# YuchenPengHW5

Question 11.1 Using the crime data set uscrime.txt from Questions 8.2, 9.1, and 10.1, build a regression model using: 1. Stepwise regression 2. Lasso 3. Elastic net

Hide

Data\_crime = read.csv("uscrime.txt",sep = "")  
str(Data\_crime)

'data.frame': 47 obs. of 16 variables:  
 $ M : num 15.1 14.3 14.2 13.6 14.1 12.1 12.7 13.1 15.7 14 ...  
 $ So : int 1 0 1 0 0 0 1 1 1 0 ...  
 $ Ed : num 9.1 11.3 8.9 12.1 12.1 11 11.1 10.9 9 11.8 ...  
 $ Po1 : num 5.8 10.3 4.5 14.9 10.9 11.8 8.2 11.5 6.5 7.1 ...  
 $ Po2 : num 5.6 9.5 4.4 14.1 10.1 11.5 7.9 10.9 6.2 6.8 ...  
 $ LF : num 0.51 0.583 0.533 0.577 0.591 0.547 0.519 0.542 0.553 0.632 ...  
 $ M.F : num 95 101.2 96.9 99.4 98.5 ...  
 $ Pop : int 33 13 18 157 18 25 4 50 39 7 ...  
 $ NW : num 30.1 10.2 21.9 8 3 4.4 13.9 17.9 28.6 1.5 ...  
 $ U1 : num 0.108 0.096 0.094 0.102 0.091 0.084 0.097 0.079 0.081 0.1 ...  
 $ U2 : num 4.1 3.6 3.3 3.9 2 2.9 3.8 3.5 2.8 2.4 ...  
 $ Wealth: int 3940 5570 3180 6730 5780 6890 6200 4720 4210 5260 ...  
 $ Ineq : num 26.1 19.4 25 16.7 17.4 12.6 16.8 20.6 23.9 17.4 ...  
 $ Prob : num 0.0846 0.0296 0.0834 0.0158 0.0414 ...  
 $ Time : num 26.2 25.3 24.3 29.9 21.3 ...  
 $ Crime : int 791 1635 578 1969 1234 682 963 1555 856 705 ...

Hide

print(summary(Data\_crime))

M So Ed Po1 Po2 LF   
 Min. :11.9 Min. :0.00 Min. : 8.70 Min. : 4.50 Min. : 4.10 Min. :0.480   
 1st Qu.:13.0 1st Qu.:0.00 1st Qu.: 9.75 1st Qu.: 6.25 1st Qu.: 5.85 1st Qu.:0.530   
 Median :13.6 Median :0.00 Median :10.80 Median : 7.80 Median : 7.30 Median :0.560   
 Mean :13.9 Mean :0.34 Mean :10.56 Mean : 8.50 Mean : 8.02 Mean :0.561   
 3rd Qu.:14.6 3rd Qu.:1.00 3rd Qu.:11.45 3rd Qu.:10.45 3rd Qu.: 9.70 3rd Qu.:0.593   
 Max. :17.7 Max. :1.00 Max. :12.20 Max. :16.60 Max. :15.70 Max. :0.641   
 M.F Pop NW U1 U2 Wealth   
 Min. : 93.4 Min. : 3.0 Min. : 0.2 Min. :0.0700 Min. :2.00 Min. :2880   
 1st Qu.: 96.4 1st Qu.: 10.0 1st Qu.: 2.4 1st Qu.:0.0805 1st Qu.:2.75 1st Qu.:4595   
 Median : 97.7 Median : 25.0 Median : 7.6 Median :0.0920 Median :3.40 Median :5370   
 Mean : 98.3 Mean : 36.6 Mean :10.1 Mean :0.0955 Mean :3.40 Mean :5254   
 3rd Qu.: 99.2 3rd Qu.: 41.5 3rd Qu.:13.2 3rd Qu.:0.1040 3rd Qu.:3.85 3rd Qu.:5915   
 Max. :107.1 Max. :168.0 Max. :42.3 Max. :0.1420 Max. :5.80 Max. :6890   
 Ineq Prob Time Crime   
 Min. :12.6 Min. :0.0069 Min. :12.2 Min. : 342   
 1st Qu.:16.6 1st Qu.:0.0327 1st Qu.:21.6 1st Qu.: 658   
 Median :17.6 Median :0.0421 Median :25.8 Median : 831   
 Mean :19.4 Mean :0.0471 Mean :26.6 Mean : 905   
 3rd Qu.:22.8 3rd Qu.:0.0544 3rd Qu.:30.5 3rd Qu.:1058   
 Max. :27.6 Max. :0.1198 Max. :44.0 Max. :1993

Hide

cor(Data\_crime)[16,]

M So Ed Po1 Po2 LF M.F Pop NW U1 U2 Wealth Ineq   
-0.0895 -0.0906 0.3228 0.6876 0.6667 0.1889 0.2139 0.3375 0.0326 -0.0505 0.1773 0.4413 -0.1790   
 Prob Time Crime   
-0.4274 0.1499 1.0000

Hide

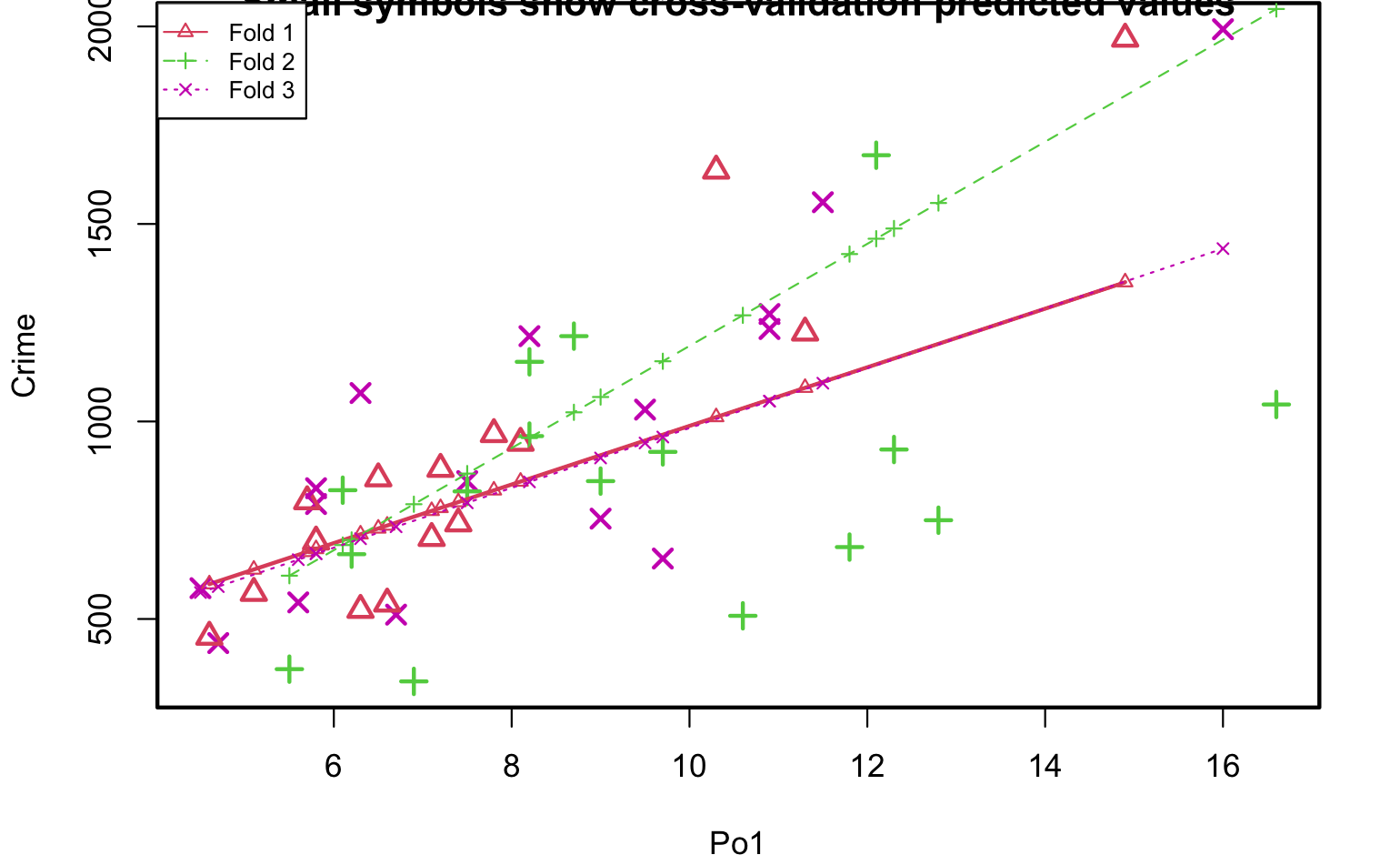
#Split data into test and train  
set.seed(1)  
train = sample(1:nrow(Data\_crime), 0.75\*nrow(Data\_crime))  
x <- model.matrix(Crime~., Data\_crime)[,-16]  
y <- Data\_crime$Crime  
x.train <- x[train, ]  
x.test <- x[-train, ]  
y.train <- y[train]  
y.test <- y[-train]  
  
train\_data <- Data\_crime[train,]  
test\_data <- Data\_crime[-train,]  
  
## 1. Build a regression model using Stepwise regression   
  
cor(Data\_crime)[16,]

M So Ed Po1 Po2 LF M.F Pop NW U1 U2 Wealth Ineq   
-0.0895 -0.0906 0.3228 0.6876 0.6667 0.1889 0.2139 0.3375 0.0326 -0.0505 0.1773 0.4413 -0.1790   
 Prob Time Crime   
-0.4274 0.1499 1.0000

Hide

# M So Ed Po1 Po2 LF M.F Pop   
# -0.08947240 -0.09063696 0.32283487 0.68760446 0.66671414 0.18886635 0.21391426 0.33747406   
# NW U1 U2 Wealth Ineq Prob Time Crime   
# 0.03259884 -0.05047792 0.17732065 0.44131995 -0.17902373 -0.42742219 0.14986606 1.00000000   
  
# Start with the variable that has the highest correlation with the outcome (Crime)  
  
library(DAAG)  
# Start with the variable that has the highest correlation with the outcome (Crime)  
cv.lm(Data\_crime, form.lm = formula(Crime ~ Po1), m=3, dots = FALSE, seed=29, plotit=TRUE, printit=TRUE)

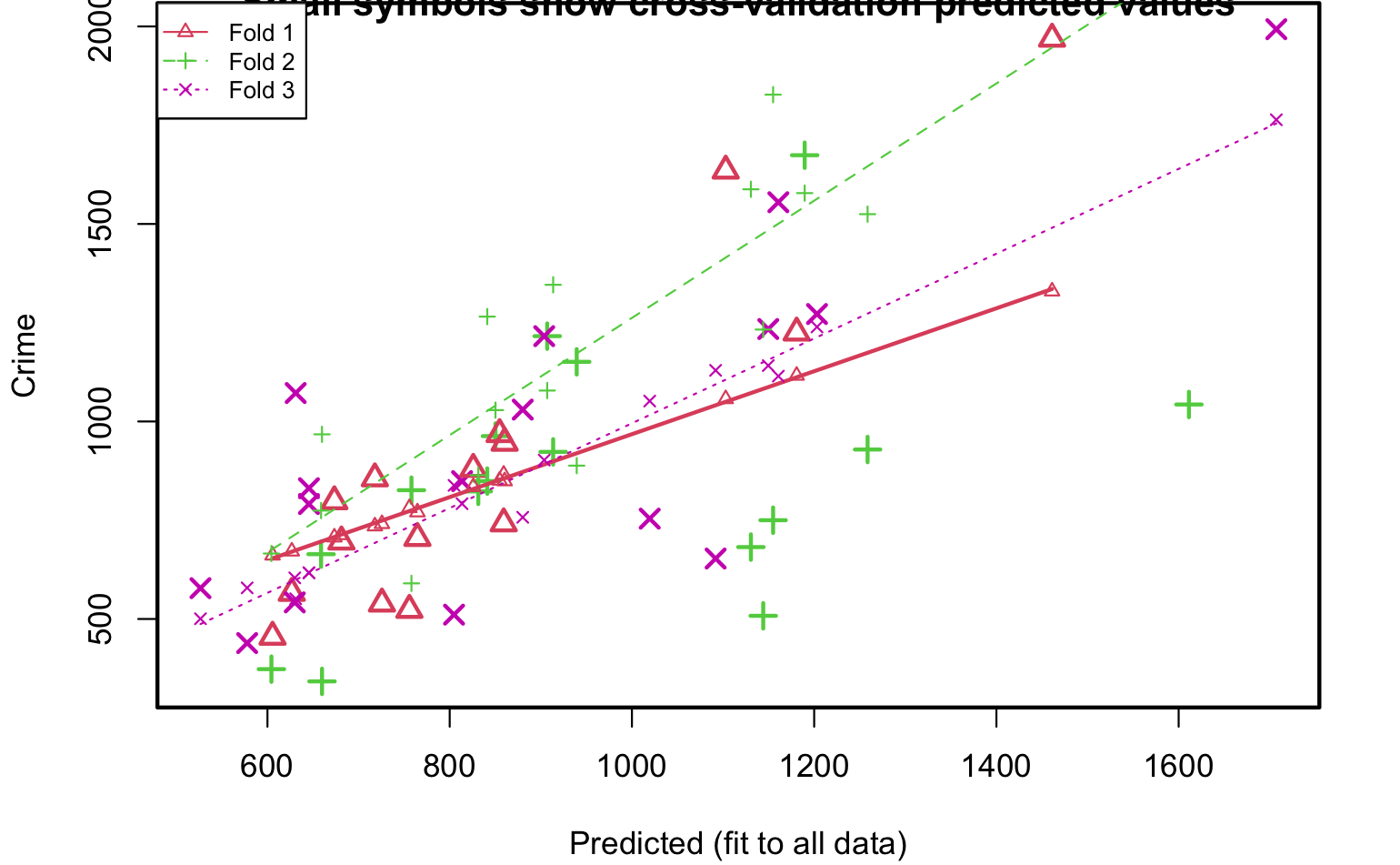
Analysis of Variance Table  
  
Response: Crime  
 Df Sum Sq Mean Sq F value Pr(>F)   
Po1 1 3253302 3253302 40.4 9.3e-08 \*\*\*  
Residuals 45 3627626 80614   
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
  
fold 1   
Observations in test set: 15   
 2 4 9 10 15 16 17 20 21 24 25 30 38 41  
Po1 10.3 14.9 6.5 7.1 5.7 8.1 6.6 11.3 7.4 7.8 6.3 5.8 5.1 7.2  
cvpred 1011.3 1352.7 729.2 773.8 669.9 848.0 736.7 1085.5 796.0 825.7 714.4 677.3 625.3 781.2  
Crime 1635.0 1969.0 856.0 705.0 798.0 946.0 539.0 1225.0 742.0 968.0 523.0 696.0 566.0 880.0  
CV residual 623.7 616.3 126.8 -68.8 128.1 98.0 -197.7 139.5 -54.0 142.3 -191.4 18.7 -59.3 98.8  
 45  
Po1 4.6  
cvpred 588.2  
Crime 455.0  
CV residual -133.2  
  
Sum of squares = 965367 Mean square = 64358 n = 15   
  
fold 2   
Observations in test set: 16   
 6 7 11 14 18 19 23 27 29 31 34 39 40  
Po1 11.8 8.20 12.1 6.2 12.3 12.8 8.7 6.9 16.6 5.5 9.7 6.1 8.2  
cvpred 1423.8 958.54 1462.6 700.1 1488.4 1553.1 1023.2 790.5 2044.2 609.6 1152.4 687.1 958.5  
Crime 682.0 963.00 1674.0 664.0 929.0 750.0 1216.0 342.0 1043.0 373.0 923.0 826.0 1151.0  
CV residual -741.8 4.46 211.4 -36.1 -559.4 -803.1 192.8 -448.5 -1001.2 -236.6 -229.4 138.9 192.5  
 43 46 47  
Po1 7.5 10.6 9  
cvpred 868.1 1268.7 1062  
Crime 823.0 508.0 849  
CV residual -45.1 -760.7 -213  
  
Sum of squares = 3585941 Mean square = 224121 n = 16   
  
fold 3   
Observations in test set: 16   
 1 3 5 8 12 13 22 26 28 32 33 35 36 37  
Po1 5.8 4.5 10.9 11.5 7.5 6.7 4.7 16 8.2 9 6.3 9.7 10.9 5.8  
cvpred 665.3 566.9 1051.4 1096.9 794.0 733.4 582.0 1438 847.0 908 703.2 960.6 1051.4 665.3  
Crime 791.0 578.0 1234.0 1555.0 849.0 511.0 439.0 1993 1216.0 754 1072.0 653.0 1272.0 831.0  
CV residual 125.7 11.1 182.6 458.1 55.0 -222.4 -143.0 555 369.0 -154 368.8 -307.6 220.6 165.7  
 42 44  
Po1 5.6 9.5  
cvpred 650.2 945.4  
Crime 542.0 1030.0  
CV residual -108.2 84.6  
  
Sum of squares = 1125941 Mean square = 70371 n = 16   
  
Overall (Sum over all 16 folds)   
 ms   
120793



Hide

# Overall (Sum over all 16 folds)   
# ms   
# 120793   
  
# Add the variable that has the 2nd highest correlation with the outcome (Crime)  
cv.lm(Data\_crime, form.lm = formula(Crime ~ Po1 + Po2), m=3, dots = FALSE, seed=29, plotit=TRUE, printit=TRUE)

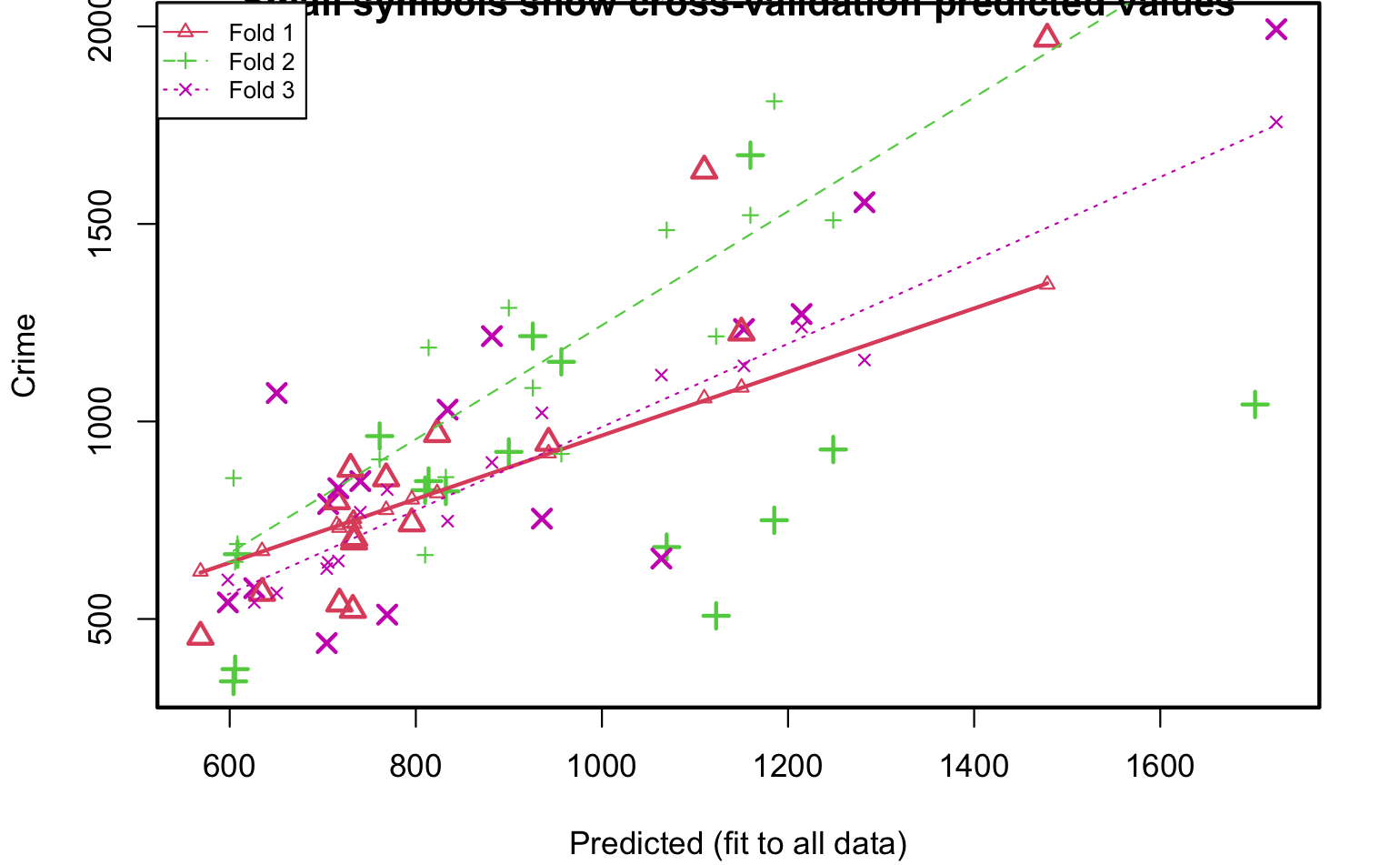
Analysis of Variance Table  
  
Response: Crime  
 Df Sum Sq Mean Sq F value Pr(>F)   
Po1 1 3253302 3253302 41.12 8.4e-08 \*\*\*  
Po2 1 146167 146167 1.85 0.18   
Residuals 44 3481459 79124   
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
 As there is >1 explanatory variable, cross-validation  
 predicted values for a fold are not a linear function  
 of corresponding overall predicted values. Lines that  
 are shown for the different folds are approximate  
fold 1   
Observations in test set: 15   
 2 4 9 10 15 16 17 20 21 24 25 30 38 41 45  
Predicted 1103 1461 718 764.6 673.4 860.3 726 1181 859 855 756 681.2 627 825.9 606  
cvpred 1057 1330 735 770.4 706.8 849.1 741 1116 866 851 781 712.8 671 834.5 661  
Crime 1635 1969 856 705.0 798.0 946.0 539 1225 742 968 523 696.0 566 880.0 455  
CV residual 578 639 121 -65.4 91.2 96.9 -202 109 -124 117 -258 -16.8 -105 45.5 -206  
  
Sum of squares = 983784 Mean square = 65586 n = 15   
  
fold 2   
Observations in test set: 16   
 6 7 11 14 18 19 23 27 29 31 34 39 40 43 46 47  
Predicted 1131 850.2 1189.6 659 1259 1155 907 660 1611 604 914 758 939 831.4 1144 841  
cvpred 1588 1028.7 1578.1 774 1525 1827 1078 967 2164 666 1346 590 888 863.8 1233 1265  
Crime 682 963.0 1674.0 664 929 750 1216 342 1043 373 923 826 1151 823.0 508 849  
CV residual -906 -65.7 95.9 -110 -596 -1077 138 -625 -1121 -293 -423 236 263 -40.8 -725 -416  
  
Sum of squares = 5118968 Mean square = 319935 n = 16   
  
fold 3   
Observations in test set: 16   
 1 3 5 8 12 13 22 26 28 32 33 35 36 37 42 44  
Predicted 646 526.5 1149.6 1161 813.6 805 578 1707 904 1020 631 1092 1203.1 646 630.0 880  
cvpred 617 500.6 1141.4 1115 791.8 838 579 1764 902 1052 551 1129 1239.4 617 604.1 757  
Crime 791 578.0 1234.0 1555 849.0 511 439 1993 1216 754 1072 653 1272.0 831 542.0 1030  
CV residual 174 77.4 92.6 440 57.2 -327 -140 229 314 -298 521 -476 32.6 214 -62.1 273  
  
Sum of squares = 1231300 Mean square = 76956 n = 16   
  
Overall (Sum over all 16 folds)   
 ms   
156044



Hide

# Overall (Sum over all 16 folds)   
# ms   
# 156044   
  
# Add the variable that has the 3rd highest correlation with the outcome (Crime)  
cv.lm(Data\_crime, form.lm = formula(Crime ~ Po1 + Po2 + Wealth), m=3, dots = FALSE, seed=29, plotit=TRUE, printit=TRUE)

Analysis of Variance Table  
  
Response: Crime  
 Df Sum Sq Mean Sq F value Pr(>F)   
Po1 1 3253302 3253302 41.78 7.7e-08 \*\*\*  
Po2 1 146167 146167 1.88 0.18   
Wealth 1 132892 132892 1.71 0.20   
Residuals 43 3348567 77874   
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
 As there is >1 explanatory variable, cross-validation  
 predicted values for a fold are not a linear function  
 of corresponding overall predicted values. Lines that  
 are shown for the different folds are approximate  
fold 1   
Observations in test set: 15   
 2 4 9 10 15 16 17 20 21 24 25 30 38 41 45  
Predicted 1110 1479 768.1 733.8 715.0 942.7 718 1150 795.6 823 732 733.6 635 730 569  
cvpred 1059 1347 776.3 741.1 738.3 919.7 731 1086 802.8 819 753 753.9 672 744 620  
Crime 1635 1969 856.0 705.0 798.0 946.0 539 1225 742.0 968 523 696.0 566 880 455  
CV residual 576 622 79.7 -36.1 59.7 26.3 -192 139 -60.8 149 -230 -57.9 -106 136 -165  
  
Sum of squares = 926515 Mean square = 61768 n = 15   
  
fold 2   
Observations in test set: 16   
 6 7 11 14 18 19 23 27 29 31 34 39 40 43 46 47  
Predicted 1070 761 1160 608.4 1249 1185 926 604 1702 606 900 810 956 832 1123 814  
cvpred 1484 904 1522 689.8 1509 1810 1085 857 2259 645 1288 662 918 859 1215 1187  
Crime 682 963 1674 664.0 929 750 1216 342 1043 373 923 826 1151 823 508 849  
CV residual -802 59 152 -25.8 -580 -1060 131 -515 -1216 -272 -365 164 233 -36 -707 -338  
  
Sum of squares = 4797875 Mean square = 3e+05 n = 16   
  
fold 3   
Observations in test set: 16   
 1 3 5 8 12 13 22 26 28 32 33 35 36 37 42 44  
Predicted 706 626.4 1152.6 1282 740.4 769 704 1725 882 936 651 1064 1214.6 717 598.0 834  
cvpred 643 541.9 1140.7 1156 770.2 828 627 1758 896 1022 566 1117 1239.2 647 599.2 748  
Crime 791 578.0 1234.0 1555 849.0 511 439 1993 1216 754 1072 653 1272.0 831 542.0 1030  
CV residual 148 36.1 93.3 399 78.8 -317 -188 235 320 -268 506 -464 32.8 184 -57.2 282  
  
Sum of squares = 1152320 Mean square = 72020 n = 16   
  
Overall (Sum over all 16 folds)   
 ms   
146313



Hide

# Overall (Sum over all 16 folds)   
# ms   
# 146313   
  
# To be more efficient, I decide to use stepAIC() that will automatically do the stepwise regression   
# stepAIC() choose the best model by AIC. It has an option named direction, which can take the following # # # values:   
# i) “both” (for stepwise regression, both forward and backward selection)  
# ii)“backward” (for backward selection)   
# iii)“forward” (for forward selection).   
# stepAIC() keeps on minimizing the stepAIC value to come up with the final set of features  
# It is used to simplify the model without impacting much on the performance  
  
library(MASS)  
# Fit the full model   
full.model <- lm(Crime ~., data = train\_data)  
step.model <- stepAIC(full.model, direction = 'both', trace = 'True')

Start: AIC=371  
Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +   
 U2 + Wealth + Ineq + Prob + Time  
  
 Df Sum of Sq RSS AIC  
- So 1 22 562167 369  
- Pop 1 342 562487 369  
- LF 1 613 562759 369  
- Po1 1 898 563044 369  
- Wealth 1 2164 564310 369  
- M.F 1 5320 567465 369  
- Time 1 8384 570529 369  
- NW 1 27587 589733 371  
<none> 562145 371  
- Po2 1 37849 599994 371  
- U1 1 71219 633364 373  
- U2 1 100349 662494 375  
- Prob 1 117369 679514 376  
- Ed 1 173027 735172 378  
- M 1 175756 737902 378  
- Ineq 1 301723 863868 384  
  
Step: AIC=369  
Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +   
 Wealth + Ineq + Prob + Time  
  
 Df Sum of Sq RSS AIC  
- Pop 1 341 562508 367  
- Po1 1 883 563050 367  
- LF 1 986 563153 367  
- Wealth 1 2174 564341 367  
- M.F 1 5487 567654 367  
- Time 1 8844 571011 367  
- NW 1 31800 593967 369  
<none> 562167 369  
- Po2 1 38042 600209 369  
+ So 1 22 562145 371  
- U1 1 81259 643426 372  
- U2 1 104019 666186 373  
- Prob 1 117904 680071 374  
- Ed 1 173856 736023 376  
- M 1 178112 740279 377  
- Ineq 1 344499 906666 384  
  
Step: AIC=367  
Crime ~ M + Ed + Po1 + Po2 + LF + M.F + NW + U1 + U2 + Wealth +   
 Ineq + Prob + Time  
  
 Df Sum of Sq RSS AIC  
- Po1 1 813 563320 365  
- LF 1 1144 563652 365  
- Wealth 1 2316 564824 365  
- M.F 1 7577 570084 365  
- Time 1 10139 572646 366  
- NW 1 31598 594106 367  
<none> 562508 367  
- Po2 1 37769 600276 367  
+ Pop 1 341 562167 369  
+ So 1 21 562487 369  
- U1 1 86559 649066 370  
- U2 1 104631 667138 371  
- Prob 1 117649 680156 372  
- Ed 1 177123 739631 375  
- M 1 177827 740335 375  
- Ineq 1 387422 949929 383  
  
Step: AIC=365  
Crime ~ M + Ed + Po2 + LF + M.F + NW + U1 + U2 + Wealth + Ineq +   
 Prob + Time  
  
 Df Sum of Sq RSS AIC  
- LF 1 1462 564783 363  
- Wealth 1 1958 565278 363  
- M.F 1 6765 570085 363  
- Time 1 15578 578899 364  
- NW 1 31526 594846 365  
<none> 563320 365  
+ Po1 1 813 562508 367  
+ Pop 1 271 563050 367  
+ So 1 6 563314 367  
- U1 1 88906 652226 368  
- U2 1 120049 683369 370  
- Prob 1 137223 700544 371  
- Ed 1 178156 741476 373  
- M 1 178885 742205 373  
- Ineq 1 386793 950114 381  
- Po2 1 870963 1434284 396  
  
Step: AIC=363  
Crime ~ M + Ed + Po2 + M.F + NW + U1 + U2 + Wealth + Ineq + Prob +   
 Time  
  
 Df Sum of Sq RSS AIC  
- Wealth 1 1849 566632 361  
- M.F 1 5319 570102 361  
- Time 1 16678 581460 362  
- NW 1 32864 597646 363  
<none> 564783 363  
+ LF 1 1462 563320 365  
+ Po1 1 1131 563652 365  
+ So 1 456 564326 365  
+ Pop 1 423 564360 365  
- U1 1 88177 652959 366  
- U2 1 120700 685483 368  
- Prob 1 136144 700927 369  
- M 1 187876 752659 371  
- Ed 1 191458 756241 371  
- Ineq 1 385671 950454 379  
- Po2 1 883734 1448517 394  
  
Step: AIC=361  
Crime ~ M + Ed + Po2 + M.F + NW + U1 + U2 + Ineq + Prob + Time  
  
 Df Sum of Sq RSS AIC  
- M.F 1 5175 571807 360  
- Time 1 17455 584087 360  
- NW 1 31172 597804 361  
<none> 566632 361  
+ Wealth 1 1849 564783 363  
+ LF 1 1354 565278 363  
+ Po1 1 698 565934 363  
+ Pop 1 576 566056 363  
+ So 1 154 566478 363  
- U1 1 86591 653223 364  
- U2 1 118877 685509 366  
- Prob 1 134384 701016 367  
- Ed 1 190935 757567 369  
- M 1 192100 758732 369  
- Ineq 1 562487 1129119 383  
- Po2 1 1138086 1704718 398  
  
Step: AIC=360  
Crime ~ M + Ed + Po2 + NW + U1 + U2 + Ineq + Prob + Time  
  
 Df Sum of Sq RSS AIC  
- Time 1 22324 594131 359  
<none> 571807 360  
- NW 1 36100 607907 360  
+ M.F 1 5175 566632 361  
+ Pop 1 2567 569240 361  
+ Wealth 1 1705 570102 361  
+ So 1 55 571751 362  
+ Po1 1 28 571778 362  
+ LF 1 11 571796 362  
- U1 1 88895 660702 363  
- U2 1 117933 689739 364  
- Prob 1 141952 713759 365  
- Ed 1 226820 798627 369  
- M 1 231408 803215 369  
- Ineq 1 659048 1230855 384  
- Po2 1 1433153 2004960 401  
  
Step: AIC=359  
Crime ~ M + Ed + Po2 + NW + U1 + U2 + Ineq + Prob  
  
 Df Sum of Sq RSS AIC  
<none> 594131 359  
- NW 1 46109 640240 359  
+ Time 1 22324 571807 360  
+ M.F 1 10044 584087 360  
+ Pop 1 8383 585748 360  
- U1 1 68507 662637 361  
+ Wealth 1 2512 591619 361  
+ Po1 1 2000 592131 361  
+ So 1 1106 593025 361  
+ LF 1 14 594116 361  
- U2 1 97605 691735 362  
- Prob 1 129377 723507 364  
- M 1 214236 808366 368  
- Ed 1 248286 842417 369  
- Ineq 1 707289 1301420 384  
- Po2 1 1571219 2165349 402

Hide

step.model

Call:  
lm(formula = Crime ~ M + Ed + Po2 + NW + U1 + U2 + Ineq + Prob,   
 data = train\_data)  
  
Coefficients:  
(Intercept) M Ed Po2 NW U1 U2 Ineq   
 -4378.53 95.15 153.51 152.07 -6.78 -5312.08 153.81 70.93   
 Prob   
 -3111.12

Hide

summary(step.model)

Call:  
lm(formula = Crime ~ M + Ed + Po2 + NW + U1 + U2 + Ineq + Prob,   
 data = train\_data)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-285.5 -96.3 11.3 89.2 283.9   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -4378.53 844.39 -5.19 2.1e-05 \*\*\*  
M 95.15 31.08 3.06 0.0051 \*\*   
Ed 153.51 46.57 3.30 0.0028 \*\*   
Po2 152.07 18.34 8.29 9.0e-09 \*\*\*  
NW -6.78 4.77 -1.42 0.1673   
U1 -5312.08 3067.98 -1.73 0.0952 .   
U2 153.81 74.42 2.07 0.0489 \*   
Ineq 70.93 12.75 5.56 7.6e-06 \*\*\*  
Prob -3111.12 1307.51 -2.38 0.0250 \*   
---  
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
Residual standard error: 151 on 26 degrees of freedom  
Multiple R-squared: 0.9, Adjusted R-squared: 0.869   
F-statistic: 29.3 on 8 and 26 DF, p-value: 4.09e-11

Hide

# Start: AIC=515  
# Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +   
# U2 + Wealth + Ineq + Prob + Time  
#   
# Df Sum of Sq RSS AIC  
# - So 1 29 1354974 513  
# - LF 1 8917 1363862 513  
# - Time 1 10304 1365250 513  
# - Pop 1 14122 1369068 513  
# - NW 1 18395 1373341 513  
# - M.F 1 31967 1386913 514  
# - Wealth 1 37613 1392558 514  
# - Po2 1 37919 1392865 514  
# <none> 1354946 515  
# - U1 1 83722 1438668 515  
# - Po1 1 144306 1499252 517  
# - U2 1 181536 1536482 519  
# - M 1 193770 1548716 519  
# - Prob 1 199538 1554484 519  
# - Ed 1 402117 1757063 525  
# - Ineq 1 423031 1777977 525  
#   
# Step: AIC=513  
# Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +   
# Wealth + Ineq + Prob + Time  
#   
# Df Sum of Sq RSS AIC  
# - Time 1 10341 1365315 511  
# - LF 1 10878 1365852 511  
# - Pop 1 14127 1369101 511  
# - NW 1 21626 1376600 511  
# - M.F 1 32449 1387423 512  
# - Po2 1 37954 1392929 512  
# - Wealth 1 39223 1394197 512  
# <none> 1354974 513  
# - U1 1 96420 1451395 514  
# + So 1 29 1354946 515  
# - Po1 1 144302 1499277 515  
# - U2 1 189859 1544834 517  
# - M 1 195084 1550059 517  
# - Prob 1 204463 1559437 517  
# - Ed 1 403140 1758114 523  
# - Ineq 1 488834 1843808 525  
#   
# Step: AIC=511  
# Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +   
# Wealth + Ineq + Prob  
#   
# Df Sum of Sq RSS AIC  
# - LF 1 10533 1375848 509  
# - NW 1 15482 1380797 510  
# - Pop 1 21846 1387161 510  
# - Po2 1 28932 1394247 510  
# - Wealth 1 36070 1401385 510  
# - M.F 1 41784 1407099 510  
# <none> 1365315 511  
# - U1 1 91420 1456735 512  
# + Time 1 10341 1354974 513  
# + So 1 65 1365250 513  
# - Po1 1 134137 1499452 513  
# - U2 1 184143 1549458 515  
# - M 1 186110 1551425 515  
# - Prob 1 237493 1602808 517  
# - Ed 1 409448 1774763 521  
# - Ineq 1 502909 1868224 524  
#   
# Step: AIC=509  
# Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 + U2 + Wealth +   
# Ineq + Prob  
#   
# Df Sum of Sq RSS AIC  
# - NW 1 11675 1387523 508  
# - Po2 1 21418 1397266 508  
# - Pop 1 27803 1403651 508  
# - M.F 1 31252 1407100 508  
# - Wealth 1 35035 1410883 509  
# <none> 1375848 509  
# - U1 1 80954 1456802 510  
# + LF 1 10533 1365315 511  
# + Time 1 9996 1365852 511  
# + So 1 3046 1372802 511  
# - Po1 1 123896 1499744 511  
# - U2 1 190746 1566594 513  
# - M 1 217716 1593564 514  
# - Prob 1 226971 1602819 515  
# - Ed 1 413254 1789103 520  
# - Ineq 1 500944 1876792 522  
#   
# Step: AIC=508  
# Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + U1 + U2 + Wealth + Ineq +   
# Prob  
#   
# Df Sum of Sq RSS AIC  
# - Po2 1 16706 1404229 506  
# - Pop 1 25793 1413315 507  
# - M.F 1 26785 1414308 507  
# - Wealth 1 31551 1419073 507  
# <none> 1387523 508  
# - U1 1 83881 1471404 509  
# + NW 1 11675 1375848 509  
# + So 1 7207 1380316 510  
# + LF 1 6726 1380797 510  
# + Time 1 4534 1382989 510  
# - Po1 1 118348 1505871 510  
# - U2 1 201453 1588976 512  
# - Prob 1 216760 1604282 513  
# - M 1 309214 1696737 515  
# - Ed 1 402754 1790276 518  
# - Ineq 1 589736 1977259 522  
#   
# Step: AIC=506  
# Crime ~ M + Ed + Po1 + M.F + Pop + U1 + U2 + Wealth + Ineq +   
# Prob  
#   
# Df Sum of Sq RSS AIC  
# - Pop 1 22345 1426575 505  
# - Wealth 1 32142 1436371 505  
# - M.F 1 36808 1441037 506  
# <none> 1404229 506  
# - U1 1 86373 1490602 507  
# + Po2 1 16706 1387523 508  
# + NW 1 6963 1397266 508  
# + So 1 3807 1400422 508  
# + LF 1 1986 1402243 508  
# + Time 1 575 1403654 508  
# - U2 1 205814 1610043 511  
# - Prob 1 218607 1622836 511  
# - M 1 307001 1711230 514  
# - Ed 1 389502 1793731 516  
# - Ineq 1 608627 2012856 521  
# - Po1 1 1050202 2454432 531  
#   
# Step: AIC=505  
# Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Wealth + Ineq + Prob  
#   
# Df Sum of Sq RSS AIC  
# - Wealth 1 26493 1453068 504  
# <none> 1426575 505  
# - M.F 1 84491 1511065 506  
# - U1 1 99463 1526037 506  
# + Pop 1 22345 1404229 506  
# + Po2 1 13259 1413315 507  
# + NW 1 5927 1420648 507  
# + So 1 5724 1420851 507  
# + LF 1 5176 1421398 507  
# + Time 1 3913 1422661 507  
# - Prob 1 198571 1625145 509  
# - U2 1 208880 1635455 509  
# - M 1 320926 1747501 513  
# - Ed 1 386773 1813348 514  
# - Ineq 1 594779 2021354 519  
# - Po1 1 1127277 2553852 530  
#   
# Step: AIC=504  
# Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob  
#   
# Df Sum of Sq RSS AIC  
# <none> 1453068 504  
# + Wealth 1 26493 1426575 505  
# - M.F 1 103159 1556227 505  
# + Pop 1 16697 1436371 505  
# + Po2 1 14148 1438919 505  
# + So 1 9329 1443739 506  
# + LF 1 4374 1448694 506  
# + NW 1 3799 1449269 506  
# + Time 1 2293 1450775 506  
# - U1 1 127044 1580112 506  
# - Prob 1 247978 1701046 509  
# - U2 1 255443 1708511 510  
# - M 1 296790 1749858 511  
# - Ed 1 445788 1898855 515  
# - Ineq 1 738244 2191312 521  
# - Po1 1 1672038 3125105 538  
  
#The model with the best AIC   
  
# Call:  
# lm.default(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq +   
# Prob, data = Data\_crime)  
#   
# Residuals:  
# Min 1Q Median 3Q Max   
# -445 -111 3 122 483   
#   
# Coefficients:  
# Estimate Std. Error t value Pr(>|t|)   
# (Intercept) -6426.1 1194.6 -5.38 4.0e-06 \*\*\*  
# M 93.3 33.5 2.79 0.0083 \*\*   
# Ed 180.1 52.8 3.41 0.0015 \*\*   
# Po1 102.7 15.5 6.61 8.3e-08 \*\*\*  
# M.F 22.3 13.6 1.64 0.1087   
# U1 -6086.6 3339.3 -1.82 0.0762 .   
# U2 187.3 72.5 2.58 0.0137 \*   
# Ineq 61.3 14.0 4.39 8.6e-05 \*\*\*  
# Prob -3796.0 1490.6 -2.55 0.0151 \*   
# ---  
# Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
#   
# Residual standard error: 196 on 38 degrees of freedom  
# Multiple R-squared: 0.789, Adjusted R-squared: 0.744   
# F-statistic: 17.7 on 8 and 38 DF, p-value: 1.16e-10  
  
  
  
class(x.test)

[1] "matrix" "array"

Hide

# Make predictions on the test data  
predictions <- predict(step.model, as.data.frame(x.test))  
  
# Model performance metrics  
#data.frame(RMSE = RMSE(predictions, y.test), Rsquare = R2(predictions, y.test))  
#Best Model RMSE and R^2  
# RMSE Rsquare  
# 1 378 0.204  
  
## 2. Lasso  
library(glmnet)  
library(caret)  
lambda\_seq <- 10^seq(3,-1, by = -0.1)  
lambda\_seq

[1] 1000.000 794.328 630.957 501.187 398.107 316.228 251.189 199.526 158.489 125.893 100.000  
[12] 79.433 63.096 50.119 39.811 31.623 25.119 19.953 15.849 12.589 10.000 7.943  
[23] 6.310 5.012 3.981 3.162 2.512 1.995 1.585 1.259 1.000 0.794 0.631  
[34] 0.501 0.398 0.316 0.251 0.200 0.158 0.126 0.100

Hide

# [1] 1000.0000000 794.3282347 630.9573445 501.1872336 398.1071706 316.2277660 251.1886432  
# [8] 199.5262315 158.4893192 125.8925412 100.0000000 79.4328235 63.0957344 50.1187234  
# [15] 39.8107171 31.6227766 25.1188643 19.9526231 15.8489319 12.5892541 10.0000000  
# [22] 7.9432823 6.3095734 5.0118723 3.9810717 3.1622777 2.5118864 1.9952623  
# [29] 1.5848932 1.2589254 1.0000000 0.7943282 0.6309573 0.5011872 0.3981072  
# [36] 0.3162278 0.2511886 0.1995262 0.1584893 0.1258925 0.1000000  
  
  
#Split data into test and train  
set.seed(1)  
train = sample(1:nrow(Data\_crime), 0.75\*nrow(Data\_crime))  
x <- model.matrix(Crime~., Data\_crime)[,-16]  
y <- Data\_crime$Crime  
x.train <- x[train, ]  
x.test <- x[-train, ]  
y.train <- y[train]  
y.test <- y[-train]  
  
test <- Data\_crime[-train,]  
  
cv\_output <- cv.glmnet(x.train, y.train, alpha=1, lambda = lambda\_seq, nfold = 5)  
summary(cv\_output)

Length Class Mode   
lambda 41 -none- numeric   
cvm 41 -none- numeric   
cvsd 41 -none- numeric   
cvup 41 -none- numeric   
cvlo 41 -none- numeric   
nzero 41 -none- numeric   
call 6 -none- call   
name 1 -none- character  
glmnet.fit 12 elnet list   
lambda.min 1 -none- numeric   
lambda.1se 1 -none- numeric

Hide

cv\_output$cvm

[1] 199758 199758 199758 199758 199758 186731 143510 116022 101055 91565 85980 82841 80169 74339  
[15] 67999 62284 59031 56508 52196 48725 47337 47380 47833 48004 48661 49602 50265 50918  
[29] 51351 51657 51952 52877 53728 54489 55085 55704 56162 56558 56898 57193 57396

Hide

# The mean cross-validated error  
# [1] 199758 199758 199758 199758 199758 186731 143510 116022 101055 91565 85980 82841 80169 74339 67999  
# [16] 62284 59031 56508 52196 48725 47337 47380 47833 48004 48661 49602 50265 50918 51351 51657  
# [31] 51952 52877 53728 54489 55085 55704 56162 56558 56898 57193 57396  
  
  
# identify lamda with minimum cvm  
best\_lam <- cv\_output$lambda.min  
# best lambda is 10  
  
# Fit the final model on the training data  
model <- glmnet(x.train, y.train, alpha = 1, lambda = cv\_output$lambda.min)  
# Dsiplay regression coefficients  
coef(model)

16 x 1 sparse Matrix of class "dgCMatrix"  
 s0  
(Intercept) -2561.03  
(Intercept) .   
M 44.11  
So 0.02  
Ed 83.20  
Po1 90.20  
Po2 53.01  
LF .   
M.F .   
Pop .   
NW .   
U1 .   
U2 .   
Wealth .   
Ineq 51.10  
Prob -3409.48

Hide

# Make predictions on the test data  
x.test <- model.matrix(Crime ~., test <- Data\_crime[-train,])[,-1]  
predictions <- predict(model, x.test)  
  
  
# Model performance metrics  
data.frame(RMSE = RMSE(predictions, y.test),Rsquare = R2(predictions, y.test))

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **RMSE**  <dbl> | **s0**  <dbl> |
| 90619 | 0.00543 |

1 row

Hide

#Best Model RMSE and R^2  
# RMSE s0  
# 1 90619 0.00543  
  
  
# 3. Elastic Net need to tune two parameters: 1.alpha 2.lambda  
library("glmnetUtils")  
  
# Train model.  
set.seed(1)  
fit <- cva.glmnet(x.train, y.train)  
  
# Get alpha.  
get\_alpha <- function(fit) {  
 alpha <- fit$alpha  
 error <- sapply(fit$modlist, function(mod) {min(mod$cvm)})  
 alpha[which.min(error)]  
}  
  
# Get all parameters.  
get\_model\_params <- function(fit) {  
 alpha <- fit$alpha  
 lambdaMin <- sapply(fit$modlist, `[[`, "lambda.min")  
 lambdaSE <- sapply(fit$modlist, `[[`, "lambda.1se")  
 error <- sapply(fit$modlist, function(mod) {min(mod$cvm)})  
 best <- which.min(error)  
 data.frame(alpha = alpha[best], lambdaMin = lambdaMin[best],  
 lambdaSE = lambdaSE[best], eror = error[best])  
}  
  
# Refer to the code from: <https://stackoverflow.com/questions/54803990/extract-the-best-parameters-from-cva-glmnet-object>get\_alpha(fit)

[1] 0.512

Hide

# Best alpha is 0.512  
get\_model\_params(fit)

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **alpha**  <dbl> | **lambdaMin**  <dbl> | **lambdaSE**  <dbl> | **eror**  <dbl> |
| 0.512 | 11 | 85.4 | 63549 |

1 row

Hide

# Best alpha and lambda:  
# alpha lambdaMin lambdaSE eror  
# 1 0.512 9.16 25.5 42500  
  
  
# Fit the final model on the training data  
model <- glmnet(x.train, y.train, alpha = 0.512, lambda = 9.16)  
  
# Dsiplay regression coefficients  
# coef(model)  
# 16 x 1 sparse Matrix of class "dgCMatrix"  
# s0  
# (Intercept) -3537.18  
# (Intercept) .   
# M 58.02  
# So 14.36  
# Ed 109.33  
# Po1 59.72  
# Po2 79.28  
# LF .   
# M.F 5.42  
# Pop .   
# NW -1.17  
# U1 -2761.32  
# U2 72.34  
# Wealth .   
# Ineq 53.85  
# Prob -3528.65  
  
  
# Make predictions on the test data  
x.test <- model.matrix(Crime ~., test <- Data\_crime[-train,])[,-1]  
predictions <- predict(model, x.test)  
  
# Model performance metrics  
data.frame(RMSE = RMSE(predictions, y.test),Rsquare = R2(predictions, y.test))

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **RMSE**  <dbl> | **s0**  <dbl> |
| 318888 | 0.00198 |

1 row

Hide

#Best Model RMSE and R^2  
# RMSE s0  
# 1 318888 0.00198  
  
# Elastic Net on the test set is performing far worse than ridge and stepwise regression.

Question 12.1 Describe a situation or problem from your job, everyday life, current events, etc., for which a design of experiments approach would be appropriate. A design of experiment if quite useful in direct mailing marketing. One can design multiple versions of mail piece by changing (1) the offerings (1 product or 2 products), (2) the fonts, (3) the layout, (4) the messages. Then one can find out which combination of factors will lead to higher response rates from potential customers.

Question 12.2 To determine the value of 10 different yes/no features to the market value of a house (large yard, solar roof, etc.), a real estate agent plans to survey 50 potential buyers, showing a fictitious house with different combinations of features. To reduce the survey size, the agent wants to show just 16 fictitious houses.

Use R’s FrF2 function (in the FrF2 package) to find a fractional factorial design for this experiment: what set of features should each of the 16 fictitious houses have?

Note: the output of FrF2 is “1” (include) or “-1” (don’t include) for each feature.

Hide

require(FrF2)

Loading required package: FrF2  
Loading required package: DoE.base  
Loading required package: grid  
Loading required package: conf.design  
Registered S3 method overwritten by 'partitions':  
 method from  
 print.equivalence lava  
Registered S3 method overwritten by 'DoE.base':  
 method from   
 factorize.factor conf.design  
  
Attaching package: ‘DoE.base’  
  
The following objects are masked from ‘package:stats’:  
  
 aov, lm  
  
The following object is masked from ‘package:graphics’:  
  
 plot.design  
  
The following object is masked from ‘package:base’:  
  
 lengths  
  
Warning messages:  
1: In readChar(file, size, TRUE) : truncating string with embedded nuls  
2: In readChar(file, size, TRUE) : truncating string with embedded nuls  
3: In readChar(file, size, TRUE) : truncating string with embedded nuls  
4: In readChar(file, size, TRUE) : truncating string with embedded nuls  
5: In readChar(file, size, TRUE) : truncating string with embedded nuls  
6: In readChar(file, size, TRUE) : truncating string with embedded nuls

Hide

## maximum resolution minimum aberration design with 10 factors (yes/no) in 16 runs  
result <- FrF2(16, 10, default.levels = c("1", "-1"))  
  
#Feature: A, B, C, ..., K  
#Ficticious House: 1, 2, 3, ... , 16  
#Interpretation: ex.   
# A B C D E F G H J K  
# 1 Yes No No No Yes Yes No Yes No Yes  
# Show the first fictitious house that has feature A, E, F, H, K  
  
# A B C D E F G H J K  
# 1 Yes No No No Yes Yes No Yes No Yes  
# 2 No No No Yes No No No Yes Yes Yes  
# 3 Yes Yes No No No Yes Yes Yes Yes No  
# 4 Yes No Yes Yes Yes No Yes No No Yes  
# 5 No Yes Yes No Yes Yes No No No No  
# 6 No No No No No No No No No No  
# 7 No No Yes No No Yes Yes No Yes Yes  
# 8 Yes Yes No Yes No Yes Yes No No Yes  
# 9 No Yes No Yes Yes No Yes Yes No No  
# 10 No Yes No No Yes No Yes No Yes Yes  
# 11 Yes Yes Yes Yes No No No No Yes No  
# 12 Yes Yes Yes No No No No Yes No Yes  
# 13 No Yes Yes Yes Yes Yes No Yes Yes Yes  
# 14 No No Yes Yes No Yes Yes Yes No No  
# 15 Yes No Yes No Yes No Yes Yes Yes No  
# 16 Yes No No Yes Yes Yes No No Yes No

Question 13.1

For each of the following distributions, give an example of data that you would expect to follow this distribution (besides the examples already discussed in class). a. Binomial

Whether a global pandemic will happen in 2020

1. Geometric

How many times one has to practice a speech before one can present it flawlessly in front of audiences

1. Poisson

No of diners in a Chinese restaurant every day given the average weekly number of diners is known (which is λ)

1. Exponential

The time between each diner arrival in a Chinese restaurant in a certain time period is exponentially distributed.

1. Weibull

This can be used for warranty analysis. For example, one can analyze how long it will take for a printer to break down. This information is very important for the company to determine the length of their warranty or how much they want to charge for their extended warranty.