Linear Regression Prediction

Veerasak Kritsanapraphan

2/1/2020

### Loading Required R packages

library(tidyverse)  
library(modelr)  
library(broom)  
library(dplyr)  
library(ggplot2)

#### Load the data

data("Wage", package = "ISLR")

### Data Sampling

set.seed(123)  
index <- sample(2, nrow(Wage), replace=TRUE, prob=c(0.7,0.3) )  
traindata <- Wage[index==1,]  
testdata <- Wage[index==2,]  
sprintf("Number of Record in Training Dataset is %d" , nrow(traindata))

## [1] "Number of Record in Training Dataset is 2101"

sprintf("Number of Record in Testing Dataset is %d" , nrow(testdata))

## [1] "Number of Record in Testing Dataset is 899"

### Compute linear regression model:

### Build the model

model <- lm(wage ~ age, data = traindata)

### Model performance

linearmodel <- data.frame(  
 name = 'Linear Regression model',  
 R2 = rsquare(model, data=testdata),  
 RMSE = rmse(model, data=testdata),  
 MAE = mae(model, data=testdata),  
 P.value = glance(model)$p.value  
)  
linearmodel

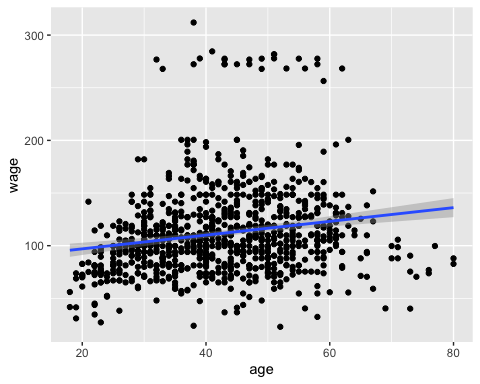
## name R2 RMSE MAE P.value  
## 1 Linear Regression model 0.03276877 41.44839 29.12218 1.061134e-20

glance(model) %>%  
 dplyr::select(r.squared, adj.r.squared, sigma, AIC, BIC, p.value)

## # A tibble: 1 x 6  
## r.squared adj.r.squared sigma AIC BIC p.value  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0406 0.0402 40.7 21541. 21558. 1.06e-20

### Visualize the data:

ggplot(testdata, aes(age, wage) ) +  
 geom\_point() +  
 stat\_smooth(method = lm, formula = y ~ x)



## Polynomial regression

lm(wage ~ age + I(age^2), data = traindata)

##   
## Call:  
## lm(formula = wage ~ age + I(age^2), data = traindata)  
##   
## Coefficients:  
## (Intercept) age I(age^2)   
## -7.30202 5.13217 -0.05111

An alternative simple solution is to use this:

lm(wage ~ poly(age, 2, raw = TRUE), data = traindata)

##   
## Call:  
## lm(formula = wage ~ poly(age, 2, raw = TRUE), data = traindata)  
##   
## Coefficients:  
## (Intercept) poly(age, 2, raw = TRUE)1   
## -7.30202 5.13217   
## poly(age, 2, raw = TRUE)2   
## -0.05111

The following example computes a sixfth-order polynomial fit:

lm(wage ~ poly(age, 6, raw = TRUE), data = traindata) %>%  
 summary()

##   
## Call:  
## lm(formula = wage ~ poly(age, 6, raw = TRUE), data = traindata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -99.019 -24.811 -4.628 15.601 201.764   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 4.284e+02 5.112e+02 0.838 0.402  
## poly(age, 6, raw = TRUE)1 -6.759e+01 7.752e+01 -0.872 0.383  
## poly(age, 6, raw = TRUE)2 4.517e+00 4.703e+00 0.960 0.337  
## poly(age, 6, raw = TRUE)3 -1.404e-01 1.464e-01 -0.959 0.338  
## poly(age, 6, raw = TRUE)4 2.258e-03 2.473e-03 0.913 0.361  
## poly(age, 6, raw = TRUE)5 -1.823e-05 2.154e-05 -0.846 0.397  
## poly(age, 6, raw = TRUE)6 5.837e-08 7.579e-08 0.770 0.441  
##   
## Residual standard error: 39.76 on 2094 degrees of freedom  
## Multiple R-squared: 0.08691, Adjusted R-squared: 0.08429   
## F-statistic: 33.22 on 6 and 2094 DF, p-value: < 2.2e-16

### Build the fifth polynomial model

model <- lm(wage ~ poly(age, 5, raw = TRUE), data = traindata)

### Model performance

polymodel <- data.frame(  
 name = 'Polynomial Regression model',  
 R2 = rsquare(model, data=testdata),  
 RMSE = rmse(model, data=testdata),  
 MAE = mae(model, data=testdata),  
 P.value = glance(model)$p.value  
)  
polymodel

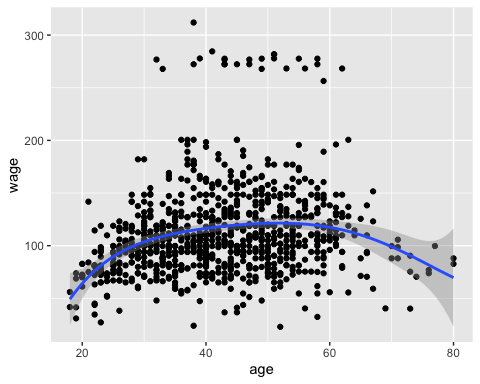
## name R2 RMSE MAE P.value  
## 1 Polynomial Regression model 0.08410985 40.33335 28.08625 3.868386e-39

glance(model) %>%  
 dplyr::select(r.squared, adj.r.squared, sigma, AIC, BIC, p.value)

## # A tibble: 1 x 6  
## r.squared adj.r.squared sigma AIC BIC p.value  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0867 0.0845 39.8 21445. 21485. 3.87e-39

Visualize the fith polynomial regression line as follow:

ggplot(testdata, aes(age, wage) ) +  
 geom\_point() +  
 stat\_smooth(method = lm, formula = y ~ poly(x, 5, raw = TRUE))



## Log transformation

### Build the model

model <- lm(wage ~ log(age), data = traindata)

### Model performance

logmodel <- data.frame(  
 name = 'Log Transform Regression model',  
 R2 = rsquare(model, data=testdata),  
 RMSE = rmse(model, data=testdata),  
 MAE = mae(model, data=testdata),  
 P.value = glance(model)$p.value  
)  
logmodel

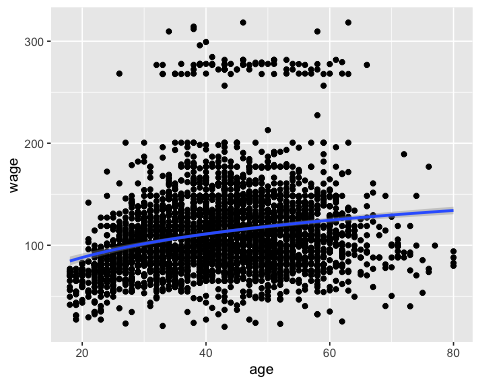
## name R2 RMSE MAE P.value  
## 1 Log Transform Regression model 0.04811535 41.11812 28.79437 1.355396e-27

glance(model) %>%  
 dplyr::select(r.squared, adj.r.squared, sigma, AIC, BIC, p.value)

## # A tibble: 1 x 6  
## r.squared adj.r.squared sigma AIC BIC p.value  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0549 0.0544 40.4 21509. 21526. 1.36e-27

Visualize the data:

ggplot(Wage, aes(age, wage) ) +  
 geom\_point() +  
 stat\_smooth(method = lm, formula = y ~ log(x))



## Spline regression

knots <- quantile(traindata$age, p = c(0.25, 0.5, 0.75))

We’ll create a model using a cubic spline (degree = 3):

library(splines)

#### Build the model

knots <- quantile(traindata$age, p = c(0.25, 0.5, 0.75))  
model <- lm (wage ~ bs(age, knots = knots), data = traindata)

#### Model performance

splinemodel <- data.frame(  
 name = 'Splines model',  
 R2 = rsquare(model, data=testdata),  
 RMSE = rmse(model, data=testdata),  
 MAE = mae(model, data=testdata),  
 P.value = glance(model)$p.value  
)  
splinemodel

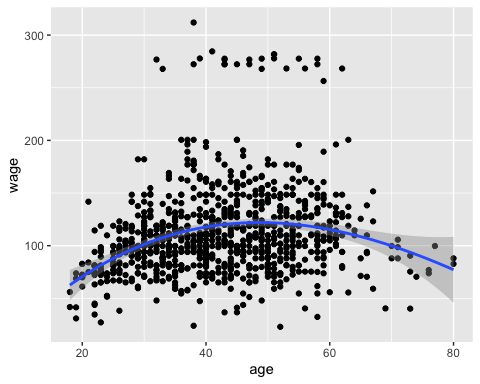
## name R2 RMSE MAE P.value  
## 1 Splines model 0.0854888 40.30298 28.06109 1.10668e-38

glance(model) %>%  
 dplyr::select(r.squared, adj.r.squared, sigma, AIC, BIC, p.value)

## # A tibble: 1 x 6  
## r.squared adj.r.squared sigma AIC BIC p.value  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0874 0.0848 39.7 21446. 21491. 1.11e-38

Visualize the cubic spline as follow:

ggplot(testdata, aes(age, wage) ) +  
 geom\_point() +  
 stat\_smooth(method = lm, formula = y ~ splines::bs(x, df = 3))



## Generalized additive models

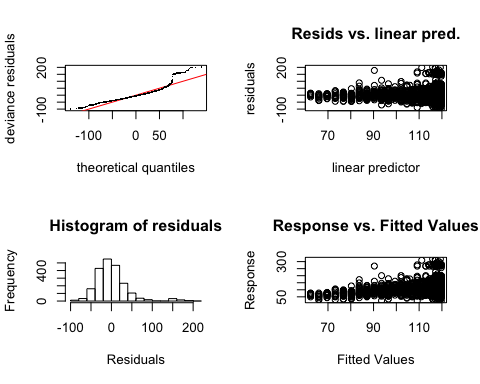
library(mgcv)

#### Build the model

model <- gam(wage ~ s(age), data = traindata)

#### Model performance

r <- capture.output(gam.check(model))



p <- strsplit(r[12], " ")[[1]][11]  
  
GAMmodel <- data.frame(  
 name = 'Generalized additive model',  
 R2 = rsquare(model, data=testdata),  
 RMSE = rmse(model, data=testdata),  
 MAE = mae(model, data=testdata),  
 P.value = p  
)  
GAMmodel

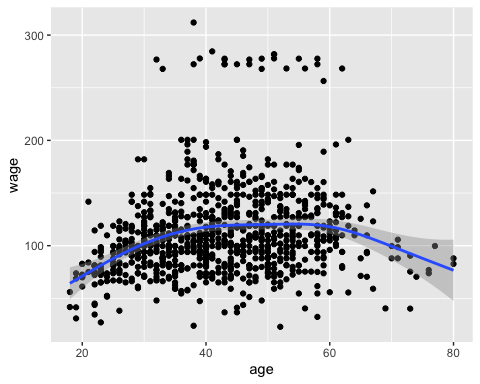
## name R2 RMSE MAE P.value  
## 1 Generalized additive model 0.08507699 40.31197 28.07897 0.56

glance(model)

## # A tibble: 1 x 6  
## df logLik AIC BIC deviance df.residual  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 5.31 -10715. 21443. 21479. 3310014. 2096.

Visualize the data:

ggplot(testdata, aes(age, wage) ) +  
 geom\_point() +  
 stat\_smooth(method = gam, formula = y ~ s(x))



## Comparing the models

totalmodel <- do.call('rbind', list(linearmodel, polymodel, splinemodel, GAMmodel))  
totalmodel

## name R2 RMSE MAE  
## 1 Linear Regression model 0.03276877 41.44839 29.12218  
## 2 Polynomial Regression model 0.08410985 40.33335 28.08625  
## 3 Splines model 0.08548880 40.30298 28.06109  
## 4 Generalized additive model 0.08507699 40.31197 28.07897  
## P.value  
## 1 1.06113393579415e-20  
## 2 3.86838574340366e-39  
## 3 1.10668018705477e-38  
## 4 0.56