Cross Validation Regular

2024-05-03

```r
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
##
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
##
## Attaching package: 'lme4'
##
##
## The following object is masked from 'package:gamlss':
##
##     refit
##
##
## The following object is masked from 'package:nlme':
##
##     lmList
```

```
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
```

```r
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
# 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(
  "regressao_linearp" = lm(formula = WINP ~ PlusMinus + DREB + TEAM, data = dados_regressaop),
  "beta_logitop" = betareg(formula = WINP_transformado ~ TEAM + FTP + REB + PlusMinus, data = dados_regi
  "beta_loglogp" = betareg(formula = WINP_transformado ~ TEAM + REB + PlusMinus, data = dados_regressaop
  "beta_probitp" = betareg(formula = WINP_transformado ~ TEAM + FTP + REB + PlusMinus, data = dados_regi
  "beta_cloglogp" = betareg(formula = WINP_transformado ~ TEAM + FTP + REB + PlusMinus, data = dados_reg
  "beta_cauchitp" = betareg(formula = WINP_transformado ~ TEAM + FTP + PlusMinus, data = dados_regressao
  "gamlss_betap" = gamlss(formula = WINP ~ TEAM + PF + PlusMinus, family = BEZI, data = dados_regressaop
  "misto_normal_teamp" = lmer(formula = WINP ~ (1 | TEAM) + PlusMinus + DREB, data = dados_regressaop),
  "misto_normal_tempp" = lmer(formula = WINP ~ (1 | Numero_temporada) + PlusMinus + DREB + TEAM, data =
  "misto_beta_teamp" = gamlss(formula = WINP ~ (re(random = ~1 | TEAM)) + PlusMinus + PF + BLKA + FTP, r
  "misto_beta_tempp" = gamlss(formula = WINP ~ (re(random = ~1 | Numero_temporada)) + PlusMinus + PF + T
)
```

```
## GAMLSS-RS iteration 1: Global Deviance = -175.3582
## GAMLSS-RS iteration 2: Global Deviance = -370.9928
## GAMLSS-RS iteration 3: Global Deviance = -372.818
## GAMLSS-RS iteration 4: Global Deviance = -372.8183
```

```
## boundary (singular) fit: see help('isSingular')
```

```
## GAMLSS-RS iteration 1: Global Deviance = -167.6882
## GAMLSS-RS iteration 2: Global Deviance = -348.4351
## GAMLSS-RS iteration 3: Global Deviance = -349.3214
## GAMLSS-RS iteration 4: Global Deviance = -349.2541
## GAMLSS-RS iteration 5: Global Deviance = -349.2469
## GAMLSS-RS iteration 6: Global Deviance = -349.2462
## GAMLSS-RS iteration 1: Global Deviance = -175.3582
## GAMLSS-RS iteration 2: Global Deviance = -370.9928
## GAMLSS-RS iteration 3: Global Deviance = -372.8181
## GAMLSS-RS iteration 4: Global Deviance = -372.8183
```

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

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

# Loop para execução da validação cruzada
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_data
      }
    }
  }
}

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

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

```
print(medias_resultados)
```

```
## $regressao_linearp_R2
## [1] 0.8083465
##
## $regressao_linearp_RMSE
## [1] 0.08962513
##
## $regressao_linearp_MAE
## [1] 0.06901616
##
## $beta_logitop_R2
## [1] 0.7896018
##
## $beta_logitop_RMSE
## [1] 0.09709543
##
## $beta_logitop_MAE
## [1] 0.07776968
##
## $beta_loglogp_R2
## [1] 0.7884603
##
## $beta_loglogp_RMSE
## [1] 0.09603766
##
## $beta_loglogp_MAE
## [1] 0.07530452
##
## $beta_probitp_R2
## [1] 0.7903194
##
```

```
## $beta_probitp_RMSE
## [1] 0.09670432
##
## $beta_probitp_MAE
## [1] 0.07691644
##
## $beta_cloglogp_R2
## [1] 0.7781836
##
## $beta_cloglogp_RMSE
## [1] 0.09968129
##
## $beta_cloglogp_MAE
## [1] 0.08084617
##
## $beta_cauchitp_R2
## [1] 0.775023
##
## $beta_cauchitp_RMSE
## [1] 0.0990659
##
## $beta_cauchitp_MAE
## [1] 0.08025891
##
## $gamlss_betap_R2
## [1] 0.7823425
##
## $gamlss_betap_RMSE
## [1] 0.9303888
##
## $gamlss_betap_MAE
## [1] 0.7775557
##
## $misto_normal_teamp_R2
## [1] 0.7785554
##
## $misto_normal_teamp_RMSE
## [1] 0.09620622
##
## $misto_normal_teamp_MAE
## [1] 0.07431771
##
## $misto_normal_tempp_R2
## [1] 0.8083465
##
## $misto_normal_tempp_RMSE
## [1] 0.08962513
##
## $misto_normal_tempp_MAE
## [1] 0.06901616
##
## $misto_beta_teamp_R2
## [1] 0.7599708
##
```

```
## $misto_beta_teamp_RMSE
## [1] 0.9360577
##
## $misto_beta_teamp_MAE
## [1] 0.7770674
##
## $misto_beta_tempp_R2
## [1] 0.7823425
##
## $misto_beta_tempp_RMSE
## [1] 0.930395
##
## $misto_beta_tempp_MAE
## [1] 0.7775623
```

```
data.frame(Modelo = c("Regressão Linear", "Betareg Logito",
                      "Betareg loglog","Betareg probito",
                      "Betareg cloglog","Betareg Cauchit","Gamlss Beta",
                      "Misto Normal TEAM","Misto Normal TEMPORADA","Misto Beta TEAM",
                      "Misto Beta TEMPORADA"),
           R2 = c(medias_resultados[[1]], medias_resultados[[4]],medias_resultados[[7]],
                  medias_resultados[[10]], medias_resultados[[13]], medias_resultados[[16]],
                  medias_resultados[[19]], medias_resultados[[22]],medias_resultados[[25]],
                  medias_resultados[[28]], medias_resultados[[31]]),
           RMSE = c(medias_resultados[[2]], medias_resultados[[5]],medias_resultados[[8]],
                    medias_resultados[[11]], medias_resultados[[14]], medias_resultados[[17]],
                    medias_resultados[[20]], medias_resultados[[23]],medias_resultados[[26]],
                    medias_resultados[[29]], medias_resultados[[32]]),
           MAE = c(medias_resultados[[3]], medias_resultados[[6]],medias_resultados[[9]],
                   medias_resultados[[12]], medias_resultados[[15]], medias_resultados[[18]],
                   medias_resultados[[21]], medias_resultados[[24]],medias_resultados[[27]],
                   medias_resultados[[30]], medias_resultados[[33]])
)
```

```
##                    Modelo        R2       RMSE        MAE
## 1        Regressão Linear 0.8083465 0.08962513 0.06901616
## 2          Betareg Logito 0.7896018 0.09709543 0.07776968
## 3          Betareg loglog 0.7884603 0.09603766 0.07530452
## 4         Betareg probito 0.7903194 0.09670432 0.07691644
## 5         Betareg cloglog 0.7781836 0.09968129 0.08084617
## 6         Betareg Cauchit 0.7750230 0.09906590 0.08025891
## 7             Gamlss Beta 0.7823425 0.93038880 0.77755566
## 8       Misto Normal TEAM 0.7785554 0.09620622 0.07431771
## 9  Misto Normal TEMPORADA 0.8083465 0.08962513 0.06901616
## 10        Misto Beta TEAM 0.7599708 0.93605766 0.77706745
## 11   Misto Beta TEMPORADA 0.7823425 0.93039505 0.77756229
```

```
#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
```

```r
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
"regressao_linearp" = lm(formula = WINP ~ PlusMinus + DREB + TEAM, data = training_dataset)
"beta_logitop" = betareg(formula = WINP_transformado ~ TEAM + FTP + REB + PlusMinus, data = training_da
"beta_loglogp" = betareg(formula = WINP_transformado ~ TEAM + REB + PlusMinus, data = training_dataset,
"beta_probitp" = betareg(formula = WINP_transformado ~ TEAM + FTP + REB + PlusMinus, data = training_da
"beta_cloglogp" = betareg(formula = WINP_transformado ~ TEAM + FTP + REB + PlusMinus, data = training_d
"beta_cauchitp" = betareg(formula = WINP_transformado ~ TEAM + FTP + PlusMinus, data = training_dataset
"gamlss_betap" = gamlss(formula = WINP ~ TEAM + PF + PlusMinus, family = BEZI, data = training_dataset)
```

```
## GAMLSS-RS iteration 1: Global Deviance = -136.8669
## GAMLSS-RS iteration 2: Global Deviance = -288.5186
## GAMLSS-RS iteration 3: Global Deviance = -289.8117
## GAMLSS-RS iteration 4: Global Deviance = -289.8119
```

```r
"misto_normal_teamp" = lmer(formula = WINP ~ (1 | TEAM) + PlusMinus + DREB, data = training_dataset)
"misto_normal_tempp" = lmer(formula = WINP ~ (1 | Numero_temporada) + PlusMinus + DREB + TEAM, data = t
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
"misto_beta_teamp" = gamlss(formula = WINP ~ (re(random = ~1 | TEAM)) + PlusMinus + PF + BLKA + FTP, fa
```

```
## GAMLSS-RS iteration 1: Global Deviance = -127.6279
## GAMLSS-RS iteration 2: Global Deviance = -261.2496
## GAMLSS-RS iteration 3: Global Deviance = -261.6532
## GAMLSS-RS iteration 4: Global Deviance = -261.5991
## GAMLSS-RS iteration 5: Global Deviance = -261.5936
## GAMLSS-RS iteration 6: Global Deviance = -261.5931
```

```r
"misto_beta_tempp" = gamlss(formula = WINP ~ (re(random = ~1 | Numero_temporada)) + PlusMinus + PF + TE
```

```
## GAMLSS-RS iteration 1: Global Deviance = -136.8669
## GAMLSS-RS iteration 2: Global Deviance = -288.5186
## GAMLSS-RS iteration 3: Global Deviance = -289.8117
## GAMLSS-RS iteration 4: Global Deviance = -289.8119
```

```r
# predicting the target variable
predictions1 <- predict(regressao_linearp, newdata = testing_dataset)
predictions3 <- predict(beta_logitop, newdata = testing_dataset)
predictions4 <- predict(beta_loglogp, newdata = testing_dataset)
predictions5 <- predict(beta_probitp, newdata = testing_dataset)
predictions6 <- predict(beta_cloglogp, newdata = testing_dataset)
predictions7 <- predict(beta_cauchitp, newdata = testing_dataset)
```

```r
predictions8 <- predict(gamlss_betap, newdata = testing_dataset)
predictions9 <- predict(misto_normal_teamp, newdata = testing_dataset)
predictions10 <- predict(misto_normal_tempp, newdata = testing_dataset)
predictions11 <- predict(misto_beta_teamp, newdata = testing_dataset)
predictions12 <- predict(misto_beta_tempp, newdata = testing_dataset)

# computing model performance metrics
data.frame(Modelo = c("Regressão linear","betareg logito",
                      "betareg loglog","betareg probito",
                      "betareg cloglog","Betareg Cauchit","Gamlss Beta",
                      "Misto Normal TEAM","Misto Normal TEMPORADA","Misto Beta TEAM",
                      "Misto Beta TEMPORADA"),
           R2 = c(R2(predictions1, 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), R2(predictions12, testing_dataset$WINP)),
           RMSE = c(RMSE(predictions1, testing_dataset$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$WINP), RMSE(predictions12, testing_dataset$WINP
           MAE = c(MAE(predictions1, 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), MAE(predictions12, testing_dataset$WINP)))
```

```
##                    Modelo        R2       RMSE        MAE
## 1        Regressão linear 0.6887048 0.11454052 0.08282424
## 2          betareg logito 0.6771863 0.12593457 0.09555294
## 3          betareg loglog 0.6573441 0.12476109 0.09464211
## 4         betareg probito 0.6691903 0.12580441 0.09575071
## 5         betareg cloglog 0.6671769 0.12801645 0.09718686
## 6         Betareg Cauchit 0.6893367 0.12522576 0.09019257
## 7             Gamlss Beta 0.5522583 0.95317668 0.79756146
## 8       Misto Normal TEAM 0.8169375 0.09090035 0.06941593
## 9  Misto Normal TEMPORADA 0.6887048 0.11454052 0.08282424
## 10        Misto Beta TEAM 0.8003264 0.96197836 0.82269193
## 11   Misto Beta TEMPORADA 0.5522583 0.95318118 0.79756570
```