Ridge & Lasso reg

2024-04-02

The difference between Lasso and Ridge regression is lasso(alpha=1)

df <- read.csv("C:/Users/SCIENCE/Documents/ndata.csv")  
df

## SALES PRICEEACH ORDERNUMBER QUANTITYORDERED  
## 1 2871.00 95.70 10107 30  
## 2 2765.90 81.35 10121 34  
## 3 3884.34 94.74 10134 41  
## 4 3746.70 83.26 10145 45  
## 5 5205.27 100.00 10159 49  
## 6 3479.76 96.66 10168 36  
## 7 2497.77 86.13 10180 29  
## 8 5512.32 100.00 10188 48  
## 9 2168.54 98.57 10201 22  
## 10 4708.44 100.00 10211 41  
## 11 3965.66 100.00 10223 37  
## 12 2333.12 100.00 10237 23  
## 13 3188.64 100.00 10251 28  
## 14 3676.76 100.00 10263 34  
## 15 4177.35 92.83 10275 45  
## 16 4099.68 100.00 10285 36  
## 17 2597.39 100.00 10299 23  
## 18 4394.38 100.00 10309 41  
## 19 4358.04 94.74 10318 46  
## 20 4396.14 100.00 10329 42  
## 21 7737.93 100.00 10341 41  
## 22 1451.00 72.55 10361 20  
## 23 733.11 34.91 10375 21  
## 24 3207.12 76.36 10388 42  
## 25 2434.56 100.00 10403 24  
## 26 7516.08 100.00 10417 66  
## 27 5404.62 100.00 10103 26  
## 28 7209.11 100.00 10112 29  
## 29 7329.06 100.00 10126 38  
## 30 7374.10 100.00 10140 37  
## 31 10993.50 100.00 10150 45  
## 32 4860.24 100.00 10163 21  
## 33 8014.82 100.00 10174 34  
## 34 5372.57 100.00 10183 23  
## 35 7290.36 100.00 10194 42  
## 36 9064.89 100.00 10206 47  
## 37 6075.30 100.00 10215 35  
## 38 6463.23 100.00 10228 29  
## 39 6120.34 100.00 10245 34  
## 40 7680.64 100.00 10258 32  
## 41 4905.39 100.00 10270 21  
## 42 8014.82 100.00 10280 34  
## 43 7136.19 100.00 10291 37  
## 44 10172.70 100.00 10304 47  
## 45 11623.70 100.00 10312 48  
## 46 6000.40 100.00 10322 40  
## 47 3003.00 100.00 10333 26  
## 48 3944.70 100.00 10347 30  
## 49 5691.84 100.00 10357 32  
## 50 4514.92 100.00 10369 41  
## 51 8254.80 100.00 10381 36  
## 52 2416.56 100.00 10391 24  
## 53 4140.23 100.00 10411 23  
## 54 12001.00 100.00 10424 50  
## 55 3896.49 99.91 10107 39  
## 56 2793.86 96.34 10120 29  
## 57 3307.77 100.00 10134 27  
## 58 5192.95 100.00 10145 37  
## 59 5016.83 100.00 10159 37  
## 60 3660.93 100.00 10168 27  
## 61 4695.60 100.00 10180 42  
## 62 3660.92 96.34 10188 38  
## 63 3025.92 100.00 10201 24  
## 64 3009.09 100.00 10210 23  
## 65 5422.39 100.00 10223 47  
## 66 2852.08 100.00 10236 22  
## 67 5756.52 100.00 10251 44  
## 68 4472.00 100.00 10263 40  
## 69 2904.44 100.00 10275 22

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.3.3

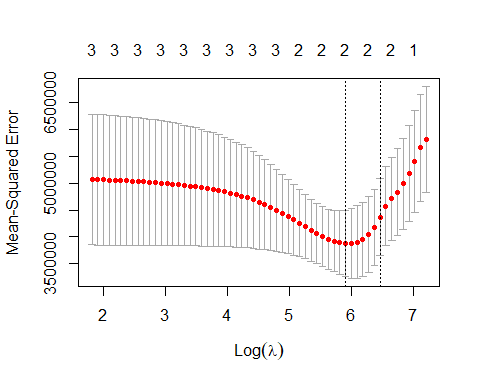
## Loading required package: Matrix

## Loaded glmnet 4.1-8

x <- data.matrix(df[,c("ORDERNUMBER", "QUANTITYORDERED", "PRICEEACH")])  
y<-df$SALES  
x

## ORDERNUMBER QUANTITYORDERED PRICEEACH  
## [1,] 10107 30 95.70  
## [2,] 10121 34 81.35  
## [3,] 10134 41 94.74  
## [4,] 10145 45 83.26  
## [5,] 10159 49 100.00  
## [6,] 10168 36 96.66  
## [7,] 10180 29 86.13  
## [8,] 10188 48 100.00  
## [9,] 10201 22 98.57  
## [10,] 10211 41 100.00  
## [11,] 10223 37 100.00  
## [12,] 10237 23 100.00  
## [13,] 10251 28 100.00  
## [14,] 10263 34 100.00  
## [15,] 10275 45 92.83  
## [16,] 10285 36 100.00  
## [17,] 10299 23 100.00  
## [18,] 10309 41 100.00  
## [19,] 10318 46 94.74  
## [20,] 10329 42 100.00  
## [21,] 10341 41 100.00  
## [22,] 10361 20 72.55  
## [23,] 10375 21 34.91  
## [24,] 10388 42 76.36  
## [25,] 10403 24 100.00  
## [26,] 10417 66 100.00  
## [27,] 10103 26 100.00  
## [28,] 10112 29 100.00  
## [29,] 10126 38 100.00  
## [30,] 10140 37 100.00  
## [31,] 10150 45 100.00  
## [32,] 10163 21 100.00  
## [33,] 10174 34 100.00  
## [34,] 10183 23 100.00  
## [35,] 10194 42 100.00  
## [36,] 10206 47 100.00  
## [37,] 10215 35 100.00  
## [38,] 10228 29 100.00  
## [39,] 10245 34 100.00  
## [40,] 10258 32 100.00  
## [41,] 10270 21 100.00  
## [42,] 10280 34 100.00  
## [43,] 10291 37 100.00  
## [44,] 10304 47 100.00  
## [45,] 10312 48 100.00  
## [46,] 10322 40 100.00  
## [47,] 10333 26 100.00  
## [48,] 10347 30 100.00  
## [49,] 10357 32 100.00  
## [50,] 10369 41 100.00  
## [51,] 10381 36 100.00  
## [52,] 10391 24 100.00  
## [53,] 10411 23 100.00  
## [54,] 10424 50 100.00  
## [55,] 10107 39 99.91  
## [56,] 10120 29 96.34  
## [57,] 10134 27 100.00  
## [58,] 10145 37 100.00  
## [59,] 10159 37 100.00  
## [60,] 10168 27 100.00  
## [61,] 10180 42 100.00  
## [62,] 10188 38 96.34  
## [63,] 10201 24 100.00  
## [64,] 10210 23 100.00  
## [65,] 10223 47 100.00  
## [66,] 10236 22 100.00  
## [67,] 10251 44 100.00  
## [68,] 10263 40 100.00  
## [69,] 10275 22 100.00

cv\_model <- cv.glmnet(x,y,alpha=1)  
plot(cv\_model)



lambda <- cv\_model$lambda.min  
lambda

## [1] 368.3179

model <- glmnet(x,y,alpha=1)  
coef(model)

## 4 x 59 sparse Matrix of class "dgCMatrix"

## [[ suppressing 59 column names 's0', 's1', 's2' ... ]]

##   
## (Intercept) 5041.273 4595.2514 4188.85342 3818.55869 3481.15991 3173.73472  
## ORDERNUMBER . . . . . .   
## QUANTITYORDERED . 12.8071 24.47646 35.10914 44.79725 53.62469  
## PRICEEACH . . . . . .   
##   
## (Intercept) 2666.57079 1823.56921 1055.45774 355.58319 -282.11645  
## ORDERNUMBER . . . . .   
## QUANTITYORDERED 61.28462 67.62106 73.39457 78.65518 83.44845  
## PRICEEACH 2.47560 8.88428 14.72363 20.04423 24.89217  
##   
## (Intercept) -863.16460 -1392.59404 -1874.99042 -2314.53204 -2715.02599  
## ORDERNUMBER . . . . .   
## QUANTITYORDERED 87.81590 91.79536 95.42129 98.72511 101.73542  
## PRICEEACH 29.30942 33.33426 37.00155 40.34304 43.38768  
##   
## (Intercept) -3079.94117 -3412.43828 -3715.39727 -3991.44222 -4242.9641  
## ORDERNUMBER . . . . .   
## QUANTITYORDERED 104.47831 106.97753 109.25472 111.32961 113.2202  
## PRICEEACH 46.16185 48.68956 50.99273 53.09128 55.0034  
##   
## (Intercept) -4472.14146 -4680.95934 -4871.22641 -5044.59069 -6.228002e+03  
## ORDERNUMBER . . . . 9.889004e-02  
## QUANTITYORDERED 114.94279 116.51237 117.94251 119.24560 1.203455e+02  
## PRICEEACH 56.74566 58.33314 59.77959 61.09755 6.245766e+01  
##   
## (Intercept) -7876.4726405 -9377.9389673 -1.074602e+04 -1.199256e+04  
## ORDERNUMBER 0.2439851 0.3761354 4.965458e-01 6.062593e-01  
## QUANTITYORDERED 121.2972606 122.1650668 1.229558e+02 1.236762e+02  
## PRICEEACH 63.7857553 64.9957006 6.609816e+01 6.710268e+01  
##   
## (Intercept) -1.312837e+04 -1.416327e+04 -1.510623e+04 -1.596543e+04  
## ORDERNUMBER 7.062262e-01 7.973123e-01 8.803065e-01 9.559278e-01  
## QUANTITYORDERED 1.243327e+02 1.249309e+02 1.254759e+02 1.259725e+02  
## PRICEEACH 6.801795e+01 6.885192e+01 6.961180e+01 7.030418e+01  
##   
## (Intercept) -16748.293510 -17461.611589 -18111.560455 -18703.769655  
## ORDERNUMBER 1.024831 1.087613 1.144818 1.196941  
## QUANTITYORDERED 126.424930 126.837208 127.212860 127.555141  
## PRICEEACH 70.935046 71.509868 72.033624 72.510851  
##   
## (Intercept) -19243.368623 -19733.41950 -20181.541811 -20589.859555  
## ORDERNUMBER 1.244433 1.28755 1.326992 1.362931  
## QUANTITYORDERED 127.867015 128.15197 128.410824 128.646681  
## PRICEEACH 72.945683 73.34153 73.702566 74.031529  
##   
## (Intercept) -20961.903494 -21300.896074 -21609.773489 -21891.211085  
## ORDERNUMBER 1.395677 1.425514 1.452701 1.477472  
## QUANTITYORDERED 128.861585 129.057397 129.235815 129.398382  
## PRICEEACH 74.331268 74.604379 74.853228 75.079969  
##   
## (Intercept) -22147.646540 -22381.300977 -22594.198197 -22788.182215  
## ORDERNUMBER 1.500043 1.520608 1.539347 1.556421  
## QUANTITYORDERED 129.546507 129.681473 129.804449 129.916500  
## PRICEEACH 75.286568 75.474813 75.646334 75.802618  
##   
## (Intercept) -22964.933228 -23125.982167 -23272.723963 -23406.429622  
## ORDERNUMBER 1.571978 1.586153 1.599069 1.610837  
## QUANTITYORDERED 130.018596 130.111623 130.196386 130.273618  
## PRICEEACH 75.945019 76.074768 76.192992 76.300712  
##   
## (Intercept) -23528.25724 -23639.262026 -23740.405454 -23829.328358  
## ORDERNUMBER 1.62156 1.631331 1.640233 1.648029  
## QUANTITYORDERED 130.34399 130.408110 130.466533 130.522094  
## PRICEEACH 76.39886 76.488295 76.569782 76.643196  
##   
## (Intercept) -23913.479874  
## ORDERNUMBER 1.655437  
## QUANTITYORDERED 130.570478  
## PRICEEACH 76.710893

model1 <- glmnet(x,y,alpha=1, lambda=lambda)  
coef(model1)

## 4 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) -2314.53194  
## ORDERNUMBER .   
## QUANTITYORDERED 98.72510  
## PRICEEACH 40.34304

library(caTools)

## Warning: package 'caTools' was built under R version 4.3.3

data <- subset(df, select=c("SALES","ORDERNUMBER","QUANTITYORDERED","PRICEEACH"))  
set.seed(2)  
split <- sample.split(data, SplitRatio=0.7)  
train <- subset(data, split="TRUE")

## Warning: In subset.data.frame(data, split = "TRUE") :  
## extra argument 'split' will be disregarded

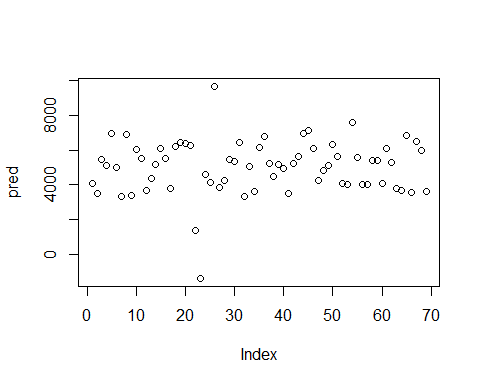
test <- subset(data, split="FALSE")

## Warning: In subset.data.frame(data, split = "FALSE") :  
## extra argument 'split' will be disregarded

test

## SALES ORDERNUMBER QUANTITYORDERED PRICEEACH  
## 1 2871.00 10107 30 95.70  
## 2 2765.90 10121 34 81.35  
## 3 3884.34 10134 41 94.74  
## 4 3746.70 10145 45 83.26  
## 5 5205.27 10159 49 100.00  
## 6 3479.76 10168 36 96.66  
## 7 2497.77 10180 29 86.13  
## 8 5512.32 10188 48 100.00  
## 9 2168.54 10201 22 98.57  
## 10 4708.44 10211 41 100.00  
## 11 3965.66 10223 37 100.00  
## 12 2333.12 10237 23 100.00  
## 13 3188.64 10251 28 100.00  
## 14 3676.76 10263 34 100.00  
## 15 4177.35 10275 45 92.83  
## 16 4099.68 10285 36 100.00  
## 17 2597.39 10299 23 100.00  
## 18 4394.38 10309 41 100.00  
## 19 4358.04 10318 46 94.74  
## 20 4396.14 10329 42 100.00  
## 21 7737.93 10341 41 100.00  
## 22 1451.00 10361 20 72.55  
## 23 733.11 10375 21 34.91  
## 24 3207.12 10388 42 76.36  
## 25 2434.56 10403 24 100.00  
## 26 7516.08 10417 66 100.00  
## 27 5404.62 10103 26 100.00  
## 28 7209.11 10112 29 100.00  
## 29 7329.06 10126 38 100.00  
## 30 7374.10 10140 37 100.00  
## 31 10993.50 10150 45 100.00  
## 32 4860.24 10163 21 100.00  
## 33 8014.82 10174 34 100.00  
## 34 5372.57 10183 23 100.00  
## 35 7290.36 10194 42 100.00  
## 36 9064.89 10206 47 100.00  
## 37 6075.30 10215 35 100.00  
## 38 6463.23 10228 29 100.00  
## 39 6120.34 10245 34 100.00  
## 40 7680.64 10258 32 100.00  
## 41 4905.39 10270 21 100.00  
## 42 8014.82 10280 34 100.00  
## 43 7136.19 10291 37 100.00  
## 44 10172.70 10304 47 100.00  
## 45 11623.70 10312 48 100.00  
## 46 6000.40 10322 40 100.00  
## 47 3003.00 10333 26 100.00  
## 48 3944.70 10347 30 100.00  
## 49 5691.84 10357 32 100.00  
## 50 4514.92 10369 41 100.00  
## 51 8254.80 10381 36 100.00  
## 52 2416.56 10391 24 100.00  
## 53 4140.23 10411 23 100.00  
## 54 12001.00 10424 50 100.00  
## 55 3896.49 10107 39 99.91  
## 56 2793.86 10120 29 96.34  
## 57 3307.77 10134 27 100.00  
## 58 5192.95 10145 37 100.00  
## 59 5016.83 10159 37 100.00  
## 60 3660.93 10168 27 100.00  
## 61 4695.60 10180 42 100.00  
## 62 3660.92 10188 38 96.34  
## 63 3025.92 10201 24 100.00  
## 64 3009.09 10210 23 100.00  
## 65 5422.39 10223 47 100.00  
## 66 2852.08 10236 22 100.00  
## 67 5756.52 10251 44 100.00  
## 68 4472.00 10263 40 100.00  
## 69 2904.44 10275 22 100.00

linear <- glm(df$SALES~., data = train)  
pred <- predict(linear, newdata = test)  
plot(pred)



library(Metrics)

## Warning: package 'Metrics' was built under R version 4.3.3

rmse(pred, test$SALES)

## [1] 1818.152

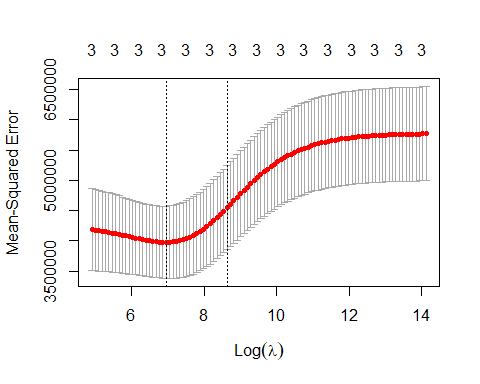
summary(linear)

##   
## Call:  
## glm(formula = df$SALES ~ ., data = train)  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -24777.787 26553.356 -0.933 0.3542   
## ORDERNUMBER 1.732 2.550 0.679 0.4995   
## QUANTITYORDERED 131.066 24.403 5.371 1.13e-06 \*\*\*  
## PRICEEACH 77.406 24.915 3.107 0.0028 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 3509104)  
##   
## Null deviance: 388713702 on 68 degrees of freedom  
## Residual deviance: 228091730 on 65 degrees of freedom  
## AIC: 1241.6  
##   
## Number of Fisher Scoring iterations: 2

library(Metrics)  
pred1 <- predict(model1, x)  
rmse(pred1, y)

## [1] 1887.778

cv\_model1 <- cv.glmnet(x,y,alpha=0)  
plot(cv\_model1)



lambda1 <- cv\_model1$lambda.min  
lambda1

## [1] 1048.983

model2 <- glmnet(x,y,alpha=1, lambda=lambda1)  
coef(model2)

## 4 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 3907.92816  
## ORDERNUMBER .   
## QUANTITYORDERED 32.54298  
## PRICEEACH .

library(Metrics)  
pred2 <- predict(model2, x)  
rmse(pred2, y)

## [1] 2213.228

summary(pred2)

## s0   
## Min. :4559   
## 1st Qu.:4787   
## Median :5079   
## Mean :5041   
## 3rd Qu.:5242   
## Max. :6056



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.