

project

Qiao

November 11, 2019

```
library(caret)

## Loading required package: lattice
## Loading required package: ggplot2
library(glm2)
library(glmnet)

## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-16
library(Metrics)

##
## Attaching package: 'Metrics'
## The following object is masked from 'package:glmnet':
##
##      auc
## The following objects are masked from 'package:caret':
##
##      precision, recall
set.seed(1)
team <- read.csv("/home/qiaoqiao/Desktop/basketball/CSP571/final.csv" , sep=",", header=T)
train <- createDataPartition(team$pct, p = 0.8,list=FALSE)
team.train <- team[train, ]
team.test <- team[-train, ]
team.fit <- lm(pct~., data = team.train)
summary(team.fit)

##
## Call:
## lm(formula = pct ~ ., data = team.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23342 -0.05414 -0.01079  0.05376  0.20014
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.565e+00  1.708e+00  -2.673  0.00816 **
## OffPoss       -7.434e-05  1.523e-05  -4.881  2.2e-06 ***
## Fg2Pct        -1.400e-01  2.206e+00  -0.063  0.94947
## Fg3Pct         2.764e+00  5.179e+00   0.534  0.59422
## NonHeaveFg3Pct -1.634e+00  5.108e+00  -0.320  0.74932
## FtsMade        8.150e-03  7.286e-03   1.119  0.26472
```

```

## PtsAssisted2s          2.115e-02  2.266e-02  0.934  0.35172
## PtsUnassisted2s       -4.180e-03  2.225e-02 -0.188  0.85121
## PtsAssisted3s         -1.291e-02  1.699e-02 -0.760  0.44806
## PtsUnassisted3s        1.589e-02  1.807e-02  0.879  0.38042
## Assisted2sPct         -7.714e-01  2.311e+00 -0.334  0.73894
## Assisted3sPct          9.785e-01  4.681e-01  2.090  0.03791 *
## FG3APct                1.667e+00  2.039e+00  0.818  0.41458
## ShotQualityAvg         6.557e-01  8.878e-01  0.739  0.46109
## TsPct                  4.306e+00  2.366e+00  1.820  0.07027 .
## PtsPutbacks           -3.279e-02  1.194e-02 -2.746  0.00660 **
## Fg2aBlocked            1.097e-01  1.153e-01  0.952  0.34230
## FG2APctBlocked        -8.711e+00  7.096e+00 -1.227  0.22113
## Fg3aBlocked           -3.154e-01  4.232e-01 -0.745  0.45695
## FG3APctBlocked         7.400e+00  9.611e+00  0.770  0.44227
## AtRimAssists          -2.364e-02  1.274e-02 -1.855  0.06506 .
## ShortMidRangeAssists  -2.716e-02  1.508e-02 -1.802  0.07317 .
## Corner3Assists         3.708e-02  2.076e-02  1.786  0.07563 .
## Def2ptReboundPct       1.421e+00  1.156e+00  1.229  0.22075
## Def3ptReboundPct       3.470e+00  5.454e+00  0.636  0.52532
## OffFTReboundPct       -2.008e-01  1.719e-01 -1.168  0.24441
## Off2ptReboundPct       3.882e+00  1.652e+00  2.350  0.01977 *
## Off3ptReboundPct       4.942e+00  4.261e+00  1.160  0.24753
## DefAtRimReboundPct     4.067e-01  4.858e-01  0.837  0.40349
## DefShortMidRangeReboundPct 3.474e-02  4.854e-01  0.072  0.94301
## DefLongMidRangeReboundPct -2.586e-02  5.414e-01 -0.048  0.96195
## DefArc3ReboundPct      -2.608e+00  4.085e+00 -0.638  0.52397
## DefCorner3ReboundPct   -9.290e-01  1.393e+00 -0.667  0.50567
## OffAtRimReboundPct     -6.220e-01  6.059e-01 -1.027  0.30590
## OffShortMidRangeReboundPct -1.402e+00  6.168e-01 -2.273  0.02411 *
## OffLongMidRangeReboundPct -8.327e-01  6.235e-01 -1.336  0.18324
## OffArc3ReboundPct      -2.543e+00  3.249e+00 -0.783  0.43473
## OffCorner3ReboundPct    -1.026e+00  1.077e+00 -0.953  0.34196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08425 on 193 degrees of freedom
## Multiple R-squared:  0.7495, Adjusted R-squared:  0.7014
## F-statistic: 15.6 on 37 and 193 DF, p-value: < 2.2e-16

predict_0 <- predict(team.fit,team.train,type = "response")
prediction<-ifelse(predict_0<=0.5,0,1)
true<-ifelse(team.train$pct<=0.5,0,1)
confusionMatrix(table(prediction, true),mode="everything")

## Confusion Matrix and Statistics
##
##           true
## prediction  0  1
##           0 92 26
##           1 14 99
##
##              Accuracy : 0.8268
##              95% CI : (0.7718, 0.8733)
##              No Information Rate : 0.5411
##              P-Value [Acc > NIR] : < 2e-16

```

```
##
##           Kappa : 0.6543
## Mcnemar's Test P-Value : 0.08199
##
##           Sensitivity : 0.8679
##           Specificity : 0.7920
##           Pos Pred Value : 0.7797
##           Neg Pred Value : 0.8761
##           Precision : 0.7797
##           Recall : 0.8679
##           F1 : 0.8214
##           Prevalence : 0.4589
##           Detection Rate : 0.3983
##           Detection Prevalence : 0.5108
##           Balanced Accuracy : 0.8300
##
##           'Positive' Class : 0
##
```

```
mse(team.test$pct,predict_0)
```

```
## Warning in actual - predicted: longer object length is not a multiple of
## shorter object length
```

```
## [1] 0.03717523
```

```
predict_1 <- predict(team.fit,team.test,type = "response")
prediction<-ifelse(predict_1<=0.5,0,1)
true<-ifelse(team.test$pct<=0.5,0,1)
confusionMatrix(table(prediction, true),mode="everything")
```

```
## Confusion Matrix and Statistics
```

```
##
##           true
## prediction  0  1
##           0 23  4
##           1  4 25
##
##           Accuracy : 0.8571
##           95% CI : (0.7378, 0.9362)
##           No Information Rate : 0.5179
##           P-Value [Acc > NIR] : 9.573e-08
##
##           Kappa : 0.7139
## Mcnemar's Test P-Value : 1
##
##           Sensitivity : 0.8519
##           Specificity : 0.8621
##           Pos Pred Value : 0.8519
##           Neg Pred Value : 0.8621
##           Precision : 0.8519
##           Recall : 0.8519
##           F1 : 0.8519
##           Prevalence : 0.4821
##           Detection Rate : 0.4107
##           Detection Prevalence : 0.4821
```

```
##          Balanced Accuracy : 0.8570
##
##          'Positive' Class : 0
##
```

```
mse(team.test$pct,predict_1)
```

```
## [1] 0.007277765
```

```
predict_1 <- predict(team.fit,team.test,type = "response")
prediction<-ifelse(predict_1<=0.7,0,1)
true<-ifelse(team.test$pct<=0.7,0,1)
confusionMatrix(table(prediction, true),mode="everything")
```

```
## Confusion Matrix and Statistics
```

```
##
##           true
## prediction  0  1
##           0 50  2
##           1  2  2
##
##           Accuracy : 0.9286
##           95% CI : (0.8271, 0.9802)
##       No Information Rate : 0.9286
##       P-Value [Acc > NIR] : 0.6289
##
##           Kappa : 0.4615
##  Mcnemar's Test P-Value : 1.0000
##
##           Sensitivity : 0.9615
##           Specificity : 0.5000
##       Pos Pred Value : 0.9615
##       Neg Pred Value : 0.5000
##           Precision : 0.9615
##           Recall : 0.9615
##              F1 : 0.9615
##       Prevalence : 0.9286
##       Detection Rate : 0.8929
##       Detection Prevalence : 0.9286
##       Balanced Accuracy : 0.7308
##
##       'Positive' Class : 0
##
```

```
mse(team.test$pct,predict_1)
```

```
## [1] 0.007277765
```

```
# fit0 <- lm(pct~OffPoss+PtsPutbacks+Assisted3sPct+Off2ptReboundPct+OffShortMidRangeReboundPct, data =
# summary(fit0)
```

```
fit <- glm(pct~., data = team.train, family = binomial)
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial
## glm!
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = pct ~ ., family = binomial, data = team.train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49776  -0.11283  -0.01468   0.10745   0.40294
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.239e+01  4.241e+01  -0.528   0.598
## OffPoss        -3.097e-04  3.762e-04  -0.823   0.410
## Fg2Pct         -5.544e-01  5.444e+01  -0.010   0.992
## Fg3Pct          1.217e+01  1.286e+02   0.095   0.925
## NonHeaveFg3Pct  -7.806e+00  1.263e+02  -0.062   0.951
## FtsMade          3.110e-02  1.786e-01   0.174   0.862
## PtsAssisted2s     8.310e-02  5.657e-01   0.147   0.883
## PtsUnassisted2s  -1.563e-02  5.532e-01  -0.028   0.977
## PtsAssisted3s    -5.171e-02  4.283e-01  -0.121   0.904
## PtsUnassisted3s   8.455e-02  4.551e-01   0.186   0.853
## Assisted2sPct    -2.695e+00  5.740e+01  -0.047   0.963
## Assisted3sPct     4.449e+00  1.181e+01   0.377   0.706
## FG3APct          6.532e+00  5.123e+01   0.128   0.899
## ShotQualityAvg    2.775e+00  2.182e+01   0.127   0.899
## TsPct            1.909e+01  5.817e+01   0.328   0.743
## PtsPutbacks      -1.407e-01  2.953e-01  -0.476   0.634
## Fg2aBlocked       5.013e-01  2.867e+00   0.175   0.861
## FG2APctBlocked   -3.875e+01  1.764e+02  -0.220   0.826
## Fg3aBlocked      -1.446e+00  1.035e+01  -0.140   0.889
## FG3APctBlocked    3.366e+01  2.348e+02   0.143   0.886
## AtRimAssists     -1.001e-01  3.148e-01  -0.318   0.750
## ShortMidRangeAssists -1.153e-01  3.696e-01  -0.312   0.755
## Corner3Assists    1.635e-01  5.112e-01   0.320   0.749
## Def2ptReboundPct  6.702e+00  2.882e+01   0.233   0.816
## Def3ptReboundPct  1.370e+01  1.346e+02   0.102   0.919
## OffFTReboundPct  -8.161e-01  4.239e+00  -0.193   0.847
## Off2ptReboundPct  1.603e+01  4.056e+01   0.395   0.693
## Off3ptReboundPct  2.018e+01  1.060e+02   0.190   0.849
## DefAtRimReboundPct 1.488e+00  1.213e+01   0.123   0.902
## DefShortMidRangeReboundPct -1.654e-02  1.197e+01  -0.001   0.999
## DefLongMidRangeReboundPct -3.516e-01  1.336e+01  -0.026   0.979
## DefArc3ReboundPct -1.019e+01  1.009e+02  -0.101   0.920
## DefCorner3ReboundPct -3.639e+00  3.438e+01  -0.106   0.916
## OffAtRimReboundPct -2.344e+00  1.497e+01  -0.157   0.876
## OffShortMidRangeReboundPct -5.809e+00  1.517e+01  -0.383   0.702
## OffLongMidRangeReboundPct -3.342e+00  1.531e+01  -0.218   0.827
## OffArc3ReboundPct -1.038e+01  8.076e+01  -0.129   0.898
## OffCorner3ReboundPct -4.047e+00  2.681e+01  -0.151   0.880
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
## Null deviance: 22.8267 on 230 degrees of freedom
## Residual deviance: 5.8598 on 193 degrees of freedom
## AIC: 325.01
##
## Number of Fisher Scoring iterations: 4
predict_fit <- predict(fit, team.train,type ="response")
prediction_fit<-ifelse(predict_fit<=0.5,0,1)
true<-ifelse(team.train$pct<=0.5,0,1)
confusionMatrix(table(prediction_fit, true),mode="everything")
```

```
## Confusion Matrix and Statistics
##
##           true
## prediction_fit 0  1
##           0 93 25
##           1 13 100
##
##           Accuracy : 0.8355
##           95% CI : (0.7813, 0.8809)
##           No Information Rate : 0.5411
##           P-Value [Acc > NIR] : < 2e-16
##
##           Kappa : 0.6716
##           McNemar's Test P-Value : 0.07435
##
##           Sensitivity : 0.8774
##           Specificity : 0.8000
##           Pos Pred Value : 0.7881
##           Neg Pred Value : 0.8850
##           Precision : 0.7881
##           Recall : 0.8774
##           F1 : 0.8304
##           Prevalence : 0.4589
##           Detection Rate : 0.4026
##           Detection Prevalence : 0.5108
##           Balanced Accuracy : 0.8387
##
##           'Positive' Class : 0
##
```

```
mse(team.test$pct,predict_fit)
```

```
## Warning in actual - predicted: longer object length is not a multiple of
## shorter object length
```

```
## [1] 0.03725496
```

```
predict_fit_1 <- predict(fit, team.test,type ="response")
prediction_fit_1<-ifelse(predict_fit_1<=0.5,0,1)
true_1<-ifelse(team.test$pct<=0.5,0,1)
confusionMatrix(table(prediction_fit_1, true_1),mode="everything")
```

```
## Confusion Matrix and Statistics
##
##           true_1
```

```
## prediction_fit_1 0 1
##                0 23 4
##                1  4 25
##
##                Accuracy : 0.8571
##                95% CI : (0.7378, 0.9362)
##      No Information Rate : 0.5179
##      P-Value [Acc > NIR] : 9.573e-08
##
##                Kappa : 0.7139
##  McNemar's Test P-Value : 1
##
##      Sensitivity : 0.8519
##      Specificity : 0.8621
##      Pos Pred Value : 0.8519
##      Neg Pred Value : 0.8621
##      Precision : 0.8519
##      Recall : 0.8519
##      F1 : 0.8519
##      Prevalence : 0.4821
##      Detection Rate : 0.4107
##      Detection Prevalence : 0.4821
##      Balanced Accuracy : 0.8570
##
##      'Positive' Class : 0
##
```

```
mse(team.test$pct,predict_fit_1)
```

```
## [1] 0.00712028
```

```
predict_fit_2 <- predict(fit, team.test,type ="response")
prediction_fit_2<-ifelse(predict_fit_2<=0.7,0,1)
true_2<-ifelse(team.test$pct<=0.7,0,1)
confusionMatrix(table(prediction_fit_2, true_2),mode="everything")
```

```
## Confusion Matrix and Statistics
##
##                true_2
## prediction_fit_2 0 1
##                0 50 2
##                1  2 2
##
##                Accuracy : 0.9286
##                95% CI : (0.8271, 0.9802)
##      No Information Rate : 0.9286
##      P-Value [Acc > NIR] : 0.6289
##
##                Kappa : 0.4615
##  McNemar's Test P-Value : 1.0000
##
##      Sensitivity : 0.9615
##      Specificity : 0.5000
##      Pos Pred Value : 0.9615
##      Neg Pred Value : 0.5000
```

```
##           Precision : 0.9615
##           Recall   : 0.9615
##           F1        : 0.9615
##           Prevalence : 0.9286
##           Detection Rate : 0.8929
##           Detection Prevalence : 0.9286
##           Balanced Accuracy : 0.7308
##
##           'Positive' Class : 0
##
```

```
mse(team.test$pct,predict_fit_2)
```

```
## [1] 0.00712028
```

```
# fit1 <- glm(pct~OffPoss+PtsAssisted3s+Assisted3sPct+Corner3Assists, data = team.train, family = binom
# summary(fit1)
# predict_fit <- predict(fit1, team.train,type = "response")
# prediction<-ifelse(predict_fit<=0.5,0,1)
# true<-ifelse(team.train$pct<=0.5,0,1)
# confusionMatrix(table(prediction, true),mode="everything")

# grid<-seq(0,10,0.01)
# y <- team.train$pct
# x <- model.matrix(team.train$pct ~OffPoss+PtsPutbacks+Assisted3sPct+Off2ptReboundPct+OffShortMidRange
# lasso_fit <- glmnet(x, y, alpha = 1, lambda = grid)
# cross.validation <- cv.glmnet(x, y, alpha = 1,lambda = grid)
# minimum.lambda <- cross.validation$lambda.min
# cat("minimum lambda", minimum.lambda, "\n")
# x.test <- model.matrix(team.test$pct ~OffPoss+PtsPutbacks+Assisted3sPct+Off2ptReboundPct+OffShortMidR
# lasso_pred <- predict(lasso_fit, s = minimum.lambda, newx = x.test)
#
# cat("test MSE" , mse(team.test$pct, lasso_pred), "\n")
#
# lasso.mod <- glmnet(x, y, alpha = 1)
# coefs <- predict(lasso.mod, type = "coefficients", s = minimum.lambda)
# coefs
# library(corrplot)
# correlations <- cor(team[,2:8])
# corrplot(correlations, method="circle")
```