Estátistica II

Lista 2

Problema

Estimar três modelos (Ridge, Lasso e Elasticnet) para explicar a variável Y (price), as demais variáveis da base de dados são todas variáveis explicativas; particione a base de dados em 80% para treino e 20% para teste; e apresente os resultados:

Separando o treino

```
> set.seed(1)
>
> indices <- createDataPartition(dataset$price, p=0.8, list=F)
> treino <- dataset[indices,]
> teste <- dataset[-indices,]
```

Alterando escala das variáveis

```
> cols <- c('price', 'age', 'parea', 'tarea', 'bath',
+            'ensuit', 'garag', 'plaz','park', 'trans',
+            'kidca', 'school', 'health', 'bike')
>
> pre_proc_val <- preProcess(treino[,cols], method = c("center", "scale"))
>
> treino[,cols] <- predict(pre_proc_val, treino[,cols])
> teste[,cols] <- predict(pre_proc_val, teste[,cols])
>
> print("valores de treino")
[1] "valores de treino"
> summary(treino)
```

```
> print("valores de treino")
[1] "valores de treino"
> summary(treino)
     price              age              parea               tarea               bath             ensuit
 Min.   :-1.3512   Min.   :-1.0471   Min.   :-2.17312   Min.   :-1.872522   Min.   :-2.5996   Min.   :-1.5890
 1st Qu.:-0.7324   1st Qu.:-0.8890   1st Qu.:-0.83396   1st Qu.:-0.882164   1st Qu.:-0.8758   1st Qu.:-0.4978
 Median :-0.1389   Median :-0.3354   Median :-0.04264   Median : 0.005744   Median :-0.0139   Median :-0.4978
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.4155   3rd Qu.: 0.7716   3rd Qu.: 0.74108   3rd Qu.: 0.782663   3rd Qu.: 0.8480   3rd Qu.: 0.5934
 Max.   : 6.1479   Max.   : 2.9857   Max.   : 2.39220   Max.   : 3.762277   Max.   : 2.5718   Max.   : 1.6845
     garag              plaz              park              trans              kidca              school
 Min.   :-2.7684   Min.   :-1.6640   Min.   :-2.2628   Min.   :-2.4925   Min.   :-3.2493   Min.   :-2.06525
 1st Qu.:-1.2590   1st Qu.:-0.8550   1st Qu.:-0.7801   1st Qu.:-0.7636   1st Qu.:-0.6062   1st Qu.:-0.80119
 Median : 0.2504   Median :-0.1820   Median : 0.2407   Median : 0.2202   Median : 0.2238   Median :-0.01037
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
 3rd Qu.: 0.2504   3rd Qu.: 0.7331   3rd Qu.: 0.8295   3rd Qu.: 0.8089   3rd Qu.: 0.7240   3rd Qu.: 0.62998
 Max.   : 3.2692   Max.   : 3.2624   Max.   : 1.8369   Max.   : 1.4371   Max.   : 2.0784   Max.   : 3.47389
     health             bike              barb            balc             elev             fitg            party
 Min.   :-1.7846   Min.   :-1.7295   Min.   :0.00   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:-0.7214   1st Qu.:-0.7519   1st Qu.:0.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median :-0.2081   Median :-0.1259   Median :0.00   Median :0.0000   Median :0.0000   Median :0.0000   Median :1.0000
 Mean   : 0.0000   Mean   : 0.0000   Mean   :0.47   Mean   :0.4401   Mean   :0.2972   Mean   :0.2926   Mean   :0.5369
 3rd Qu.: 0.5379   3rd Qu.: 0.7515   3rd Qu.:1.00   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   : 3.9310   Max.   : 3.5095   Max.   :1.00   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
     categ
 Min.   :0.0000
 1st Qu.:1.0000
 Median :1.0000
 Mean   :0.9562
 3rd Qu.:1.0000
 Max.   :1.0000

[1] "valores de teste"
> summary(teste)
     price              age               parea               tarea              bath              ensuit
 Min.   :-1.2572   Min.   :-1.04711   Min.   :-1.68615   Min.   :-1.68470   Min.   :-2.5996   Min.   :-1.588981
 1st Qu.:-0.7310   1st Qu.:-0.88896   1st Qu.:-0.84918   1st Qu.:-0.80533   1st Qu.:-0.8758   1st Qu.:-0.497814
 Median :-0.1577   Median :-0.49358   Median :-0.04264   Median :-0.07963   Median :-0.0139   Median :-0.497814
 Mean   :-0.0544   Mean   :-0.08712   Mean   :-0.03041   Mean   :-0.06399   Mean   :-0.1508   Mean   :-0.008318
 3rd Qu.: 0.3500   3rd Qu.: 0.73209   3rd Qu.: 0.67260   3rd Qu.: 0.64606   3rd Qu.: 0.8480   3rd Qu.: 0.593354
 Max.   : 2.3357   Max.   : 2.27406   Max.   : 1.90523   Max.   : 1.91816   Max.   : 1.7099   Max.   : 1.684521
     garag               plaz               park              trans               kidca               school
 Min.   :-2.76839   Min.   :-1.67190   Min.   :-1.7396   Min.   :-2.19678   Min.   :-3.00038   Min.   :-1.9975
 1st Qu.:-1.25899   1st Qu.:-0.77106   1st Qu.:-0.4292   1st Qu.:-0.76358   1st Qu.:-0.51831   1st Qu.:-0.7239
 Median : 0.25041   Median :-0.02648   Median : 0.5308   Median : 0.06041   Median : 0.09492   Median :-0.1170
 Mean   : 0.08113   Mean   : 0.07582   Mean   : 0.2440   Mean   :-0.01099   Mean   :-0.02816   Mean   :-0.1620
 3rd Qu.: 0.25041   3rd Qu.: 0.92688   3rd Qu.: 0.9718   3rd Qu.: 0.77202   3rd Qu.: 0.67673   3rd Qu.: 0.2248
 Max.   : 3.26920   Max.   : 2.65944   Max.   : 1.5623   Max.   : 1.43446   Max.   : 1.47962   Max.   : 2.3377
     health             bike              barb              balc              elev              fitg
 Min.   :-1.5980   Min.   :-1.7051   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:-0.7324   1st Qu.:-0.5777   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median :-0.1705   Median : 0.1026   Median :1.0000   Median :0.0000   Median :0.0000   Median :0.0000
 Mean   : 0.1157   Mean   : 0.2439   Mean   :0.5981   Mean   :0.4766   Mean   :0.3551   Mean   :0.3738
 3rd Qu.: 0.8431   3rd Qu.: 0.9821   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   : 3.9310   Max.   : 3.1196   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
     party            categ
 Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:1.0000
 Median :1.0000   Median :1.0000
 Mean   :0.5794   Mean   :0.9533
 3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.0000
```

Função que calcula e retorna R^2 e RMSE

```
> eval_results <- function(true, predicted, df) {
+     SSE <- sum((predicted - true)^2)
+     SST <- sum((true - mean(true))^2)
+     R_square <- 1 - SSE / SST
+     RMSE = sqrt(SSE/nrow(df))
+
+     # Model performance metrics
+     data.frame(
+         RMSE = RMSE,
+         Rsquare = R_square
+     )
+ }
```

Regressão Ridge

```
> cols_reg <- c('price', 'age', 'parea', 'tarea', 'bath',
+               'ensuit', 'garag', 'plaz','park', 'trans',
+               'kidca', 'school', 'health', 'bike', 'barb',
+               'balc', 'elev', 'fitg', 'party', 'categ')
>
> dummies <- dummyVars(price ~ age+parea+tarea+bath+
+                          ensuit+garag+plaz+park+trans+kidca+
+                          school+health+bike+barb+balc+elev+fitg+
+                          party+categ,
+                      data = dataset[,cols_reg])
>
> train_dummies <- predict(dummies, newdata = treino[,cols_reg])
>
> test_dummies <- predict(dummies, newdata = teste[,cols_reg])
>
> print(dim(train_dummies)); print(dim(test_dummies))
```

```
> print(dim(train_dummies)); print(dim(test_dummies))
[1] 434  19
[1] 107  19
```

i. O valor ótimo do lambda para os modelos;

O melhor lamba 0.1

```
> x = as.matrix(train_dummies)
> y_train = treino$price
>
> x_test = as.matrix(test_dummies)
> y_test = teste$price
> lambdas <- 10^seq(2, -3, by = -.1)
> ridge_lamb <- cv.glmnet(x, y_train, alpha = 0,
+                          lambda = lambdas)
> best_lambda_ridge <- ridge_lamb$lambda.min
> best_lambda_ridge
[1] 0.1
```

ii. O valor do alpha para o modelo ElasticNet;

iii. Os valores dos parâmetros para os modelos;

```
> ridge_reg[["beta"]]
19 x 1 sparse Matrix of class "dgCMatrix"
                s0
age    -0.17994688
parea   0.15907495
tarea   0.21631677
bath    0.04261451
ensuit  0.18954877
garag   0.20506605
plaz    0.04714112
park   -0.05375536
trans   0.03345027
kidca   0.01621329
school -0.00131847
health -0.01054514
bike   -0.04790725
barb   -0.06878716
balc    0.15377909
elev   -0.18946965
fitg    0.23383601
party   0.06497685
categ   0.46258571
```

iv. O R^2 e RMSE dos modelos estimados;

```
> predictions_test <- predict(ridge_reg, s = best_lambda_ridge,
+                              newx = x_test)
> eval_results(y_test, predictions_test, test)
Error in nrow(df) : object 'test' not found
> predictions_test <- predict(ridge_reg, s = best_lambda_ridge,
+                              newx = x_test)
> eval_results(y_test, predictions_test, teste)
       RMSE   Rsquare
1 0.3282966 0.8487347
```

O R^2 está próximo de 1, mas não tão próximo, o que é um bom sinal pois significa que tem menos chances de o modelo estar sofrendo overfitting. O RMSE está próximo de zero, o que é bom sinal, significa que poucos erros foram cometidos.

v. Apresente os resultados de uma predição proposta por você mesmo para os modelos (valor estimado e intervalos de confiança).

Valores para a predição

```
> price <- (median(dataset$price)-pre_proc_val[["mean"]][["price"]])/pre_proc_val[["std"]][["price"]]
> age <- (median(dataset$age)-pre_proc_val[["mean"]][["age"]])/pre_proc_val[["std"]][["age"]]
> parea <- (median(dataset$parea)-pre_proc_val[["mean"]][["parea"]])/pre_proc_val[["std"]][["parea"]]
> tarea <- (median(dataset$tarea)-pre_proc_val[["mean"]][["tarea"]])/pre_proc_val[["std"]][["tarea"]]
> bath <- (median(dataset$bath)-pre_proc_val[["mean"]][["bath"]])/pre_proc_val[["std"]][["bath"]]
> ensuit <- (median(dataset$ensuit)-pre_proc_val[["mean"]][["ensuit"]])/pre_proc_val[["std"]][["ensuit"]]
> garag <- (median(dataset$garag)-pre_proc_val[["mean"]][["garag"]])/pre_proc_val[["std"]][["garag"]]
> plaz <- (median(dataset$plaz)-pre_proc_val[["mean"]][["plaz"]])/pre_proc_val[["std"]][["plaz"]]
> park <- (median(dataset$park)-pre_proc_val[["mean"]][["park"]])/pre_proc_val[["std"]][["park"]]
> trans <- (median(dataset$trans)-pre_proc_val[["mean"]][["trans"]])/pre_proc_val[["std"]][["trans"]]
> kidca <- (median(dataset$kidca)-pre_proc_val[["mean"]][["kidca"]])/pre_proc_val[["std"]][["kidca"]]
> school <- (median(dataset$school)-pre_proc_val[["mean"]][["school"]])/pre_proc_val[["std"]][["school"]]
> health <- (median(dataset$health)-pre_proc_val[["mean"]][["health"]])/pre_proc_val[["std"]][["health"]]
> bike <- (median(dataset$bike)-pre_proc_val[["mean"]][["bike"]])/pre_proc_val[["std"]][["bike"]]
> barb <- 0
> balc <- 0
> elev <- 0
> fitg <- 0
> party <- 0
> categ <- 0
>
> # Constuirndo matriz com dados para predição
> our_pred <- as.matrix(data.frame(age=age,
+                                  parea=parea,
+                                  tarea=tarea,
+                                  bath=bath,
+                                  ensuit=ensuit,
+                                  garag=garag,
+                                  plaz=plaz,
+                                  park=park,
+                                  trans=trans,
+                                  kidca=kidca,
+                                  school=school,
+                                  health=health,
+                                  bike=bike,
+                                  barb=barb,
+                                  balc=balc,
+                                  elev=elev,
+                                  fitg=fitg,
+                                  party=party,
+                                  categ=categ))
```

Predição Ridge

```
> predict_our_ridge <- predict(ridge_reg, s = best_lambda_ridge,
+                              newx = our_pred)
> predict_our_ridge
             s1
[1,] -0.5111749
```

Intervalo de confiança

```
> n <- nrow(treino)
> m <- predict_our_ridge
> s <- pre_proc_val[["std"]][["price"]]
> dam <- s/sqrt(n)
> CIlwr_ridge <- m + (qnorm(0.025))*dam
> CIupr_ridge <- m - (qnorm(0.025))*dam
>
> CIlwr_ridge
            s1
[1,] -50056.16
> CIupr_ridge
           s1
[1,] 50055.14
```

Nota-se que o valor obtido assim como o intervalo de confiança tiveram grande divergência com o valor esperado.

Regressão de Lasso

```
> lambdas <- 10^seq(2, -3, by = -.1)
>
> lasso_lamb <- cv.glmnet(x, y_train, alpha = 1,
+                         lambda = lambdas,
+                         standardize = TRUE, nfolds = 5)
> best_lambda_lasso <- lasso_lamb$lambda.min
> best_lambda_lasso
[1] 0.007943282
>
> lasso_model <- glmnet(x, y_train, alpha = 1,
+                       lambda = best_lambda_lasso,
+                       standardize = TRUE)
```

Melhor lambda: 0.007943282

```
> lasso_model[["beta"]]
19 x 1 sparse Matrix of class "dgCMatrix"
                 s0
age    -0.177787541
parea   0.160244253
tarea   0.239394524
bath    0.005550922
ensuit  0.215324160
garag   0.214458904
plaz    0.043167047
park   -0.056688502
trans   0.030263567
kidca   0.008624390
school  .
health -0.002502411
bike   -0.038298740
barb   -0.066286730
balc    0.147531615
elev   -0.179246307
fitg    0.240380064
party   0.043609304
categ   0.490774990
```

Testando o Modelo Lasso

```
> predictions_test <- predict(lasso_model, s = best_lambda_lasso,
+                             newx = x_test)
> eval_results(y_test, predictions_test, teste)
       RMSE   Rsquare
1 0.3313211 0.8459347
```

O R^2 está próximo de 1, mas não tão próximo, o que é um bom sinal pois significa que tem menos chances de o modelo estar sofrendo overfitting. O RMSE está menos próximo de zero do que o modelo de regressão Ridge.

Predição Lasso

```
> predict_our_lasso <- predict(lasso_model, s = best_lambda_lasso,
+                              newx = our_pred)
> predict_our_lasso
              s1
[1,] -0.5468196
>
> n <- nrow(treino)
> m <- predict_our_lasso
> s <- pre_proc_val[["std"]][["price"]]
> dam <- s/sqrt(n)
> CIlwr_lasso <- m + (qnorm(0.025))*dam
> CIupr_lasso <- m - (qnorm(0.025))*dam
>
> CIlwr_lasso
             s1
[1,] -50056.19
> CIupr_lasso
           s1
[1,] 50055.1
```

O valor obtido assim como o intervalo de confiança tiveram grande divergência com o valor esperado, mas o resultado foi melhor que o do modelo Ridge.

Regressão ElasticNet

```
> train_cont <- trainControl(method = "repeatedcv",
+                            number = 10,
+                            repeats = 5,
+                            search = "random",
+                            verboseIter = TRUE)
>
```

```
> elastic_reg <- train(price ~ age+parea+tarea+bath+ensuit+garag+plaz+park+trans+
                           kidca+school+health+bike+barb+balc+elev+fitg+party+categ,
                       data = train,
                       method = "glmnet",
                       tuneLength = 10,
                       trControl = train_cont)
```

```
> elastic_reg$bestTune
       alpha      lambda
1 0.02516404 0.002316695
```

```
> elastic_reg[["finalModel"]][["beta"]]
19 x 70 sparse Matrix of class "dgCMatrix"
   [[ suppressing 50 column names ‘s0’, ‘s1’, ‘s2’ ... ]]

age    . .           .           .          .          .           -0.007009236 -0.01637457 -0.02563691
parea  . .           0.005819123 0.01481200 0.02349956 0.03184101   0.039779582  0.04736121  0.05455519
tarea  . 0.016886342 0.029512088 0.04074504 0.05186117 0.06283176   0.073213844  0.08323505  0.09300322
bath   . .           0.005745696 0.01479389 0.02355185 0.03197556   0.039776974  0.04710588  0.05401747
ensuit . .           0.009000744 0.01932437 0.02953738 0.03960692   0.048672584  0.05716185  0.06532448
garag  . 0.004561377 0.016728786 0.02783887 0.03888944 0.04984706   0.059804547  0.06919708  0.07829788
plaz   . .           .           .          .          .           .            .           .
park   . .           .           .          .          .           .            .           .
trans  . .           .           .          .          .           .            .           .
kidca  . .           .           .          .          .           .            .           .
school . .           .           .          .          .           .            .           .
health . .           .           .          .          .           .            .           .
bike   . .           .           .          .          .           .            .           .
barb   . .           .           .          .          .           .            .           .
balc   . .           .           .          .          .           .            .           .
elev   . .           .           .          .          .           .            .           .
fitg   . .           .           .          .          .           .            .           .
party  . .           .           .          .          .           .            .           .
categ  . .           .           .          .          .           .            .           .

age    -0.03477236 -0.043573121 -0.051598926 -0.05941257 -0.06697887 -0.073776499 -0.080263292 -0.08648627
parea   0.06134721  0.067742459  0.073734751  0.07935013  0.08457870  0.089427892  0.093892967  0.09801571
tarea   0.10250962  0.111671471  0.120445605  0.12892899  0.13714142  0.144936369  0.152488749  0.15978762
bath    0.06049863  0.066400667  0.071530561  0.07618631  0.08038434  0.083909083  0.086983102  0.08964726
ensuit  0.07314368  0.080417798  0.086788801  0.09276045  0.09833191  0.103450579  0.108223938  0.11265187
garag   0.08708289  0.095402187  0.103027470  0.11027904  0.11715480  0.123279281  0.129036420  0.13442224
plaz    .           .            .            .           .           .            .            .
park    .           .            .            .           .           .            .            .
trans   .           .            .            .           .           .            .            .
kidca   .           .            .            .           .           .            .            .
school  .           .            .            .           .           .            .            .
health  .           .            .            .           .           .            .            .
bike    .           .            .            .           .           .            .            .
barb    .           .            .            .           .           .            .            .
balc    .           .            .            .           .           0.004326172  0.008545438  0.01249279
elev    .           .            .            .           .           .            .            .
fitg    .           0.002069965  0.009473501  0.01670793  0.02375965  0.029790348  0.035569650  0.04112360
party   .           .            .            .           .           .            .            .
categ   .           .            .            .           .           .            .            .

age    -0.09243051 -0.09809461 -0.103606179 -0.109124893 -0.114373903 -0.119374025 -0.12412446 -0.12860437
parea   0.10179507  0.10524250  0.108144764  0.110627403  0.112768847  0.114643843  0.11626960  0.11756785
tarea   0.16684978  0.17368380  0.179921670  0.185474860  0.190811996  0.195927190  0.20083035  0.20550250
bath    0.09191661  0.09381279  0.095110336  0.095746122  0.096088131  0.096133267  0.09590775  0.09549211
ensuit  0.11675637  0.12055554  0.124345527  0.128251046  0.131970619  0.135526194  0.13893196  0.14220791
garag   0.13945292  0.14413906  0.148384032  0.152158201  0.155656130  0.158877939  0.16184010  0.16459193
plaz    .           .           .            .            .            .            .           .
park    .           .          -0.002908972 -0.009188567 -0.015272199 -0.021148153 -0.02680627 -0.03224544
trans   .           .           .            .            .            .            .           .
kidca   .           .           .            .            .            .            .           .
school  .           .           .            .            .            .            .           .
health  .           .           .            .            .            .            .           .
bike    .           .           .            .            .            .            .           .
barb    .           .           .            .            .            .            .           .
balc    0.01617750  0.01960534  0.022248935  0.024250943  0.026034267  0.027605831  0.02898188  0.03018789
elev    .           .           .            .            .            .            .           .
```

```
bath    0.085199550  0.0838115043  0.082680736  0.081528628  0.080398280  0.079272601  0.078032303  0.076890291
ensuit  0.166680433  0.1683688326  0.169734994  0.170996122  0.172165770  0.173304335  0.174640609  0.175825659
garag   0.180041844  0.1812585355  0.182722640  0.184068392  0.185309687  0.186496073  0.187655313  0.188749907
plaz    .            .             .            .            .            0.001354329  0.004030818  0.006482205
park   -0.055565035 -0.0565156797 -0.057187165 -0.057774557 -0.058291199 -0.059054286 -0.059984208 -0.060872222
trans   0.013537230  0.0157378812  0.017617838  0.019381877  0.021040782  0.022609225  0.024139608  0.025571020
kidca   0.005320989  0.0063061550  0.007231698  0.008073819  0.008845412  0.009611625  0.010495832  0.011256081
school  .            .             .            .            .            .            .            .
health  .            .             .            .            .            .            .            .
bike   -0.015129523 -0.0169907358 -0.018405091 -0.019699220 -0.020884566 -0.022214518 -0.023779544 -0.025180713
barb    .           -0.0009872164 -0.005281610 -0.009280631 -0.013013639 -0.016286637 -0.019130596 -0.021757180
balc    0.037228557  0.0383785163  0.040087885  0.041681539  0.043166537  0.044508076  0.045819284  0.047013967
elev   -0.012373887 -0.0181428622 -0.023340694 -0.028225190 -0.032813899 -0.037046132 -0.040896011 -0.044478504
fitg    0.090793036  0.0921112018  0.093275213  0.094313436  0.095239701  0.096020236  0.096746684  0.097415987
party   0.026017745  0.0288202637  0.031712681  0.034451752  0.037052440  0.039391265  0.041441026  0.043344947
categ   0.034892678  0.0387663576  0.042412656  0.045835375  0.049030475  0.052047172  0.054809446  0.057376585

age    -0.187116035 -0.18930028 -0.19127537 -0.192861425 -0.194380844 -0.19577849 -0.197064417 -0.198246428
parea   0.137896544  0.13848367  0.13935726  0.139538289  0.140037631  0.14050514  0.140942297  0.141351943
tarea   0.240122535  0.24076487  0.24130877  0.241910495  0.242395925  0.24283208  0.243219960  0.243565225
bath    0.075721825  0.07460580  0.07351596  0.072684901  0.071735743  0.07084584  0.070004334  0.069209639
ensuit  0.176923807  0.17794520  0.17891762  0.180108258  0.181190495  0.18219072  0.183124145  0.183993789
garag   0.189717260  0.19060946  0.19134377  0.192049756  0.192655927  0.19321625  0.193730869  0.194203366
plaz    0.008734664  0.01081382  0.01268674  0.014464379  0.016086532  0.01757732  0.018948370  0.020208278
park   -0.061668120 -0.06239667 -0.06325205 -0.064496546 -0.065591479 -0.06660555 -0.067544140 -0.068411874
trans   0.026859512  0.02805502  0.02898712  0.029616795  0.030189413  0.03070971  0.031182111  0.031610762
kidca   0.011973025  0.01263233  0.01345910  0.014567491  0.015578238  0.01651134  0.017374672  0.018172488
school  .            .           .           .            .            .           .            .
health  .            .           0.00088009  0.002812835  0.004552489  0.00615728  0.007635923  0.008997058
bike   -0.026481598 -0.02767867 -0.02863423 -0.029305467 -0.029938754 -0.03051350 -0.031037217 -0.031514381
barb   -0.024187723 -0.02643207 -0.02852914 -0.030502324 -0.032312638 -0.03397742 -0.035507644 -0.036913022
balc    0.048099371  0.04910191  0.05002566  0.050992737  0.051856139  0.05265161  0.053384378  0.054058762
elev   -0.047835562 -0.05095542 -0.05387882 -0.056514102 -0.058979960 -0.06125669 -0.063357273 -0.065293266
fitg    0.097941254  0.09840496  0.09880648  0.099385014  0.099829692  0.10022865  0.100585294  0.100904138
party   0.045179387  0.04689775  0.04846556  0.049689719  0.050896029  0.05201353  0.053050074  0.054010264
categ   0.059791918  0.06202857  0.06413369  0.065910528  0.067605222  0.06916508  0.070599053  0.071916466

age    -0.19933201 ......
parea   0.14173503 ......
tarea   0.24387260 ......
bath    0.06846135 ......
ensuit  0.18480284 ......
garag   0.19463711 ......
plaz    0.02136524 ......
park   -0.06921329 ......
trans   0.03199968 ......
kidca   0.01890897 ......
school  .          ......
health  0.01024899 ......
bike   -0.03194908 ......
barb   -0.03820282 ......
balc    0.05467891 ......
elev   -0.06707586 ......
fitg    0.10118930 ......
party   0.05489859 ......
categ   0.07312594 ......

 .....suppressing 20 columns in show(); maybe adjust 'options(max.print= *, width = *)'
 ..............................
```

Treino

```
> predictions_train <- predict(elastic_reg, x)
> eval_results(y_train, predictions_train, train)
       RMSE   Rsquare
1 0.3981597 0.841101
```

Teste

```
> predictions_test <- predict(elastic_reg, x_test)
> eval_results(y_test, predictions_test, test)
       RMSE    Rsquare
1 0.6545521 0.7042617
```

Predição

```
>
> price_pred_elastic=(predict_our_elastic*
+                         pre_proc_val[["std"]][["price"]])+
+     pre_proc_val[["mean"]][["price"]]
> price_pred_elastic
[1] 1020881
```

```
> n <- nrow(train)
> m <- price_pred_elastic
> s <- pre_proc_val[["std"]][["price"]]
> dam <- s/sqrt(n)
> CIlwr_elastic <- m + (qnorm(0.025))*dam
> CIupr_elastic <- m - (qnorm(0.025))*dam
>
> CIlwr_elastic
[1] 974239.1
> CIupr_elastic
[1] 1067522
```

Dentre os modelos testados o que obteve melhor resultado foi o lasso. Todavia em todos modelos o valor obtido assim como o intervalo de confiança tiveram grande divergência com o valor esperado.