Evaluating Logistic Regression Models

Logistic Regression example

This is the R code from the R-bloggers (https://www.r-bloggers.com/) post Evaluating Logisitic Regression Models (https://www.r-bloggers.com/evaluating-logistic-regression-models/).

This post uses the same German credit data that is used in the book.

In this post the caret (http://topepo.github.io/caret/index.html) package is used to split, train and predict using functions from the package.

A logisitic regression model is fitted to the data to predict default.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
data(GermanCredit)
head(GermanCredit)
```

```
##   Duration Amount InstallmentRatePercentage ResidenceDuration Age
## 1        6   1169                         4                 4  67
## 2       48   5951                         2                 2  22
## 3       12   2096                         2                 3  49
## 4       42   7882                         2                 4  45
## 5       24   4870                         3                 4  53
## 6       36   9055                         2                 4  35
##   NumberExistingCredits NumberPeopleMaintenance Telephone Foreign
Worker
## 1                     2                       1         0
```

```
1
## 2                    1                       1         1
1
## 3                    1                       2         1
1
## 4                    1                       2         1
1
## 5                    2                       2         1
1
## 6                    1                       2         0
1
##   Class CheckingAccountStatus.lt.0 CheckingAccountStatus.0.to.200
## 1  Good                          1                              0
## 2   Bad                          0                              1
## 3  Good                          0                              0
## 4  Good                          1                              0
## 5   Bad                          1                              0
## 6  Good                          0                              0
##   CheckingAccountStatus.gt.200 CheckingAccountStatus.none
## 1                            0                          0
## 2                            0                          0
## 3                            0                          1
## 4                            0                          0
## 5                            0                          0
## 6                            0                          1
##   CreditHistory.NoCredit.AllPaid CreditHistory.ThisBank.AllPaid
## 1                              0                              0
## 2                              0                              0
## 3                              0                              0
## 4                              0                              0
## 5                              0                              0
## 6                              0                              0
##   CreditHistory.PaidDuly CreditHistory.Delay CreditHistory.Critic
al
## 1                      0                   0
1
## 2                      1                   0
0
## 3                      0                   0
1
## 4                      1                   0
0
```

```
## 5                    0                   1
0
## 6                    1                   0
0
##   Purpose.NewCar Purpose.UsedCar Purpose.Furniture.Equipment
## 1              0               0                           0
## 2              0               0                           0
## 3              0               0                           0
## 4              0               0                           1
## 5              1               0                           0
## 6              0               0                           0
##   Purpose.Radio.Television Purpose.DomesticAppliance Purpose.Repa
irs
## 1                        1                         0
0
## 2                        1                         0
0
## 3                        0                         0
0
## 4                        0                         0
0
## 5                        0                         0
0
## 6                        0                         0
0
##   Purpose.Education Purpose.Vacation Purpose.Retraining Purpose.B
usiness
## 1                 0                0                  0
0
## 2                 0                0                  0
0
## 3                 1                0                  0
0
## 4                 0                0                  0
0
## 5                 0                0                  0
0
## 6                 1                0                  0
0
##   Purpose.Other SavingsAccountBonds.lt.100 SavingsAccountBonds.10
0.to.500
## 1             0                          0
```

```
0
## 2          0                          1
0
## 3          0                          1
0
## 4          0                          1
0
## 5          0                          1
0
## 6          0                          0
0
##   SavingsAccountBonds.500.to.1000 SavingsAccountBonds.gt.1000
## 1                               0                           0
## 2                               0                           0
## 3                               0                           0
## 4                               0                           0
## 5                               0                           0
## 6                               0                           0
##   SavingsAccountBonds.Unknown EmploymentDuration.lt.1
## 1                           1                       0
## 2                           0                       0
## 3                           0                       0
## 4                           0                       0
## 5                           0                       0
## 6                           1                       0
##   EmploymentDuration.1.to.4 EmploymentDuration.4.to.7
## 1                         0                         0
## 2                         1                         0
## 3                         0                         1
## 4                         0                         1
## 5                         1                         0
## 6                         1                         0
##   EmploymentDuration.gt.7 EmploymentDuration.Unemployed
## 1                       1                             0
## 2                       0                             0
## 3                       0                             0
## 4                       0                             0
## 5                       0                             0
## 6                       0                             0
##   Personal.Male.Divorced.Seperated Personal.Female.NotSingle
## 1                                0                         0
## 2                                0                         1
```

```
## 3                               0                          0
## 4                               0                          0
## 5                               0                          0
## 6                               0                          0
##   Personal.Male.Single Personal.Male.Married.Widowed
## 1                    1                             0
## 2                    0                             0
## 3                    1                             0
## 4                    1                             0
## 5                    1                             0
## 6                    1                             0
##   Personal.Female.Single OtherDebtorsGuarantors.None
## 1                      0                           1
## 2                      0                           1
## 3                      0                           1
## 4                      0                           0
## 5                      0                           1
## 6                      0                           1
##   OtherDebtorsGuarantors.CoApplicant OtherDebtorsGuarantors.Guara
ntor
## 1                                  0
0
## 2                                  0
0
## 3                                  0
0
## 4                                  0
1
## 5                                  0
0
## 6                                  0
0
##   Property.RealEstate Property.Insurance Property.CarOther
## 1                   1                  0                 0
## 2                   1                  0                 0
## 3                   1                  0                 0
## 4                   0                  1                 0
## 5                   0                  0                 0
## 6                   0                  0                 0
##   Property.Unknown OtherInstallmentPlans.Bank OtherInstallmentPla
ns.Stores
## 1                0                          0
```

```
0
## 2                0                          0
0
## 3                0                          0
0
## 4                0                          0
0
## 5                1                          0
0
## 6                1                          0
0
##   OtherInstallmentPlans.None Housing.Rent Housing.Own Housing.For
Free
## 1                          1            0           1
0
## 2                          1            0           1
0
## 3                          1            0           1
0
## 4                          1            0           0
1
## 5                          1            0           0
1
## 6                          1            0           0
1
##   Job.UnemployedUnskilled Job.UnskilledResident Job.SkilledEmploy
ee
## 1                       0                     0
1
## 2                       0                     0
1
## 3                       0                     1
0
## 4                       0                     0
1
## 5                       0                     0
1
## 6                       0                     1
0
##   Job.Management.SelfEmp.HighlyQualified
## 1                                      0
## 2                                      0
```

```
## 3                                        0
## 4                                        0
## 5                                        0
## 6                                        0
```

```
with(GermanCredit,table(Class))
```

```
## Class
##  Bad Good
##  300  700
```

Split the data.

```
Train <- createDataPartition(GermanCredit$Class, p=0.6, list=FALSE)
training <- GermanCredit[ Train, ]
testing <- GermanCredit[ -Train, ]
```

Fit the logistic regression model, that is a GLM using a binomial link, using the caret function train().

```
mod_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate +
Housing.Own +
                   CreditHistory.Critical,  data=training, method="g
lm", family="binomial")
```

Transform the coeficients from log-odds to odds.

```
exp(coef(mod_fit$finalModel))
```

```
##            (Intercept)                    Age           ForeignWor
ker
##              3.1475551              1.0119295               0.2162
119
##    Property.RealEstate            Housing.Own CreditHistory.Criti
cal
##              1.4981725              1.9448033               2.5952
555
```

Predict.

```
predict(mod_fit, newdata=testing)
```

```
##   [1] Good Good Good Good Bad  Good Good Good Good Good Good Good
Good Good
##  [15] Good Good Good Good Good Good Bad  Good Good Good Good Good
Good Good
##  [29] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
##  [43] Good Good Good Bad  Good Bad  Good Good Good Good Good Good
Good Good
##  [57] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
##  [71] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
##  [85] Good Bad  Good Good Good Good Good Good Good Good Good Good
Good Bad
##  [99] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [113] Good Bad  Good Good Good Good Good Good Good Good Good Good
Good Good
## [127] Good Good Good Good Good Good Good Good Bad  Good Bad  Good
Bad  Good
## [141] Good Good Good Bad  Good Good Good Good Good Good Good Good
Good Good
## [155] Good Good Good Good Good Good Good Good Bad  Good Good Good
Good Good
## [169] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [183] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [197] Good Bad  Good Good Bad  Good Good Good Good Good Good Good
Good Good
## [211] Good Good Good Good Good Good Good Good Good Bad  Good Bad
Good Bad
## [225] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [239] Good Good Good Good Good Good Good Bad  Good Good Good Good
Good Bad
## [253] Good Good Good Good Good Good Good Good Good Good Good Bad
```

```
Good Good
## [267] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [281] Good Good Bad  Good Good Good Good Good Good Good Good Good
Good Good
## [295] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [309] Bad  Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [323] Good Good Good Good Good Good Good Good Good Bad  Good Good
Good Good
## [337] Good Good Good Good Good Bad  Good Good Good Good Good Good
Good Good
## [351] Good Good Good Good Bad  Good Good Good Good Good Good Good
Good Good
## [365] Good Good Good Good Good Good Good Good Good Good Good Good
Good Good
## [379] Bad  Good Good Good Good Bad  Good Good Good Good Good Good
Good Good
## [393] Good Good Good Good Good Good Bad  Good
## Levels: Bad Good
```

```
predict(mod_fit, newdata=testing, type="prob")
```

```
##            Bad      Good
## 3   0.09802981 0.9019702
## 7   0.28724245 0.7127576
## 8   0.49245237 0.5075476
## 10  0.17278366 0.8272163
## 11  0.52208479 0.4779152
## 14  0.12504352 0.8749565
## 16  0.34078975 0.6592102
## 17  0.13441196 0.8655880
## 20  0.34345890 0.6565411
## 22  0.36791560 0.6320844
## 26  0.24759906 0.7524009
## 29  0.25204396 0.7479560
## 33  0.34613797 0.6538620
## 34  0.22360599 0.7763940
## 35  0.33813067 0.6618693
## 37  0.28161960 0.7183804
```

```
## 39   0.32759747 0.6724025
## 41   0.34613797 0.6538620
## 46   0.16123893 0.8387611
## 47   0.32239447 0.6776055
## 48   0.52799932 0.4720007
## 49   0.15492629 0.8450737
## 50   0.26569274 0.7343073
## 55   0.42773712 0.5722629
## 57   0.28967649 0.7103235
## 60   0.30120414 0.6987959
## 62   0.09698623 0.9030138
## 65   0.35695084 0.6430492
## 67   0.34882684 0.6511732
## 69   0.48652559 0.5134744
## 71   0.34613797 0.6538620
## 72   0.24706523 0.7529348
## 74   0.10674414 0.8932559
## 77   0.33548177 0.6645182
## 80   0.36791142 0.6320886
## 86   0.12109186 0.8789081
## 89   0.34613797 0.6538620
## 93   0.16447239 0.8355276
## 94   0.30874490 0.6912551
## 96   0.42483683 0.5751632
## 97   0.12375182 0.8762482
## 99   0.11253113 0.8874689
## 100  0.48948861 0.5105114
## 101  0.37622731 0.6237727
## 104  0.27212302 0.7278770
## 105  0.51912509 0.4808749
## 110  0.24981492 0.7501851
## 112  0.52799932 0.4720007
## 116  0.10014703 0.8998530
## 117  0.34613797 0.6538620
## 119  0.18142373 0.8185763
## 120  0.15964155 0.8403585
## 121  0.27269139 0.7273086
## 125  0.41591460 0.5840854
## 127  0.23886824 0.7611318
## 133  0.35423341 0.6457666
## 136  0.11018415 0.8898158
## 137  0.35423341 0.6457666
```

```
## 140  0.46582255 0.5341775
## 142  0.34613797 0.6538620
## 144  0.36791142 0.6320886
## 147  0.15492629 0.8450737
## 148  0.13719543 0.8628046
## 150  0.14436649 0.8556335
## 154  0.28644273 0.7135573
## 155  0.48948861 0.5105114
## 157  0.02322681 0.9767732
## 159  0.32499057 0.6750094
## 160  0.11492164 0.8850784
## 164  0.39048929 0.6095107
## 165  0.33021505 0.6697850
## 167  0.33813067 0.6618693
## 168  0.28460849 0.7153915
## 170  0.11859020 0.8814098
## 172  0.27034575 0.7296543
## 175  0.35695084 0.6430492
## 184  0.09698623 0.9030138
## 185  0.15964155 0.8403585
## 188  0.20177730 0.7982227
## 190  0.33813067 0.6618693
## 192  0.49541666 0.5045833
## 194  0.41016456 0.5898354
## 196  0.16284919 0.8371508
## 200  0.31980929 0.6801907
## 203  0.35695084 0.6430492
## 204  0.53390600 0.4660940
## 215  0.13719543 0.8628046
## 220  0.26129220 0.7387078
## 221  0.27034575 0.7296543
## 227  0.35423341 0.6457666
## 232  0.44523658 0.5547634
## 234  0.27269139 0.7273086
## 235  0.10561865 0.8943813
## 237  0.36515794 0.6348421
## 238  0.41616771 0.5838323
## 239  0.16123893 0.8387611
## 240  0.32239447 0.6776055
## 243  0.52504293 0.4749571
## 246  0.27269139 0.7273086
## 247  0.09492878 0.9050712
```

```
## 249  0.27034575 0.7296543
## 252  0.30453343 0.6954666
## 253  0.36241325 0.6375868
## 260  0.08995525 0.9100448
## 262  0.28241119 0.7175888
## 264  0.22360599 0.7763940
## 266  0.32759747 0.6724025
## 270  0.35967750 0.6403225
## 275  0.33548177 0.6645182
## 276  0.25654144 0.7434586
## 277  0.17620000 0.8238000
## 278  0.29705133 0.7029487
## 279  0.34078975 0.6592102
## 280  0.51023879 0.4897612
## 281  0.30120414 0.6987959
## 286  0.24981492 0.7501851
## 291  0.07416762 0.9258324
## 295  0.14436649 0.8556335
## 296  0.35695084 0.6430492
## 298  0.08742895 0.9125710
## 300  0.16610855 0.8338915
## 301  0.21000096 0.7899990
## 303  0.11135223 0.8886478
## 305  0.25150375 0.7484963
## 319  0.16610855 0.8338915
## 321  0.17278366 0.8272163
## 322  0.34078975 0.6592102
## 325  0.11253113 0.8874689
## 328  0.33548177 0.6645182
## 329  0.34345890 0.6565411
## 330  0.35152537 0.6484746
## 331  0.18873732 0.8112627
## 332  0.11983539 0.8801646
## 340  0.34345890 0.6565411
## 342  0.51912509 0.4808749
## 347  0.12887110 0.8711289
## 348  0.52799932 0.4720007
## 352  0.10380012 0.8961999
## 354  0.51320189 0.4867981
## 356  0.27742065 0.7225793
## 358  0.35695084 0.6430492
## 359  0.26569274 0.7343073
```

```
## 362  0.32759747 0.6724025
## 364  0.53390600 0.4660940
## 366  0.33021505 0.6697850
## 367  0.17448523 0.8255148
## 369  0.31467210 0.6853279
## 371  0.24759906 0.7524009
## 372  0.16447239 0.8355276
## 373  0.22342518 0.7765748
## 376  0.48652559 0.5134744
## 378  0.48948861 0.5105114
## 380  0.13579769 0.8642023
## 381  0.32239447 0.6776055
## 384  0.27034575 0.7296543
## 385  0.35695084 0.6430492
## 386  0.17792798 0.8220720
## 401  0.24103099 0.7589690
## 403  0.35423341 0.6457666
## 406  0.36791142 0.6320886
## 409  0.35695084 0.6430492
## 412  0.16123893 0.8387611
## 421  0.53685588 0.4631441
## 424  0.25204396 0.7479560
## 425  0.35967750 0.6403225
## 427  0.17109524 0.8289048
## 434  0.11613339 0.8838666
## 435  0.27269139 0.7273086
## 440  0.27034575 0.7296543
## 441  0.32239447 0.6776055
## 442  0.34613797 0.6538620
## 450  0.26821727 0.7317827
## 452  0.26338550 0.7366145
## 453  0.33548177 0.6645182
## 455  0.34345890 0.6565411
## 457  0.19781885 0.8021811
## 458  0.49245237 0.5075476
## 459  0.26801284 0.7319872
## 460  0.34078975 0.6592102
## 465  0.33548177 0.6645182
## 466  0.26358767 0.7364123
## 470  0.33284319 0.6671568
## 473  0.35152537 0.6484746
## 475  0.16447239 0.8355276
```

```
## 477  0.36241325 0.6375868
## 478  0.35967750 0.6403225
## 480  0.14732102 0.8526790
## 481  0.12887110 0.8711289
## 482  0.35695084 0.6430492
## 483  0.42773712 0.5722629
## 487  0.28967649 0.7103235
## 489  0.12282616 0.8771738
## 492  0.47172882 0.5282712
## 498  0.15338001 0.8466200
## 501  0.35423341 0.6457666
## 502  0.47172882 0.5282712
## 503  0.22000712 0.7799929
## 505  0.52504293 0.4749571
## 510  0.30705088 0.6929491
## 511  0.27034575 0.7296543
## 512  0.50134600 0.4986540
## 524  0.36515794 0.6348421
## 525  0.27034575 0.7296543
## 526  0.34613797 0.6538620
## 527  0.16775772 0.8322423
## 528  0.10561865 0.8943813
## 530  0.23671884 0.7632812
## 531  0.34078975 0.6592102
## 536  0.27684618 0.7231538
## 538  0.15805701 0.8419430
## 543  0.34345890 0.6565411
## 546  0.37067774 0.6293223
## 547  0.15805701 0.8419430
## 548  0.34078975 0.6592102
## 549  0.27504971 0.7249503
## 553  0.25204396 0.7479560
## 554  0.17448523 0.8255148
## 557  0.35152537 0.6484746
## 558  0.34882684 0.6511732
## 562  0.52504293 0.4749571
## 565  0.32759747 0.6724025
## 566  0.52799932 0.4720007
## 567  0.33021505 0.6697850
## 573  0.51912509 0.4808749
## 575  0.35423341 0.6457666
## 580  0.35423341 0.6457666
```

```
## 581  0.11983539 0.8801646
## 582  0.09802981 0.9019702
## 583  0.41879833 0.5812017
## 586  0.43621085 0.5637892
## 589  0.22410422 0.7758958
## 592  0.47172882 0.5282712
## 602  0.34613797 0.6538620
## 606  0.27980414 0.7201959
## 607  0.07475907 0.9252409
## 611  0.36791142 0.6320886
## 612  0.45404032 0.5459597
## 613  0.34882684 0.6511732
## 614  0.19662506 0.8033749
## 618  0.19594374 0.8040563
## 625  0.40468993 0.5953101
## 627  0.23671884 0.7632812
## 630  0.19100253 0.8089975
## 631  0.35152537 0.6484746
## 632  0.30957987 0.6904201
## 634  0.53980319 0.4601968
## 639  0.33548177 0.6645182
## 642  0.32499057 0.6750094
## 644  0.16447239 0.8355276
## 649  0.48356351 0.5164365
## 654  0.31467210 0.6853279
## 656  0.53095374 0.4690463
## 659  0.35152537 0.6484746
## 663  0.21797883 0.7820212
## 665  0.13860526 0.8613947
## 666  0.17448523 0.8255148
## 667  0.33548177 0.6645182
## 668  0.26801284 0.7319872
## 674  0.17966922 0.8203308
## 676  0.29376929 0.7062307
## 683  0.25428615 0.7457139
## 684  0.24486580 0.7551342
## 686  0.49541666 0.5045833
## 688  0.50727497 0.4927250
## 690  0.34345890 0.6565411
## 693  0.34882684 0.6511732
## 695  0.35152537 0.6484746
## 696  0.44816762 0.5518324
```

```
## 699  0.17966922 0.8203308
## 700  0.47764303 0.5223570
## 706  0.49245237 0.5075476
## 710  0.24539643 0.7546036
## 713  0.22617301 0.7738270
## 715  0.35423341 0.6457666
## 716  0.12120278 0.8787972
## 717  0.15338001 0.8466200
## 720  0.34345890 0.6565411
## 723  0.27504971 0.7249503
## 729  0.42194171 0.5780583
## 733  0.23035103 0.7696490
## 741  0.34078975 0.6592102
## 744  0.27980414 0.7201959
## 747  0.52799932 0.4720007
## 748  0.24539643 0.7546036
## 749  0.35695084 0.6430492
## 751  0.29705133 0.7029487
## 752  0.36515794 0.6348421
## 755  0.43355244 0.5664476
## 757  0.01716987 0.9828301
## 759  0.25880980 0.7411902
## 762  0.22137432 0.7786257
## 765  0.17448523 0.8255148
## 769  0.28887274 0.7111273
## 770  0.08429782 0.9157022
## 771  0.41879833 0.5812017
## 772  0.17792798 0.8220720
## 773  0.15964155 0.8403585
## 775  0.20562441 0.7943756
## 776  0.42168763 0.5783124
## 782  0.08708499 0.9129150
## 783  0.25880980 0.7411902
## 784  0.36515794 0.6348421
## 785  0.19843995 0.8015600
## 788  0.15648527 0.8435147
## 789  0.44816762 0.5518324
## 790  0.17448523 0.8255148
## 792  0.31723513 0.6827649
## 794  0.44523658 0.5547634
## 795  0.50134600 0.4986540
## 804  0.16123893 0.8387611
```

```
## 805  0.42458229 0.5754177
## 806  0.36241325 0.6375868
## 813  0.17966922 0.8203308
## 814  0.28482066 0.7151793
## 815  0.45992583 0.5400742
## 816  0.43646708 0.5635329
## 817  0.19469432 0.8053057
## 819  0.31212031 0.6878797
## 821  0.26801284 0.7319872
## 826  0.16941994 0.8305801
## 829  0.45698157 0.5430184
## 835  0.27269139 0.7273086
## 837  0.37067355 0.6293265
## 838  0.42748213 0.5725179
## 843  0.27742065 0.7225793
## 845  0.30202763 0.6979724
## 846  0.33284319 0.6671568
## 847  0.39614876 0.6038512
## 849  0.20033300 0.7996670
## 851  0.07644518 0.9235548
## 853  0.27212302 0.7278770
## 854  0.50134600 0.4986540
## 855  0.30705088 0.6929491
## 856  0.25428615 0.7457139
## 858  0.28887274 0.7111273
## 859  0.34882684 0.6511732
## 862  0.35152537 0.6484746
## 863  0.24981492 0.7501851
## 864  0.11735622 0.8826438
## 865  0.42168763 0.5783124
## 868  0.15032539 0.8496746
## 870  0.52504293 0.4749571
## 872  0.02377103 0.9762290
## 873  0.12492959 0.8750704
## 874  0.27504971 0.7249503
## 876  0.15338001 0.8466200
## 877  0.48948861 0.5105114
## 880  0.15964155 0.8403585
## 881  0.32499057 0.6750094
## 885  0.31212031 0.6878797
## 887  0.16284919 0.8371508
## 891  0.14881695 0.8511831
```

```
## 892  0.14436649 0.8556335
## 894  0.16284919 0.8371508
## 897  0.51320189 0.4867981
## 898  0.03990483 0.9600952
## 899  0.39873927 0.6012607
## 900  0.31467210 0.6853279
## 901  0.25374276 0.7462572
## 902  0.10339905 0.8966010
## 904  0.37901443 0.6209856
## 905  0.24759906 0.7524009
## 907  0.07581279 0.9241872
## 908  0.35423341 0.6457666
## 913  0.35967750 0.6403225
## 914  0.26569274 0.7343073
## 917  0.25654144 0.7434586
## 919  0.33813067 0.6618693
## 920  0.48060259 0.5193974
## 921  0.17278366 0.8272163
## 922  0.32759747 0.6724025
## 925  0.28241119 0.7175888
## 926  0.30453343 0.6954666
## 927  0.30621970 0.6937803
## 932  0.36791142 0.6320886
## 934  0.15032539 0.8496746
## 935  0.36515794 0.6348421
## 937  0.26569274 0.7343073
## 938  0.50727497 0.4927250
## 944  0.10561865 0.8943813
## 954  0.35695084 0.6430492
## 955  0.34882684 0.6511732
## 959  0.30705088 0.6929491
## 960  0.50727497 0.4927250
## 961  0.20935312 0.7906469
## 964  0.33284319 0.6671568
## 967  0.18142373 0.8185763
## 970  0.10788017 0.8921198
## 977  0.19100253 0.8089975
## 978  0.31467210 0.6853279
## 981  0.14002722 0.8599728
## 982  0.49838128 0.5016187
## 984  0.35695084 0.6430492
## 985  0.16941994 0.8305801
```

```
## 986  0.29623565 0.7037643
## 988  0.19100253 0.8089975
## 994  0.34613797 0.6538620
## 996  0.25880980 0.7411902
## 999  0.52799932 0.4720007
## 1000 0.17448523 0.8255148
```

Model Evaluation and Diagnostics

Fit two models with the R function glm().

```
mod_fit_one <- glm(Class ~ Age + ForeignWorker + Property.RealEstate
+ Housing.Own +
                     CreditHistory.Critical, data=training, family="
binomial")
```

```
anova(mod_fit_one)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Class
##
## Terms added sequentially (first to last)
##
##
##                        Df Deviance Resid. Df Resid. Dev
## NULL                                     599     733.04
## Age                     1   4.7017       598     728.34
## ForeignWorker           1   8.4069       597     719.93
## Property.RealEstate     1   5.0428       596     714.89
## Housing.Own             1  14.1010       595     700.78
## CreditHistory.Critical  1  17.6791       594     683.11
```

```
mod_fit_two <- glm(Class ~ Age + ForeignWorker, data=training, famil
y="binomial")
```

```
anova(mod_fit_two)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Class
##
## Terms added sequentially (first to last)
##
##
##               Df Deviance Resid. Df Resid. Dev
## NULL                            599     733.04
## Age            1   4.7017       598     728.34
## ForeignWorker  1   8.4069       597     719.93
```

Goodness-of-Fit

Likelihood Ration Test

```
anova(mod_fit_one, mod_fit_two, test ="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Class ~ Age + ForeignWorker + Property.RealEstate + Hous
ing.Own +
##     CreditHistory.Critical
## Model 2: Class ~ Age + ForeignWorker
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       594     683.11
## 2       597     719.93 -3  -36.823 5.016e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lrtest(mod_fit_one, mod_fit_two)
```

```
## Likelihood ratio test
##
## Model 1: Class ~ Age + ForeignWorker + Property.RealEstate + Hous
ing.Own +
##     CreditHistory.Critical
## Model 2: Class ~ Age + ForeignWorker
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   6 -341.55
## 2   3 -359.96 -3 36.823  5.016e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pseudo R^2.

```
library(pscl)
```

```
## Loading required package: MASS
```

```
## Classes and Methods for R developed in the
```

```
## Political Science Computational Laboratory
```

```
## Department of Political Science
```

```
## Stanford University
```

```
## Simon Jackman
```

```
## hurdle and zeroinfl functions by Achim Zeileis
```

```
pR2(mod_fit_one)  # look for 'McFadden'
```

```
##          llh      llhNull          G2     McFadden
r2ML
## -341.55286427 -366.51858123   49.93143393    0.06811583    0.0798
5044
##         r2CU
##   0.11321811
```

Hosmer-Lemeshow Test

```
library(MKmisc)
HLgof.test(fit = fitted(mod_fit_one), obs = training$Class)
```

```
## Warning in Ops.factor(1, obs): '-' not meaningful for factors

## Warning in Ops.factor(1, obs): '-' not meaningful for factors
```

```
## $C
##
##  Hosmer-Lemeshow C statistic
##
## data:  fitted(mod_fit_one) and training$Class
## X-squared = 600, df = 8, p-value < 2.2e-16
##
##
## $H
##
##  Hosmer-Lemeshow H statistic
##
## data:  fitted(mod_fit_one) and training$Class
## X-squared = 600, df = 8, p-value < 2.2e-16
```

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-2   2017-02-28
```

```
hoslem.test(training$Class, fitted(mod_fit_one), g=10)
```

```
## Warning in Ops.factor(1, y): '-' not meaningful for factors
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  training$Class, fitted(mod_fit_one)
## X-squared = 600, df = 8, p-value < 2.2e-16
```

Statistical Tests for Individual Predictors

Wald Test

```
library(survey)
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
regTermTest(mod_fit_one, "ForeignWorker")
```

```
## Wald test for ForeignWorker
##  in glm(formula = Class ~ Age + ForeignWorker + Property.RealEsta
te +
##     Housing.Own + CreditHistory.Critical, family = "binomial",
##     data = training)
## F =  4.157408  on  1  and  594  df: p= 0.041895
```

Variable Importance

```
varImp(mod_fit)
```

```
## glm variable importance
##
##                           Overall
## CreditHistory.Critical  100.00
## Housing.Own               77.77
## ForeignWorker             24.55
## Property.RealEstate       16.12
## Age                        0.00
```

Validation of Predicted Values

Classification Rate

Accuracy.

```
pred = predict(mod_fit, newdata=testing)

accuracy <- table(pred, testing[,"Class"])
accuracy
```

```
##
## pred   Bad Good
##   Bad    9   19
##   Good 111  261
```

```
sum(diag(accuracy))/sum(accuracy)
```

```
## [1] 0.675
```

Confussion Matrix.

```
pred = predict(mod_fit, newdata=testing)
confusionMatrix(data=pred, testing$Class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Bad Good
##       Bad    9   19
##       Good 111  261
##
##                Accuracy : 0.675
##                  95% CI : (0.6267, 0.7207)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.8736
##
##                   Kappa : 0.0091
##  Mcnemar's Test P-Value : 1.449e-15
##
##             Sensitivity : 0.0750
##             Specificity : 0.9321
##          Pos Pred Value : 0.3214
##          Neg Pred Value : 0.7016
##              Prevalence : 0.3000
##          Detection Rate : 0.0225
##    Detection Prevalence : 0.0700
##       Balanced Accuracy : 0.5036
##
##        'Positive' Class : Bad
##
```

ROC

From the blog post, "The receiving operating characteristic is a measure of classifier performance. Using the proportion of positive data points that are correctly considered as positive and the proportion of negative data points that are mistakenly considered as positive, we generate a graphic that shows the trade off between the rate at which you can correctly predict something with the rate of incorrectly predicting something. Ultimately, we're concerned about the area under the ROC curve, or AUROC. That metric ranges from 0.50 to 1.00, and values above 0.80 indicate that the model does a good job in discriminating between the two categories which comprise our target variable."

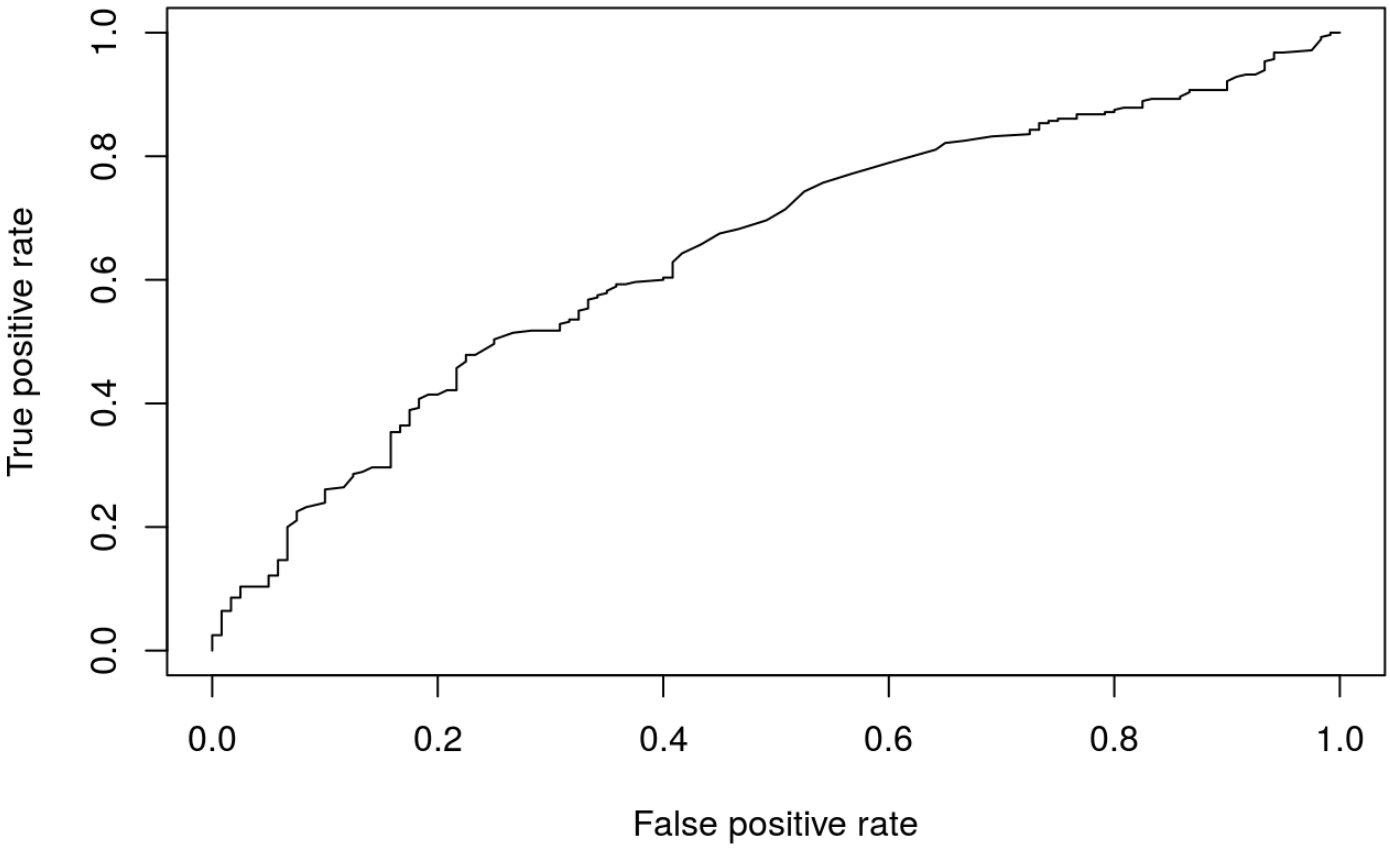
```r
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
# Compute AUC for predicting Class with the model
prob <- predict(mod_fit_one, newdata=testing, type="response")
pred <- prediction(prob, testing$Class)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf)
```




```
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.6469196
```

K-Fold Cross Validation

Split the data into k folds.

```
ctrl <- trainControl(method = "repeatedcv", number = 10, savePredict
ions = TRUE)

mod_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate +
Housing.Own +
                   CreditHistory.Critical,  data=GermanCredit, metho
d="glm", family="binomial",
                 trControl = ctrl, tuneLength = 5)

pred <- predict(mod_fit, newdata=testing)
confusionMatrix(data=pred, testing$Class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Bad Good
##       Bad    7   12
##       Good 113  268
##
##                Accuracy : 0.6875
##                  95% CI : (0.6396, 0.7326)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.7273
##
##                   Kappa : 0.0204
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.05833
##             Specificity : 0.95714
##          Pos Pred Value : 0.36842
##          Neg Pred Value : 0.70341
##              Prevalence : 0.30000
##          Detection Rate : 0.01750
##    Detection Prevalence : 0.04750
##       Balanced Accuracy : 0.50774
##
##        'Positive' Class : Bad
##
```

Boostrap

Randomly sample with replacement. B = 1000

```
start.time <- Sys.time()

ctrl <- trainControl(method = "boot632", number = 1000, savePredicti
ons = TRUE)

mod_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate +
Housing.Own +
                   CreditHistory.Critical,  data=GermanCredit, metho
d="glm", family="binomial",
                 trControl = ctrl, tuneLength = 5)

pred <- predict(mod_fit, newdata=testing)
confusionMatrix(data=pred, testing$Class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Bad Good
##       Bad    7   12
##       Good 113  268
##
##                Accuracy : 0.6875
##                  95% CI : (0.6396, 0.7326)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.7273
##
##                   Kappa : 0.0204
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.05833
##             Specificity : 0.95714
##          Pos Pred Value : 0.36842
##          Neg Pred Value : 0.70341
##              Prevalence : 0.30000
##          Detection Rate : 0.01750
##    Detection Prevalence : 0.04750
##       Balanced Accuracy : 0.50774
##
##        'Positive' Class : Bad
##
```

```
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

```
## Time difference of 14.84216 secs
```

Using parallel processing.

```
start.time <- Sys.time()

library(doMC)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
registerDoMC(cores = 4)

ctrl <- trainControl(method = "boot632", number = 1000, savePredicti
ons = TRUE,
                     allowParallel = TRUE)

mod_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate +
Housing.Own +
                   CreditHistory.Critical,  data=GermanCredit, metho
d="glm", family="binomial",
                 trControl = ctrl, tuneLength = 5)

pred <- predict(mod_fit, newdata=testing)
confusionMatrix(data=pred, testing$Class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Bad Good
##       Bad    7   12
##       Good 113  268
##
##                Accuracy : 0.6875
##                  95% CI : (0.6396, 0.7326)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.7273
##
##                   Kappa : 0.0204
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.05833
##             Specificity : 0.95714
##          Pos Pred Value : 0.36842
##          Neg Pred Value : 0.70341
##              Prevalence : 0.30000
##          Detection Rate : 0.01750
##    Detection Prevalence : 0.04750
##       Balanced Accuracy : 0.50774
##
##        'Positive' Class : Bad
##
```

```
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

```
## Time difference of 6.356072 secs
```