Logistic Regression

2024-02-06

#Logistic Regression  
library(mlbench)  
library(MASS)  
library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(readr)  
pima\_indians\_diabetes\_2 <- read\_csv("C:/Users/Lakshya Singh/Downloads/pima-indians-diabetes-2.csv")

## Rows: 768 Columns: 9

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (9): Preg, Plas, Pres, skin, test, mass, pedi, age, class  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

data = pima\_indians\_diabetes\_2  
head(data)

## # A tibble: 6 × 9  
## Preg Plas Pres skin test mass pedi age class  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 6 148 72 35 0 33.6 0.627 50 1  
## 2 1 85 66 29 0 26.6 0.351 31 0  
## 3 8 183 64 0 0 23.3 0.672 32 1  
## 4 1 89 66 23 94 28.1 0.167 21 0  
## 5 0 137 40 35 168 43.1 2.29 33 1  
## 6 5 116 74 0 0 25.6 0.201 30 0

#Descriptive Statistics  
summary(data)

## Preg Plas Pres skin   
## Min. : 0.000 Min. : 0.0 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 1.000 1st Qu.: 99.0 1st Qu.: 62.00 1st Qu.: 0.00   
## Median : 3.000 Median :117.0 Median : 72.00 Median :23.00   
## Mean : 3.845 Mean :120.9 Mean : 69.11 Mean :20.54   
## 3rd Qu.: 6.000 3rd Qu.:140.2 3rd Qu.: 80.00 3rd Qu.:32.00   
## Max. :17.000 Max. :199.0 Max. :122.00 Max. :99.00   
## test mass pedi age   
## Min. : 0.0 Min. : 0.00 Min. :0.0780 Min. :21.00   
## 1st Qu.: 0.0 1st Qu.:27.30 1st Qu.:0.2437 1st Qu.:24.00   
## Median : 30.5 Median :32.00 Median :0.3725 Median :29.00   
## Mean : 79.8 Mean :31.99 Mean :0.4719 Mean :33.24   
## 3rd Qu.:127.2 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00   
## Max. :846.0 Max. :67.10 Max. :2.4200 Max. :81.00   
## class   
## Min. :0.000   
## 1st Qu.:0.000   
## Median :0.000   
## Mean :0.349   
## 3rd Qu.:1.000   
## Max. :1.000

#Remove missing data  
newdata = na.omit(data)  
summary(newdata)

## Preg Plas Pres skin   
## Min. : 0.000 Min. : 0.0 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 1.000 1st Qu.: 99.0 1st Qu.: 62.00 1st Qu.: 0.00   
## Median : 3.000 Median :117.0 Median : 72.00 Median :23.00   
## Mean : 3.845 Mean :120.9 Mean : 69.11 Mean :20.54   
## 3rd Qu.: 6.000 3rd Qu.:140.2 3rd Qu.: 80.00 3rd Qu.:32.00   
## Max. :17.000 Max. :199.0 Max. :122.00 Max. :99.00   
## test mass pedi age   
## Min. : 0.0 Min. : 0.00 Min. :0.0780 Min. :21.00   
## 1st Qu.: 0.0 1st Qu.:27.30 1st Qu.:0.2437 1st Qu.:24.00   
## Median : 30.5 Median :32.00 Median :0.3725 Median :29.00   
## Mean : 79.8 Mean :31.99 Mean :0.4719 Mean :33.24   
## 3rd Qu.:127.2 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00   
## Max. :846.0 Max. :67.10 Max. :2.4200 Max. :81.00   
## class   
## Min. :0.000   
## 1st Qu.:0.000   
## Median :0.000   
## Mean :0.349   
## 3rd Qu.:1.000   
## Max. :1.000

set.seed(12)  
data1 = sort(sample(nrow(newdata), nrow(newdata)\*0.7))  
data1

## [1] 1 2 3 8 10 12 13 16 17 18 19 20 24 25 28 29 30 31  
## [19] 32 34 35 36 38 40 42 43 44 45 48 49 50 52 53 54 56 57  
## [37] 58 60 61 62 63 64 65 66 69 71 72 73 74 75 77 79 81 82  
## [55] 84 85 86 88 91 92 93 96 100 103 104 105 109 111 112 113 114 115  
## [73] 118 119 121 122 123 125 126 127 129 131 132 133 134 136 137 138 139 140  
## [91] 141 143 144 145 146 147 148 149 150 151 152 154 155 156 157 158 159 161  
## [109] 162 163 164 165 167 168 169 170 173 174 175 176 178 180 181 183 184 185  
## [127] 186 187 188 189 191 192 193 194 195 196 198 199 201 202 203 204 205 206  
## [145] 210 211 212 213 214 216 217 218 219 221 222 223 224 226 228 231 233 235  
## [163] 236 239 241 242 243 244 245 247 248 250 252 253 254 255 257 258 259 260  
## [181] 261 263 264 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280  
## [199] 282 287 289 291 292 293 294 295 296 299 300 302 303 304 305 306 308 309  
## [217] 311 312 313 314 319 321 322 323 324 325 326 327 328 329 333 334 335 336  
## [235] 338 342 343 344 345 346 347 349 350 351 352 353 354 355 356 360 361 362  
## [253] 363 364 365 366 368 369 370 372 373 375 376 378 379 380 381 382 383 384  
## [271] 385 386 388 389 392 394 395 396 397 398 399 402 403 404 408 409 411 413  
## [289] 414 415 416 417 419 421 422 423 424 425 426 427 428 429 430 431 432 433  
## [307] 434 437 438 439 440 441 442 443 445 446 447 448 449 450 451 452 453 455  
## [325] 456 457 458 459 460 461 462 463 464 466 467 468 472 473 474 475 476 477  
## [343] 480 482 484 487 488 490 491 492 493 495 497 499 501 502 503 505 508 509  
## [361] 510 512 513 514 515 516 517 518 520 521 523 524 526 527 528 531 532 534  
## [379] 535 536 537 539 540 541 542 543 544 545 546 550 551 552 553 555 557 558  
## [397] 560 562 563 564 565 566 568 569 570 571 572 574 575 576 577 579 581 582  
## [415] 583 584 585 586 587 589 590 591 593 594 595 596 599 600 601 602 603 605  
## [433] 606 607 609 611 612 615 616 617 618 619 620 621 622 623 624 626 627 628  
## [451] 629 630 632 633 637 639 640 643 644 645 648 649 650 651 652 653 654 655  
## [469] 656 657 658 662 663 665 666 667 668 670 673 674 676 678 679 680 683 687  
## [487] 688 689 691 692 693 694 696 697 698 699 703 705 706 708 710 711 714 715  
## [505] 718 719 720 722 723 724 729 733 734 735 736 737 739 740 741 743 744 746  
## [523] 747 749 750 751 752 754 755 756 760 761 763 764 765 766 767

#creating training data set by selecting the output row  
train = newdata[data1,]; train

## # A tibble: 537 × 9  
## Preg Plas Pres skin test mass pedi age class  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 6 148 72 35 0 33.6 0.627 50 1  
## 2 1 85 66 29 0 26.6 0.351 31 0  
## 3 8 183 64 0 0 23.3 0.672 32 1  
## 4 10 115 0 0 0 35.3 0.134 29 0  
## 5 8 125 96 0 0 0 0.232 54 1  
## 6 10 168 74 0 0 38 0.537 34 1  
## 7 10 139 80 0 0 27.1 1.44 57 0  
## 8 7 100 0 0 0 30 0.484 32 1  
## 9 0 118 84 47 230 45.8 0.551 31 1  
## 10 7 107 74 0 0 29.6 0.254 31 1  
## # ℹ 527 more rows

#creating testing data set by not selecting the output row  
test = newdata[-data1,]; test

## # A tibble: 231 × 9  
## Preg Plas Pres skin test mass pedi age class  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 89 66 23 94 28.1 0.167 21 0  
## 2 0 137 40 35 168 43.1 2.29 33 1  
## 3 5 116 74 0 0 25.6 0.201 30 0  
## 4 3 78 50 32 88 31 0.248 26 1  
## 5 2 197 70 45 543 30.5 0.158 53 1  
## 6 4 110 92 0 0 37.6 0.191 30 0  
## 7 1 189 60 23 846 30.1 0.398 59 1  
## 8 5 166 72 19 175 25.8 0.587 51 1  
## 9 3 126 88 41 235 39.3 0.704 27 0  
## 10 8 99 84 0 0 35.4 0.388 50 0  
## # ℹ 221 more rows

dim(train)

## [1] 537 9

dim(test)

## [1] 231 9

#fit a logistic regression model with the training model  
logmodel = glm(class ~., data=train, family = binomial(link = "logit"))  
summary(logmodel)

##   
## Call:  
## glm(formula = class ~ ., family = binomial(link = "logit"), data = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.450405 0.931772 -10.142 < 2e-16 \*\*\*  
## Preg 0.158708 0.042401 3.743 0.000182 \*\*\*  
## Plas 0.036402 0.004598 7.917 2.44e-15 \*\*\*  
## Pres -0.016463 0.006843 -2.406 0.016134 \*   
## skin 0.004965 0.008711 0.570 0.568657   
## test -0.001431 0.001142 -1.253 0.210223   
## mass 0.112415 0.019616 5.731 1.00e-08 \*\*\*  
## pedi 1.351376 0.383737 3.522 0.000429 \*\*\*  
## age 0.010728 0.012110 0.886 0.375692   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 675.00 on 536 degrees of freedom  
## Residual deviance: 467.72 on 528 degrees of freedom  
## AIC: 485.72  
##   
## Number of Fisher Scoring iterations: 5

#to predict using logistic regression model, probabilities obtained  
prediction = predict(logmodel, test, type = "response")  
head(prediction, 10)

## 1 2 3 4 5 6 7   
## 0.02795087 0.95699831 0.10145184 0.05592582 0.63248426 0.17958290 0.51460872   
## 8 9 10   
## 0.57080032 0.42307668 0.28503994

prediction\_threshold = ifelse(prediction >0.5,1,0)  
head(prediction\_threshold, 10)

## 1 2 3 4 5 6 7 8 9 10   
## 0 1 0 0 1 0 1 1 0 0