)
Model_forward <- step(NullModel, scope = formula(FullModel),direction="forward",trace=0)
# Remove the trace=0 option if you want to function to print the entire process
# Let's see the variables the model selected
Model_forward$call
```

```
## lm(formula = log(Price) ~ CaratsCent + CaratsCent2 + Color +
##     Clarity + Certification + Color:Clarity + Color:Certification,
##     data = diamonds)
```

```
#run summary(Model_forward) to see the results of the final model
```

Back to the diamonds data

Let's do the same using BIC

```
# use k = log(n) to use BIC instead.
n <- nrow(diamonds)
Model_forward <- step(NullModel, scope = formula(FullModel),direction="forward",trace=0,
                      k = log(n))
# Let's see the variables the model selected
Model_forward$call
```

```
## lm(formula = log(Price) ~ CaratsCent + CaratsCent2 + Color +
##     Clarity, data = diamonds)
```

```
#run summary(Model_forward) to see the results of the final model
```

Back to the diamonds data

Backward selection using AIC

```
Model_backward <- step(FullModel,direction="backward",trace=0)
# Let's see the variables the model selected
Model_backward$call
```

```
## lm(formula = log(Price) ~ CaratsCent + CaratsCent2 + Color +
##     Clarity + Certification + Color:Clarity + Color:Certification,
##     data = diamonds)
```

```
#run summary(Model_backward) to see the results of the final model
```

Same result as forward selection using AIC

Back to the diamonds data

Backward selection using BIC

```
Model_backward <- step(FullModel,direction="backward",trace=0,k = log(n))
# Let's see the variables the model selected
Model_backward$call
```

```
## lm(formula = log(Price) ~ CaratsCent + CaratsCent2 + Color +
##     Clarity, data = diamonds)
```

```
#run summary(Model_backward) to see the results of the final model
```

Same result as forward selection using BIC

Back to the diamonds data

Stepwise selection using AIC

```r
Model_stepwise <- step(NullModel, scope = formula(FullModel),direction="both",trace=0)
# Let's see the variables the model selected
Model_stepwise$call
```

```
## lm(formula = log(Price) ~ CaratsCent + CaratsCent2 + Color + 
##     Clarity + Certification + Color:Clarity + Color:Certification, 
##     data = diamonds)
```

```r
#run summary(Model_backward) to see the results of the final model
```

Same result as previous results using AIC

Back to the diamonds data

Stepwise selection using BIC

```r
Model_stepwise <- step(NullModel, scope = formula(FullModel),direction="both",trace=0,
                       k = log(n))
# Let's see the variables the model selected
Model_stepwise$call
```

```
## lm(formula = log(Price) ~ CaratsCent + CaratsCent2 + Color +
##     Clarity, data = diamonds)
```

```r
#run summary(Model_backward) to see the results of the final model
```

Same result as previous results using BIC

Back to the diamonds data

Let's use the regsubsets function.

```
library(leaps)
Model_forward <- regsubsets(log(Price)~CaratsCent+CaratsCent2+Color*Clarity+
                              Color*Certification+Clarity*Certification,data=diamonds,
                            method="forward")
Select_results <- summary(Model_forward)
coef(Model_forward, which.max(Select_results$adjr2)) # Adj R-sq
```

```
## (Intercept)  CaratsCent CaratsCent2      ColorG      ColorH      ColorI
##   8.6185951   3.0050895  -2.0109553  -0.1275071  -0.2147009  -0.3185926
##  ClarityVS1  ClarityVS2 ClarityVVS2
##  -0.1688242  -0.2525954  -0.1116575
```

```
coef(Model_forward, which.min(Select_results$bic)) #BIC
```

```
## (Intercept)  CaratsCent CaratsCent2      ColorG      ColorH      ColorI
##   8.6185951   3.0050895  -2.0109553  -0.1275071  -0.2147009  -0.3185926
##  ClarityVS1  ClarityVS2 ClarityVVS2
##  -0.1688242  -0.2525954  -0.1116575
```

Back to the diamonds data

```
Model_backward <- regsubsets(log(Price)~CaratsCent+CaratsCent2+Color*Clarity+
                              Color*Certification+Clarity*Certification,data=diamonds,
                            method="backward")
Select_results <- summary(Model_backward)
coef(Model_backward, which.max(Select_results$adjr2)) # Adj R-sq
```

```
## (Intercept)  CaratsCent CaratsCent2      ColorG      ColorH      ColorI
##   8.6185951   3.0050895  -2.0109553  -0.1275071  -0.2147009  -0.3185926
##  ClarityVS1  ClarityVS2 ClarityVVS2
##  -0.1688242  -0.2525954  -0.1116575
```

```
coef(Model_backward, which.min(Select_results$bic)) #BIC
```

```
## (Intercept)  CaratsCent CaratsCent2      ColorG      ColorH      ColorI
##   8.6185951   3.0050895  -2.0109553  -0.1275071  -0.2147009  -0.3185926
##  ClarityVS1  ClarityVS2 ClarityVVS2
##  -0.1688242  -0.2525954  -0.1116575
```

What's next?

Move on to the readings for the next module!