Cross Validation Playoffs

Rubens Cortelazzi Roncato

2024-05-07

```
source("dados_playoffs.R")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Loading required package: splines
##
## Loading required package: gamlss.data
##
##
## Attaching package: 'gamlss.data'
##
##
## The following object is masked from 'package:datasets':
##
##     sleep
##
##
## Loading required package: gamlss.dist
##
## Loading required package: nlme
##
##
## Attaching package: 'nlme'
##
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
##
## Loading required package: parallel
##
##  **********   GAMLSS Version 5.4-22  **********
##
## For more on GAMLSS look at https://www.gamlss.com/
##
```

```
## Type gamlssNews() to see new features/changes/bug fixes.
##
##
## Loading required package: carData
##
##
## Attaching package: 'car'
##
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
##
## The following object is masked from 'package:purrr':
##
##     some
##
##
## Loading required package: zoo
##
##
## Attaching package: 'zoo'
##
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:gamlss':
##
##     calibration
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
set.seed(236292)

#Vamos mudar o nome porque estava dando erro na hora do gamlss
dados_regressaop$WINP_transformado <- (dados_regressaop$WINP*(240 - 1) + 0.5)/240

#### Cross Validation K Fold #########
# O processo envolve dividir o conjunto de dados em 5 partes (ou folds) e,
# em seguida, iterar sobre esses folds. Em cada iteração, um dos folds é usado
# como conjunto de teste e os outros folds são usados como conjunto de
# treinamento. Essa é a essência da validação cruzada.

#No entanto, ao invés de fazer isso apenas uma vez, o código original executa
```

```r
# a validação cruzada 5 vezes (ou seja, 5 iterações externas), o que é
# conhecido como "validação cruzada repetida". Cada repetição é essencialmente
# uma validação cruzada separada. Isso ajuda a reduzir a variabilidade dos
# resultados, fornecendo uma estimativa mais robusta do desempenho do modelo.


# Definindo os modelos
# Primeiro, você define os modelos que deseja avaliar e as métricas de
# desempenho que deseja calcular. Os modelos são definidos como uma lista de
# objetos de modelo, onde a chave é o nome do modelo e o valor é o modelo
# em si. As métricas são definidas como um vetor de strings contendo os nomes
# das métricas que você deseja calcular.

# Definindo os modelos
modelos <- list(
  "modelo_forwp" = lm(formula = WINP ~ PlusMinus + DREB, data = dados_regressaop),
  "modelo_betapt_ftp" = betareg(formula = WINP_transformado ~ FTP + REB + PlusMinus, data = dados_regres
  "modelop_loglog_reb" = betareg(formula = WINP_transformado ~ REB + PlusMinus, data = dados_regressaop
  "modelop_probit_ftp" = betareg(formula = WINP_transformado ~ FTP + REB + PlusMinus, data = dados_regre
  "modelo_betat_cloglog_ftp" = betareg(formula = WINP_transformado ~ FTP + REB + PlusMinus, data = dados
  "modelo_betat_cauchit_ftp" = betareg(formula = WINP_transformado ~ FTP + PlusMinus, data = dados_regre
  "gamlss_betap_pf" = gamlss(formula = WINP ~ PF + PlusMinus, family = BEZI, data = dados_regressaop),
  "misto_normalp_dreb" = gamlss(formula = WINP ~ (re(random = ~1 | TEAM)) + PlusMinus + DREB, family = 
  "misto_normalp_temp_team" = gamlss(formula = WINP ~ (re(random = ~1 | Numero_temporada)) + PlusMinus 
  "misto_betap_ftp" = gamlss(formula = WINP ~ (re(random = ~1 | TEAM)) + PlusMinus + PF + BLKA + FTP, fa
  "misto_betap_temp" = gamlss(formula = WINP ~ (re(random = ~1 | Numero_temporada)) + PlusMinus + PF, fa
)
```

```
## GAMLSS-RS iteration 1: Global Deviance = -161.4236
## GAMLSS-RS iteration 2: Global Deviance = -329.2529
## GAMLSS-RS iteration 3: Global Deviance = -330.669
## GAMLSS-RS iteration 4: Global Deviance = -330.6692
## GAMLSS-RS iteration 1: Global Deviance = -436.3104
## GAMLSS-RS iteration 2: Global Deviance = -436.3104
## GAMLSS-RS iteration 1: Global Deviance = -473.5697
## GAMLSS-RS iteration 2: Global Deviance = -473.5697
## GAMLSS-RS iteration 1: Global Deviance = -170.0295
## GAMLSS-RS iteration 2: Global Deviance = -349.6334
## GAMLSS-RS iteration 3: Global Deviance = -350.6895
## GAMLSS-RS iteration 4: Global Deviance = -350.6399
## GAMLSS-RS iteration 5: Global Deviance = -350.6347
## GAMLSS-RS iteration 6: Global Deviance = -350.6342
## GAMLSS-RS iteration 1: Global Deviance = -161.4236
## GAMLSS-RS iteration 2: Global Deviance = -329.2529
## GAMLSS-RS iteration 3: Global Deviance = -330.669
## GAMLSS-RS iteration 4: Global Deviance = -330.6692
```

```r
# Definindo as métricas
metricas <- c("R2", "RMSE", "MAE")

# Lista para armazenar os resultados
resultados <- list()

# Loop para execução da validação cruzada
```

```r
for (nome_modelo in names(modelos)) {
  for (m in metricas) {
    resultados[[paste(nome_modelo, m, sep = "_")]] <- c()
  }
}

# Executando a validação cruzada 5 vezes
for (i in 1:5) {
  # Criando os folds para validação cruzada
  folds <- createFolds(dados_regressaop$WINP, k = 5, returnTrain = TRUE)

  # Loop para cada fold
  for (j in 1:length(folds)) {
    # Dividindo os dados em treino e teste
    training_index <- folds[[j]]
    testing_index <- setdiff(seq_len(nrow(dados_regressaop)), training_index)
    training_data <- dados_regressaop[training_index, ]
    testing_data <- dados_regressaop[testing_index, ]

    # Treinando e testando cada modelo
    for (nome_modelo in names(modelos)) {
      modelo <- modelos[[nome_modelo]]

      # Verificando se o modelo utiliza a variável transformada ou não
      formula_modelo <- formula(modelo)
      if (is.character(formula_modelo)) {
        formula_modelo <- as.formula(formula_modelo)
      }
      if ("WINP_transformado" %in% all.vars(formula_modelo)) {
        predict_test <- predict(modelo, newdata = testing_data, type = "response")
      } else {
        predict_test <- predict(modelo, newdata = testing_data)
      }

      for (m in metricas) {
        resultados[[paste(nome_modelo, m, sep = "_")]] <- c(resultados[[paste(nome_modelo, m, sep = "_"
                                                            ifelse(m == "R2", R2(predict_test, testing_
                                                                   ifelse(m == "RMSE", RMSE(predict_tes
                                                                          MAE(predict_test, testing_dat
      }
    }
  }
}

# Calculando a média das métricas para cada modelo
medias_resultados <- list()
for (nome_modelo in names(modelos)) {
  for (m in metricas) {
    medias_resultados[[paste(nome_modelo, m, sep = "_")]] <- mean(resultados[[paste(nome_modelo, m, sep
  }
}

# Exibindo as médias das métricas
```

```r
print("Médias das métricas para cada modelo:")
```

```
## [1] "Médias das métricas para cada modelo:"
```

```r
print(medias_resultados)
```

```
## $modelo_forwp_R2
## [1] 0.7472946
##
## $modelo_forwp_RMSE
## [1] 0.1026929
##
## $modelo_forwp_MAE
## [1] 0.07949132
##
## $modelo_betapt_ftp_R2
## [1] 0.7631798
##
## $modelo_betapt_ftp_RMSE
## [1] 0.1036046
##
## $modelo_betapt_ftp_MAE
## [1] 0.08530677
##
## $modelop_loglog_reb_R2
## [1] 0.7611173
##
## $modelop_loglog_reb_RMSE
## [1] 0.1022045
##
## $modelop_loglog_reb_MAE
## [1] 0.08324645
##
## $modelop_probit_ftp_R2
## [1] 0.765559
##
## $modelop_probit_ftp_RMSE
## [1] 0.1025
##
## $modelop_probit_ftp_MAE
## [1] 0.08403919
##
## $modelo_betat_cloglog_ftp_R2
## [1] 0.7513722
##
## $modelo_betat_cloglog_ftp_RMSE
## [1] 0.1065006
##
## $modelo_betat_cloglog_ftp_MAE
## [1] 0.08796576
##
## $modelo_betat_cauchit_ftp_R2
## [1] 0.7417763
##
```

```
## $modelo_betat_cauchit_ftp_RMSE
## [1] 0.1073525
##
## $modelo_betat_cauchit_ftp_MAE
## [1] 0.08891667
##
## $gamlss_betap_pf_R2
## [1] 0.7467621
##
## $gamlss_betap_pf_RMSE
## [1] 0.9359013
##
## $gamlss_betap_pf_MAE
## [1] 0.7781778
##
## $misto_normalp_dreb_R2
## [1] 0.7761119
##
## $misto_normalp_dreb_RMSE
## [1] 0.09672168
##
## $misto_normalp_dreb_MAE
## [1] 0.07475841
##
## $misto_normalp_temp_team_R2
## [1] 0.8085238
##
## $misto_normalp_temp_team_RMSE
## [1] 0.08953102
##
## $misto_normalp_temp_team_MAE
## [1] 0.06884756
##
## $misto_betap_ftp_R2
## [1] 0.7609545
##
## $misto_betap_ftp_RMSE
## [1] 0.9362272
##
## $misto_betap_ftp_MAE
## [1] 0.7770889
##
## $misto_betap_temp_R2
## [1] 0.7467621
##
## $misto_betap_temp_RMSE
## [1] 0.9359075
##
## $misto_betap_temp_MAE
## [1] 0.7781845
```

```
data.frame(Modelo = c("linear forward","betareg logito",
                      "betareg loglog","betareg probito",
                      "betareg cloglog","Betareg Cauchit","Gamlss Beta Forward",
                      "Misto Normal TEAM","Misto Normal TEMPORADA","Misto Beta TEAM",
                      "Misto Beta TEMPORADA"),
           R2 = c(0.7472946, 0.7631798,  0.7611173, 0.765559, 0.7513722, 0.7417763,
                  0.7467621, 0.7761119, 0.8085238, 0.7609545,0.7467621),
           RMSE = c(0.1026929, 0.1036046, 0.1022045, 0.1025, 0.1065006,
                    0.1073525, 0.9359013,0.09672168,  0.08953102, 0.9362272,
                    0.9359075),
           MAE = c(0.07949132, 0.08530677, 0.08324645, 0.08403919,
                   0.08796576, 0.08891667, 0.7781778, 0.07475841, 0.06884756,
                   0.7770889,  0.7781845)
)
```

```
##                     Modelo        R2       RMSE        MAE
## 1          linear forward 0.7472946 0.10269290 0.07949132
## 2          betareg logito 0.7631798 0.10360460 0.08530677
## 3          betareg loglog 0.7611173 0.10220450 0.08324645
## 4         betareg probito 0.7655590 0.10250000 0.08403919
## 5         betareg cloglog 0.7513722 0.10650060 0.08796576
## 6         Betareg Cauchit 0.7417763 0.10735250 0.08891667
## 7     Gamlss Beta Forward 0.7467621 0.93590130 0.77817780
## 8       Misto Normal TEAM 0.7761119 0.09672168 0.07475841
## 9  Misto Normal TEMPORADA 0.8085238 0.08953102 0.06884756
## 10        Misto Beta TEAM 0.7609545 0.93622720 0.77708890
## 11   Misto Beta TEMPORADA 0.7467621 0.93590750 0.77818450
```

```
#Outra forma de Cross Validation, mas não será utilizada
#porque apenas faz oara uma amostra só, precisando desenvolver para mais vezes
#para ser interessante utilizar.

##### Cross Validation com apenas uma amostra #####
# R program to implement validation set approach

# setting seed to generate a reproducible random sampling
set.seed(236292)

# creating training data as 80% of the dataset
random_sample <- createDataPartition(dados_regressaop$WINP, p = 0.8, list = FALSE)

# generating training dataset from the random_sample
training_dataset  <- dados_regressaop[random_sample, ]

# generating testing dataset from rows which are not included in random_sample
testing_dataset <- dados_regressaop[-random_sample, ]

# Building the model

# training the model by assigning sales column as target variable and rest other columns
# as independent variables
modelo_forwp <- lm(formula = WINP ~ PlusMinus + DREB, data = training_dataset)
modelo_betapt_ftp <- betareg(formula = WINP_transformado ~ FTP + REB + PlusMinus, data = training_datas
modelop_loglog_reb <- betareg(formula = WINP_transformado ~ REB + PlusMinus, data = training_dataset, l
modelop_probit_ftp <- betareg(formula = WINP_transformado ~ FTP + REB + PlusMinus, data = training_datas
modelo_betat_cloglog_ftp <- betareg(formula = WINP_transformado ~ FTP + REB + PlusMinus, data = trainin
modelo_betat_cauchit_ftp <- betareg(formula = WINP_transformado ~ FTP + PlusMinus, data = training_datas
```

```r
gamlss_betap_pf <- gamlss(formula = WINP ~ PF + PlusMinus, family = BEZI, data = training_dataset)
```

```
## GAMLSS-RS iteration 1: Global Deviance = -122.2003
## GAMLSS-RS iteration 2: Global Deviance = -245.9235
## GAMLSS-RS iteration 3: Global Deviance = -246.8765
## GAMLSS-RS iteration 4: Global Deviance = -246.8767
```

```r
misto_normalp_dreb <- gamlss(formula = WINP ~ (re(random = ~1 | TEAM)) + PlusMinus + DREB, family = NO,
```

```
## GAMLSS-RS iteration 1: Global Deviance = -341.1061
## GAMLSS-RS iteration 2: Global Deviance = -341.1062
```

```r
misto_normalp_temp_team <- gamlss(formula = WINP ~ (re(random = ~1 | Numero_temporada)) + PlusMinus + D
```

```
## GAMLSS-RS iteration 1: Global Deviance = -381.6263
## GAMLSS-RS iteration 2: Global Deviance = -381.6263
```

```r
misto_betap_ftp <- gamlss(formula = WINP ~ (re(random = ~1 | TEAM)) + PlusMinus + PF + BLKA + FTP, famil
```

```
## GAMLSS-RS iteration 1: Global Deviance = -127.9024
## GAMLSS-RS iteration 2: Global Deviance = -262.9788
## GAMLSS-RS iteration 3: Global Deviance = -263.6958
## GAMLSS-RS iteration 4: Global Deviance = -263.6632
## GAMLSS-RS iteration 5: Global Deviance = -263.6602
## GAMLSS-RS iteration 6: Global Deviance = -263.6599
```

```r
misto_betap_temp <- gamlss(formula = WINP ~ (re(random = ~1 | Numero_temporada)) + PlusMinus + PF, famil
```

```
## GAMLSS-RS iteration 1: Global Deviance = -122.2003
## GAMLSS-RS iteration 2: Global Deviance = -245.9235
## GAMLSS-RS iteration 3: Global Deviance = -246.8765
## GAMLSS-RS iteration 4: Global Deviance = -246.8767
```

```r
# predicting the target variable
predictions1 <- predict(modelo_forwp , newdata = testing_dataset)
predictions2 <- predict(modelo_betapt_ftp, newdata = testing_dataset)
predictions3 <- predict(modelop_loglog_reb, newdata = testing_dataset)
predictions4 <- predict(modelop_probit_ftp, newdata = testing_dataset)
predictions5 <- predict(modelo_betat_cloglog_ftp, newdata = testing_dataset)
predictions6 <- predict(modelo_betat_cauchit_ftp, newdata = testing_dataset)
predictions7 <- predict(gamlss_betap_pf, newdata = testing_dataset)
predictions8 <- predict(misto_normalp_dreb, newdata = testing_dataset)
predictions9 <- predict(misto_normalp_temp_team, newdata = testing_dataset)
predictions10 <- predict(misto_betap_ftp, newdata = testing_dataset)
predictions11 <- predict(misto_betap_temp, newdata = testing_dataset)

# computing model performance metrics
data.frame(Modelo = c("linear forward","betareg logito",
                      "betareg loglog","betareg probito",
                      "betareg cloglog","Betareg Cauchit","Gamlss Beta Forward",
                      "Misto Normal TEAM","Misto Normal TEMPORADA","Misto Beta TEAM",
                      "Misto Beta TEMPORADA"),
           R2 = c(R2(predictions1, testing_dataset$WINP), R2(predictions2, testing_dataset$WINP),
                  R2(predictions3, testing_dataset$WINP),R2(predictions4, testing_dataset$WINP),
                  R2(predictions5, testing_dataset$WINP), R2(predictions6, testing_dataset$WINP),
                  R2(predictions7, testing_dataset$WINP), R2(predictions8, testing_dataset$WINP),
                  R2(predictions9, testing_dataset$WINP), R2(predictions10, testing_dataset$WINP),
                  R2(predictions11, testing_dataset$WINP)),
           RMSE = c(RMSE(predictions1, testing_dataset$WINP), RMSE(predictions2, testing_dataset$WINP),
                    RMSE(predictions3, testing_dataset$WINP), RMSE(predictions4, testing_dataset$WINP),
                    RMSE(predictions5, testing_dataset$WINP), RMSE(predictions6, testing_dataset$WINP),
                    RMSE(predictions7, testing_dataset$WINP), RMSE(predictions8, testing_dataset$WINP),
                    RMSE(predictions9, testing_dataset$WINP), RMSE(predictions10, testing_dataset$WINP)
                    RMSE(predictions11, testing_dataset$WINP)),
           MAE = c(MAE(predictions1, testing_dataset$WINP), MAE(predictions2, testing_dataset$WINP),
                   MAE(predictions3, testing_dataset$WINP), MAE(predictions4, testing_dataset$WINP),
                   MAE(predictions5, testing_dataset$WINP), MAE(predictions6, testing_dataset$WINP),
                   MAE(predictions7, testing_dataset$WINP), MAE(predictions8, testing_dataset$WINP),
                   MAE(predictions9, testing_dataset$WINP), MAE(predictions10, testing_dataset$WINP),
                   MAE(predictions11, testing_dataset$WINP)))
```

```
##                     Modelo        R2       RMSE        MAE
## 1          linear forward 0.8204252 0.09020151 0.06927649
## 2          betareg logito 0.7896302 0.10973882 0.08799970
## 3          betareg loglog 0.8143984 0.09961991 0.08090823
## 4         betareg probito 0.7937465 0.10822247 0.08692630
## 5         betareg cloglog 0.7798162 0.11255887 0.09147184
## 6         Betareg Cauchit 0.7730396 0.10920039 0.08824020
## 7     Gamlss Beta Forward 0.8259723 0.94319708 0.80342732
## 8       Misto Normal TEAM 0.8173185 0.09074120 0.06938673
## 9  Misto Normal TEMPORADA 0.6887285 0.11444170 0.08283110
## 10        Misto Beta TEAM 0.8002966 0.96111958 0.82167411
## 11   Misto Beta TEMPORADA 0.8259723 0.94320363 0.80343445
```