Linear Regression

Prabin Kharel

Linear regression on "mtcars" data Let 'mpg' be the dependant variable and the rest of the variable be independent variables. Let's call this linear regression model as "lrm".

```
#linear regression model
lrm <- lm(mpg~., data=mtcars)

#Getting summary of the model
summary(lrm)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4506 -1.6044 -0.1196  1.2193  4.6271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 21 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8066
## F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

Let's check for multicollinearity and remove the variables that introduce multicollinearity in our data. Generally, variables with VIF(Variance Inflation Factor) greater than 10 are discarded.

```
#install.packages("car")
library(car)
```

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

```r
# Loading required package: carData
vif(lrm)
```

```
##       cyl      disp        hp      drat        wt      qsec        vs        am 
## 15.373833 21.620241  9.832037  3.374620 15.164887  7.527958  4.965873  4.648487 
##      gear      carb 
##  5.357452  7.908747
```

Since, there are variables with vif greater than 10, we need to remove it. But we won't remove all the variables with VIF > 10 at once, but we will do it one ofter the other. It is because those variables can have lesser VIF once the highest VIF variable is discarded.

```r
#removing the variable with highest vif (i.e, disp)
lrm1 <- lm(mpg~ cyl+hp+drat+wt+qsec+vs+am+gear+carb, data = mtcars)
summary(lrm1)
```

```
## 
## Call:
## lm(formula = mpg ~ cyl + hp + drat + wt + qsec + vs + am + gear + 
##     carb, data = mtcars)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.7863 -1.4055 -0.2635  1.2029  4.4753 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 12.55052   18.52585   0.677   0.5052  
## cyl          0.09627    0.99715   0.097   0.9240  
## hp          -0.01295    0.01834  -0.706   0.4876  
## drat         0.92864    1.60794   0.578   0.5694  
## wt          -2.62694    1.19800  -2.193   0.0392 *
## qsec         0.66523    0.69335   0.959   0.3478  
## vs           0.16035    2.07277   0.077   0.9390  
## am           2.47882    2.03513   1.218   0.2361  
## gear         0.74300    1.47360   0.504   0.6191  
## carb        -0.61686    0.60566  -1.018   0.3195  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.623 on 22 degrees of freedom
## Multiple R-squared:  0.8655, Adjusted R-squared:  0.8105 
## F-statistic: 15.73 on 9 and 22 DF,  p-value: 1.183e-07
```

```r
#checking multicollinearity again to ensure there are no other variables with vif>10
vif(lrm1)
```

```
##       cyl        hp      drat        wt      qsec        vs        am      gear 
## 14.284737  7.123361  3.329298  6.189050  6.914423  4.916053  4.645108  5.324402 
##      carb 
##  4.310597
```

We now have one variable "cyl" with VIF>10. Remember, we had three of them earlier. If we had removed all three then it would have resulted in loss of data as now we found out removing only two of them is okay.

```r
#removing the variable with highest vif (i.e, cyl)
lrm2 <- lm(mpg~hp+drat+wt+qsec+vs+am+gear+carb, data = mtcars)
```

```
summary(lrm2)
```

```
##
## Call:
## lm(formula = mpg ~ hp + drat + wt + qsec + vs + am + gear + carb,
##     data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8187 -1.3903 -0.3045  1.2269  4.5183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.80810   12.88582   1.072   0.2950
## hp          -0.01225    0.01649  -0.743   0.4650
## drat         0.88894    1.52061   0.585   0.5645
## wt          -2.60968    1.15878  -2.252   0.0342 *
## qsec         0.63983    0.62752   1.020   0.3185
## vs           0.08786    1.88992   0.046   0.9633
## am           2.42418    1.91227   1.268   0.2176
## gear         0.69390    1.35294   0.513   0.6129
## carb        -0.61286    0.59109  -1.037   0.3106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.566 on 23 degrees of freedom
## Multiple R-squared:  0.8655, Adjusted R-squared:  0.8187
## F-statistic:  18.5 on 8 and 23 DF,  p-value: 2.627e-08
```

Multiple Linear regression and validation using training and testing set Now, that we know that removing "disp" and "cyl" solves the multicollinearity issue we form a dataframe that is rid of these variables and split it into training and testing data.

```
mt_cars <- mtcars[,-c(2,3)]
str(mt_cars)
```

```
## 'data.frame':    32 obs. of  9 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

#####Splitting data into training and testing sets

```
#setting seed
set.seed(1234)

#splitting data into training and testing set
ind <- sample(2,nrow(mt_cars), replace=T, prob = c(0.7,0.3))
head(train_data <- mt_cars[ind==1,])
```

```
##                   mpg  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710     22.8  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive 21.4 110 3.08 3.215 19.44  1  0    3    1
## Valiant        18.1 105 2.76 3.460 20.22  1  0    3    1
## Duster 360     14.3 245 3.21 3.570 15.84  0  0    3    4
```

```r
head(test_data <- mt_cars[ind==2,])
```

```
##                         mpg  hp drat    wt  qsec vs am gear carb
## Hornet Sportabout   18.7 175 3.15 3.440 17.02  0  0    3    2
## Merc 450SLC         15.2 180 3.07 3.780 18.00  0  0    3    3
## Lincoln Continental 10.4 215 3.00 5.424 17.82  0  0    3    4
## Fiat X1-9           27.3  66 4.08 1.935 18.90  1  1    4    1
## Lotus Europa        30.4 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8 264 4.22 3.170 14.50  0  1    5    4
```

#Training the model

```r
#loading required library
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
#fitting multiple linear regression in Training set

lm1 <- train(mpg~hp+drat+wt+qsec+vs+am+gear+carb, data = train_data, method="lm")
lm1
```

```
## Linear Regression
##
## 26 samples
##  8 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 26, 26, 26, 26, 26, 26, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   4.394714  0.6976115  3.454136
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

#####Prediction on testing data

```r
#Making predictions on test data with regression model done on train data
predict_test <- predict(lm1, newdata = test_data)
predict_test
```

```
##   Hornet Sportabout         Merc 450SLC Lincoln Continental           Fiat X1-9
##            17.27108            16.26627            11.88122            28.13255
##        Lotus Europa      Ford Pantera L
##            26.30502            20.66828
```

#####Error Metrics

```r
#Checking the errors in predicted data
R2 <- R2(predict_test,test_data$mpg)
RMSE <- RMSE(predict_test,test_data$mpg)
MAE <- MAE(predict_test,test_data$mpg)
R2
```

```
## [1] 0.8603958
```

```r
RMSE
```

```
## [1] 2.784927
```

```r
MAE
```

```
## [1] 2.295371
```

####Leave One Out Cross-Validation (LOOCV: #####Training the model

```r
set.seed(1234)
train_control_1 <- trainControl(method="LOOCV")
lm2 <- train(mpg~hp+drat+wt+qsec+vs+am+gear+carb, data = train_data, method="lm", trControl= train_cont
lm2
```

```
## Linear Regression
##
## 26 samples
##  8 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 25, 25, 25, 25, 25, 25, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   3.556472  0.6657164  2.942628
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

#####Making Predictions on test data

```r
#predictions on test data with regression model done on train data using LOOCV method
predict_test_1 <- predict(lm2,newdata = test_data)
predict_test_1
```

```
##   Hornet Sportabout         Merc 450SLC Lincoln Continental           Fiat X1-9
##            17.27108            16.26627            11.88122            28.13255
##        Lotus Europa      Ford Pantera L
##            26.30502            20.66828
```

#####Error Metrics

```r
R2 <- R2(predict_test_1,test_data$mpg)
RMSE <- RMSE(predict_test_1,test_data$mpg)
MAE <-MAE(predict_test_1,test_data$mpg)
R2
```

```
## [1] 0.8603958
```

```r
RMSE
```

```
## [1] 2.784927
```

```
MAE
```

```
## [1] 2.295371
```

####k-folds cross validation #####Training the model

```
#we need to state the method as "cv" to use cross-validation control
set.seed(1234)
train_control_2 <- trainControl(method = "cv", number=10)
lm3 <- train(mpg~hp+drat+wt+qsec+vs+am+gear+carb,data= train_data, method="lm", trControl=train_control_
lm3
```

```
## Linear Regression
##
## 26 samples
##  8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 23, 24, 23, 23, 23, 24, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   3.961679  0.9588584  3.475344
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

#####Prediction on testing set

```
#making predictions on test data with cross validation as train control method
predict_test_2 <- predict(lm3,newdata = test_data)
predict_test_2
```

```
##   Hornet Sportabout         Merc 450SLC Lincoln Continental           Fiat X1-9
##            17.27108            16.26627            11.88122            28.13255
##        Lotus Europa      Ford Pantera L
##            26.30502            20.66828
```

#####Error metrics

```
#Checking errors in prediction
R2 <-R2(predict_test_2,test_data$mpg)
RMSE <-RMSE(predict_test_2,test_data$mpg)
MAE <- MAE(predict_test_2,test_data$mpg)
R2
```

```
## [1] 0.8603958
```

```
RMSE
```

```
## [1] 2.784927
```

```
MAE
```

```
## [1] 2.295371
```

####k-folds cross validation with repeats #####Training the model

```
set.seed(1234)
train_control_3 <-trainControl(method = "repeatedcv", number=10,repeats=3)
```

```r
lm4 <-train(mpg~hp+drat+wt+qsec+vs+am+gear+carb, data=train_data, method="lm",trControl=train_control_3)
lm4
```

```
## Linear Regression
##
## 26 samples
##  8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 23, 24, 23, 23, 23, 24, ...
## Resampling results:
##
##   RMSE     Rsquared   MAE
##   3.47956  0.8321443  3.082231
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

#####Prediction on testing set

```r
#predicting on test data with 10-folds cross validation with 3 repeats
predict_test_3 <- predict(lm4, newdata = test_data)
predict_test_3
```

```
##   Hornet Sportabout         Merc 450SLC Lincoln Continental           Fiat X1-9
##            17.27108            16.26627            11.88122            28.13255
##        Lotus Europa      Ford Pantera L
##            26.30502            20.66828
```

#####Error Metrics

```r
#Checking errors on prediction with 10 folds cross validation with 3 repeats
R2 <-R2(predict_test_3,test_data$mpg)
RMSE <- RMSE(predict_test_3,test_data$mpg)
MAE <- MAE(predict_test_3,test_data$mpg)
R2
```

```
## [1] 0.8603958
```

```r
RMSE
```

```
## [1] 2.784927
```

```r
MAE
```

```
## [1] 2.295371
```

For a better model, we select those models with higher R-squared error and lower Root Mean Squared Error. Among the models we created, the linear regession model with 1o folds cross validation has the highest R-squared value and lower RMSE . So, 10-folds cross validation is our best model.