ISLR_Chapter5_LAB

SuhaasAdiraju

2024-09-18

Chapter 5 lab, Cross-val and Bootstrapping

The Validation Set Approach

```
library(ISLR)
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

```
Auto['mpg']
```

```
##       mpg
## 1    18.0
## 2    15.0
## 3    18.0
## 4    16.0
## 5    17.0
## 6    15.0
## 7    14.0
## 8    14.0
## 9    14.0
## 10   15.0
## 11   15.0
## 12   14.0
## 13   15.0
## 14   14.0
## 15   24.0
## 16   22.0
## 17   18.0
## 18   21.0
## 19   27.0
## 20   26.0
## 21   25.0
## 22   24.0
## 23   25.0
## 24   26.0
## 25   21.0
## 26   10.0
```

```
## 27   10.0
## 28   11.0
## 29    9.0
## 30   27.0
## 31   28.0
## 32   25.0
## 34   19.0
## 35   16.0
## 36   17.0
## 37   19.0
## 38   18.0
## 39   14.0
## 40   14.0
## 41   14.0
## 42   14.0
## 43   12.0
## 44   13.0
## 45   13.0
## 46   18.0
## 47   22.0
## 48   19.0
## 49   18.0
## 50   23.0
## 51   28.0
## 52   30.0
## 53   30.0
## 54   31.0
## 55   35.0
## 56   27.0
## 57   26.0
## 58   24.0
## 59   25.0
## 60   23.0
## 61   20.0
## 62   21.0
## 63   13.0
## 64   14.0
## 65   15.0
## 66   14.0
## 67   17.0
## 68   11.0
## 69   13.0
## 70   12.0
## 71   13.0
## 72   19.0
## 73   15.0
## 74   13.0
## 75   13.0
## 76   14.0
## 77   18.0
## 78   22.0
## 79   21.0
## 80   26.0
## 81   22.0
```

```
## 82   28.0
## 83   23.0
## 84   28.0
## 85   27.0
## 86   13.0
## 87   14.0
## 88   13.0
## 89   14.0
## 90   15.0
## 91   12.0
## 92   13.0
## 93   13.0
## 94   14.0
## 95   13.0
## 96   12.0
## 97   13.0
## 98   18.0
## 99   16.0
## 100 18.0
## 101 18.0
## 102 23.0
## 103 26.0
## 104 11.0
## 105 12.0
## 106 13.0
## 107 12.0
## 108 18.0
## 109 20.0
## 110 21.0
## 111 22.0
## 112 18.0
## 113 19.0
## 114 21.0
## 115 26.0
## 116 15.0
## 117 16.0
## 118 29.0
## 119 24.0
## 120 20.0
## 121 19.0
## 122 15.0
## 123 24.0
## 124 20.0
## 125 11.0
## 126 20.0
## 128 19.0
## 129 15.0
## 130 31.0
## 131 26.0
## 132 32.0
## 133 25.0
## 134 16.0
## 135 16.0
## 136 18.0
```

```
## 137 16.0
## 138 13.0
## 139 14.0
## 140 14.0
## 141 14.0
## 142 29.0
## 143 26.0
## 144 26.0
## 145 31.0
## 146 32.0
## 147 28.0
## 148 24.0
## 149 26.0
## 150 24.0
## 151 26.0
## 152 31.0
## 153 19.0
## 154 18.0
## 155 15.0
## 156 15.0
## 157 16.0
## 158 15.0
## 159 16.0
## 160 14.0
## 161 17.0
## 162 16.0
## 163 15.0
## 164 18.0
## 165 21.0
## 166 20.0
## 167 13.0
## 168 29.0
## 169 23.0
## 170 20.0
## 171 23.0
## 172 24.0
## 173 25.0
## 174 24.0
## 175 18.0
## 176 29.0
## 177 19.0
## 178 23.0
## 179 23.0
## 180 22.0
## 181 25.0
## 182 33.0
## 183 28.0
## 184 25.0
## 185 25.0
## 186 26.0
## 187 27.0
## 188 17.5
## 189 16.0
## 190 15.5
```

```
## 191 14.5
## 192 22.0
## 193 22.0
## 194 24.0
## 195 22.5
## 196 29.0
## 197 24.5
## 198 29.0
## 199 33.0
## 200 20.0
## 201 18.0
## 202 18.5
## 203 17.5
## 204 29.5
## 205 32.0
## 206 28.0
## 207 26.5
## 208 20.0
## 209 13.0
## 210 19.0
## 211 19.0
## 212 16.5
## 213 16.5
## 214 13.0
## 215 13.0
## 216 13.0
## 217 31.5
## 218 30.0
## 219 36.0
## 220 25.5
## 221 33.5
## 222 17.5
## 223 17.0
## 224 15.5
## 225 15.0
## 226 17.5
## 227 20.5
## 228 19.0
## 229 18.5
## 230 16.0
## 231 15.5
## 232 15.5
## 233 16.0
## 234 29.0
## 235 24.5
## 236 26.0
## 237 25.5
## 238 30.5
## 239 33.5
## 240 30.0
## 241 30.5
## 242 22.0
## 243 21.5
## 244 21.5
```

```
## 245 43.1
## 246 36.1
## 247 32.8
## 248 39.4
## 249 36.1
## 250 19.9
## 251 19.4
## 252 20.2
## 253 19.2
## 254 20.5
## 255 20.2
## 256 25.1
## 257 20.5
## 258 19.4
## 259 20.6
## 260 20.8
## 261 18.6
## 262 18.1
## 263 19.2
## 264 17.7
## 265 18.1
## 266 17.5
## 267 30.0
## 268 27.5
## 269 27.2
## 270 30.9
## 271 21.1
## 272 23.2
## 273 23.8
## 274 23.9
## 275 20.3
## 276 17.0
## 277 21.6
## 278 16.2
## 279 31.5
## 280 29.5
## 281 21.5
## 282 19.8
## 283 22.3
## 284 20.2
## 285 20.6
## 286 17.0
## 287 17.6
## 288 16.5
## 289 18.2
## 290 16.9
## 291 15.5
## 292 19.2
## 293 18.5
## 294 31.9
## 295 34.1
## 296 35.7
## 297 27.4
## 298 25.4
```

```
## 299 23.0
## 300 27.2
## 301 23.9
## 302 34.2
## 303 34.5
## 304 31.8
## 305 37.3
## 306 28.4
## 307 28.8
## 308 26.8
## 309 33.5
## 310 41.5
## 311 38.1
## 312 32.1
## 313 37.2
## 314 28.0
## 315 26.4
## 316 24.3
## 317 19.1
## 318 34.3
## 319 29.8
## 320 31.3
## 321 37.0
## 322 32.2
## 323 46.6
## 324 27.9
## 325 40.8
## 326 44.3
## 327 43.4
## 328 36.4
## 329 30.0
## 330 44.6
## 332 33.8
## 333 29.8
## 334 32.7
## 335 23.7
## 336 35.0
## 338 32.4
## 339 27.2
## 340 26.6
## 341 25.8
## 342 23.5
## 343 30.0
## 344 39.1
## 345 39.0
## 346 35.1
## 347 32.3
## 348 37.0
## 349 37.7
## 350 34.1
## 351 34.7
## 352 34.4
## 353 29.9
## 354 33.0
```

```
## 356 33.7
## 357 32.4
## 358 32.9
## 359 31.6
## 360 28.1
## 361 30.7
## 362 25.4
## 363 24.2
## 364 22.4
## 365 26.6
## 366 20.2
## 367 17.6
## 368 28.0
## 369 27.0
## 370 34.0
## 371 31.0
## 372 29.0
## 373 27.0
## 374 24.0
## 375 36.0
## 376 37.0
## 377 31.0
## 378 38.0
## 379 36.0
## 380 36.0
## 381 36.0
## 382 34.0
## 383 38.0
## 384 32.0
## 385 38.0
## 386 25.0
## 387 38.0
## 388 26.0
## 389 22.0
## 390 32.0
## 391 36.0
## 392 27.0
## 393 27.0
## 394 44.0
## 395 32.0
## 396 28.0
## 397 31.0
```

```r
set.seed(1)
trainAuto = sample (392, 196)
```

```r
lm.fit = lm(mpg~horsepower, data = Auto, subset=trainAuto)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto, subset = trainAuto)
##
```

```
## Residuals:
##     Min     1Q  Median      3Q     Max
## -9.3177 -3.5428 -0.5591  2.3910 14.6836
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 41.283548   1.044352   39.53   <2e-16 ***
## horsepower  -0.169659   0.009556  -17.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.032 on 194 degrees of freedom
## Multiple R-squared:  0.619,  Adjusted R-squared:  0.6171
## F-statistic: 315.2 on 1 and 194 DF,  p-value: < 2.2e-16
```

```r
lm.pred = predict(lm.fit,Auto)
length(lm.pred)
```

```
## [1] 392
```

```r
length(Auto['mpg'][,])
```

```
## [1] 392
```

```r
mpgdata = (Auto['mpg'][,])
meanMSE = mean((mpgdata[-trainAuto]-lm.pred[-trainAuto])^2)
print(meanMSE)
```

```
## [1] 23.26601
```

```r
lm.fit.Quad = lm(mpg~poly(horsepower,2),data=Auto,subset = trainAuto)

lm.pred.Quad = predict(lm.fit.Quad,Auto)
length(lm.pred.Quad)
```

```
## [1] 392
```

```r
length(Auto['mpg'][,])
```

```
## [1] 392
```

```r
mpgdata.Quad = (Auto['mpg'][,])

meanMSE.Quad = mean((mpgdata.Quad[-trainAuto]-lm.pred.Quad[-trainAuto])^2)
print(meanMSE.Quad)
```

```
## [1] 18.71646
```

```r
lm.fit.Tri = lm(mpg~poly(horsepower,3),data=Auto,subset = trainAuto)

lm.pred.Tri = predict(lm.fit.Tri,Auto)
length(lm.pred.Tri)
```

```
## [1] 392
```

```r
length(Auto['mpg'][,])
```

```
## [1] 392
```

```r
mpgdata.Tri = (Auto['mpg'][,])

meanMSE.Tri = mean((mpgdata.Tri[-trainAuto]-lm.pred.Tri[-trainAuto])^2)

print(meanMSE.Quad)
```

```
## [1] 18.71646
```

```r
print(meanMSE.Tri)
```

```
## [1] 18.79401
```

```r
lm.fit.new = lm(mpg~horsepower,data=Auto,subset=trainAuto)

mpgpred.new = predict(lm.fit.new,Auto)
mpgdata = (Auto['mpg'][,])

meanMSE.new = mean((mpgdata[-trainAuto]-mpgpred.new[-trainAuto])^2)
print(meanMSE.new)
```

```
## [1] 23.26601
```