EN

2024-12-07

Elastic net

```
set.seed(42)
library(glmnet)
```

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

```
## Loaded glmnet 4.1-8
```

```
library(data.table)
train_data <- read.csv("train_data.csv")
test_data <- read.csv("test_data.csv")
# Prepare data for elastic net
#Train test split
x <- model.matrix(result ~ ., data = train_data)[,-1]
y <- train_data$result
x_test <- model.matrix(result ~ ., data = test_data)[, -1]
y_test <- test_data$result

# Fit elastic net regression
en_model <- cv.glmnet(as.matrix(x), y, family = "multinomial", alpha = 0.5,
                      type.measure = "class")

#Optimal lambda
en_model$lambda.min
```

```
## [1] 0.01042711
```

```
#Selected coefficients for each class (tie, win, loss)
coefs <- coef(en_model, s = en_model$lambda.min)
coefs
```

```
## $`0`
## 40 x 1 sparse Matrix of class "dgCMatrix"
##                                                                  1
## (Intercept)                                           -5.201259998
## possession.team                                        .
## possession.in.contest                                 -0.271558517
## total.attempts.team                                    0.016950624
## on.target.attempts.team                               -0.034544765
## off.target.attempts.team                              -0.029452585
## attempts.inside.the.penalty.area.team                  0.081873222
## attempts.outside.the.penalty.area..team                .
## left.channel.team                                     -0.052755758
## left.inside.channel.team                              -0.089319737
## central.channel.team                                   0.123658346
## right.inside.channel.team                            -0.021914475
## right.channel.team                                    .          
## total.offers.to.receive.team                          .          
## inbehind.offers.to.receive.team                      -0.018163062
## inbetween.offers.to.receive.team                      .          
## infront.offers.to.receive.team                        .          
## receptions.between.midfield.and.defensive.lines.team -0.001548933
## attempted.line.breaks.team                            0.007349975
## completed.line.breaks.team                           -0.002498571
## attempted.defensive.line.breaks.team                  0.107237472
## completed.defensive.line.breaks.team                  .          
## yellow.cards.team                                    -0.099386115
## red.cards.team                                       -0.229614964
## fouls.committed                                       0.091174935
## fouls.drawn                                           0.077652816
## offsides.team                                        -0.131378059
## passes.team                                           0.003721572
## passes.completed.team                                 0.000055232
## crosses.team                                          .          
## crosses.completed.team                               -0.093601229
## switches.of.play.completed.team                       .          
## corners.team                                          .          
## free.kicks.team                                       .          
## goal.preventions.team                                 .          
## forced.turnovers.team                                 0.063448691
## defensive.pressures.applied.team                      .          
## total.goals                                          -0.280894463
## total.attempts                                        .          
## total.attempted.defensive.line.breaks                 .          
## 
## $`1`
## 40 x 1 sparse Matrix of class "dgCMatrix"
##                                                                   1
## (Intercept)                                           4.5820472472
## possession.team                                      -0.0742935039
## possession.in.contest                                 0.0450872394
## total.attempts.team                                   .           
## on.target.attempts.team                               0.5277888276
## off.target.attempts.team                              .           
## attempts.inside.the.penalty.area.team                 .           
## attempts.outside.the.penalty.area..team              -0.1039322043
## left.channel.team                                     0.0249476506
## left.inside.channel.team                              .           
## central.channel.team                                 -0.0350247782
## right.inside.channel.team                             .           
## right.channel.team                                   -0.0069044491
## total.offers.to.receive.team                          .           
## inbehind.offers.to.receive.team                       0.0095927859
## inbetween.offers.to.receive.team                     -0.0048195429
## infront.offers.to.receive.team                        0.0009216114
## receptions.between.midfield.and.defensive.lines.team  .           
## attempted.line.breaks.team                           -0.0150222911
## completed.line.breaks.team                            0.0293018516
## attempted.defensive.line.breaks.team                  .           
## completed.defensive.line.breaks.team                  0.0592266414
## yellow.cards.team                                     .           
## red.cards.team                                        0.9124829614
## fouls.committed                                      -0.0409275837
## fouls.drawn                                           .           
## offsides.team                                         .           
## passes.team                                          -0.0005736958
## passes.completed.team                                 .           
## crosses.team                                         -0.1356517612
## crosses.completed.team                                .           
## switches.of.play.completed.team                       .           
## corners.team                                          0.0133965585
## free.kicks.team                                       .           
## goal.preventions.team                                -0.0080140246
## forced.turnovers.team                                 .           
## defensive.pressures.applied.team                      .           
## total.goals                                           .           
## total.attempts                                       -0.0331309041
## total.attempted.defensive.line.breaks                -0.0352115397
## 
## $‘2‘
## 40 x 1 sparse Matrix of class "dgCMatrix"
##                                                                  1
## (Intercept)                                           0.619212751
## possession.team                                       0.014468748
## possession.in.contest                                 .          
## total.attempts.team                                   .          
## on.target.attempts.team                              -0.113189037
## off.target.attempts.team                              0.100885891
## attempts.inside.the.penalty.area.team                -0.113128523
## attempts.outside.the.penalty.area..team               0.065284338
## left.channel.team                                     .          
## left.inside.channel.team                              0.169754916
## central.channel.team                                  .          
## right.inside.channel.team                             .          
## right.channel.team                                    .          
## total.offers.to.receive.team                          .          
## inbehind.offers.to.receive.team                       .          
## inbetween.offers.to.receive.team                      0.003450775
## infront.offers.to.receive.team                       -0.001606506
## receptions.between.midfield.and.defensive.lines.team  .          
## attempted.line.breaks.team                            .          
## completed.line.breaks.team                            .          
## attempted.defensive.line.breaks.team                 -0.017639144
## completed.defensive.line.breaks.team                 -0.001771919
## yellow.cards.team                                     0.053530294
## red.cards.team                                        .          
## fouls.committed                                       .          
## fouls.drawn                                          -0.063248754
## offsides.team                                         0.282118695
## passes.team                                           .          
## passes.completed.team                                 .          
## crosses.team                                          0.029791623
## crosses.completed.team                                0.092773913
## switches.of.play.completed.team                       .
## corners.team                                        -0.232368807
## free.kicks.team                                     -0.006910415
## goal.preventions.team                                0.097213578
## forced.turnovers.team                               -0.050036358
## defensive.pressures.applied.team                      .
## total.goals                                          0.047233677
## total.attempts                                        .
## total.attempted.defensive.line.breaks                0.015639976
```

```
# Predicted class probabilities
predicted_probs <- predict(en_model, newx = x_test, s = "lambda.min", type = "response")
predicted_probs
```

```
## , , 1
##
##              0          1           2
## 1  0.033934488 0.16484911 0.801216403
## 2  0.015252861 0.24193912 0.742808024
## 3  0.442119848 0.43717593 0.120704225
## 4  0.555256344 0.29297057 0.151773085
## 5  0.977762417 0.01707736 0.005160223
## 6  0.080038641 0.73960967 0.180351685
## 7  0.707418642 0.23972474 0.052856620
## 8  0.014142508 0.14783870 0.838018795
## 9  0.741362138 0.10169627 0.156941595
## 10 0.868240375 0.05716647 0.074593156
## 11 0.029111675 0.05393278 0.916955544
## 12 0.010924588 0.22228589 0.766789525
## 13 0.396770941 0.57484788 0.028381175
## 14 0.519227973 0.09072939 0.390042633
## 15 0.179654540 0.62399025 0.196355214
## 16 0.055791884 0.37446211 0.569746001
## 17 0.033744443 0.05489412 0.911361437
## 18 0.323169741 0.37429684 0.302533418
## 19 0.040959922 0.06078481 0.898255267
## 20 0.271318559 0.42237486 0.306306584
## 21 0.027038196 0.86151831 0.111443491
## 22 0.531280747 0.26480560 0.203913649
## 23 0.153319152 0.66385910 0.182821749
## 24 0.028960012 0.14710304 0.823936951
## 25 0.001366045 0.60301331 0.395620642
## 26 0.052622399 0.20689736 0.740480244
```

```
# Predicted class labels
predicted_classes <- predict(en_model, newx = x_test, s = "lambda.min", type = "class")
predicted_classes
```

```
##       1
##   [1,] "2"
##   [2,] "2"
##   [3,] "0"
##   [4,] "0"
##   [5,] "0"
##   [6,] "1"
##   [7,] "0"
##  [8,] "2"
##  [9,] "0"
## [10,] "0"
## [11,] "2"
## [12,] "2"
## [13,] "1"
## [14,] "0"
## [15,] "1"
## [16,] "2"
## [17,] "2"
## [18,] "1"
## [19,] "2"
## [20,] "1"
## [21,] "1"
## [22,] "0"
## [23,] "1"
## [24,] "2"
## [25,] "1"
## [26,] "2"
```

```
#Evaluate performance

# Confusion matrix
conf_matrix <- table(Predicted = predicted_classes, Actual = y_test)
conf_matrix
```

```
##          Actual
## Predicted 0 1 2
##         0 4 2 2
##         1 0 5 3
##         2 2 0 8
```

```
write.csv(conf_matrix, "en_conf.csv")

# Calculate accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
accuracy
```

```
## [1] 0.6538462
```

```
#Coefficients
coefs <- coef(en_model, s = "lambda.min")
#Coefficient df
en_coef_df <- data.frame(Variable=c(colnames(x)),
           coef0=coefs[1][[1]][2:40],
           coef1=coefs[2][[1]][2:40],
           coef2=coefs[3][[1]][2:40])
#Convert data to long format for plotting
long_df <- melt(data.table(en_coef_df), id.vars="Variable",
     variable.name = "Model", value.name = "Value")
write.csv(long_df, "coef_df.csv")
```

The accuracy of 65.38% is significantly greater than using multinomial logistic regression without variable selection. It appears that variable selection helped improve our model.

#Permutation feature importance

The coefficient values provide info on how important the variables were for prediction. Permutation feature importance (PFI) is another metric that can be used to assess the influence of a predictor on the model's prediction. PFI is the difference in model performance when one variable in the testing set is randomly permuted. Let's calculate PFI values for the elastic net model.

```r
test_data <- read.csv("test_data.csv")

set.seed(42)
#Vector to store PFI values
pfi_vect <- NULL

for (i in 1:ncol(test_data)) {
  #Clone test data
  test_data_per <- test_data
  #Randomly permute column i
  test_data_per[,i] <- sample(test_data_per[,i])
  #Create test data with randomly permuted column i
  x_test_per <- model.matrix(result ~ ., data = test_data_per)[, -1]

  # Predicted class labels
  predicted_classes_per <- predict(en_model, newx = x_test_per, s = "lambda.min",
                               type = "class")

  # Confusion matrix
  conf_matrix_per <- table(Predicted = predicted_classes_per, Actual = y_test)

  # Calculate accuracy
  accuracy_per <- sum(diag(conf_matrix_per)) / sum(conf_matrix_per)
  acc_diff <- accuracy_per - accuracy
  #Append to pfi vector
  pfi_vect <- c(pfi_vect, acc_diff)

}


#Create pfi df
pfi_df <- data.frame(Var=colnames(test_data),
                     PFI=pfi_vect)
write.csv(pfi_df, "pfi_df.csv")
```