UNIVERSIDADE FEDERAL DO PARANÁ
Especialização em Inteligência Artificial Aplicada
IAA004 - Linguagem R
Prof. Dr. Razer A N R Montaño

SAMUEL KOJICOVSKI
WELLINTHON DA SILVEIRA KIILLER

LINGUAGEM R – MODELOS DE CLASSIFICAÇÃO E REGRESSÃO LINEAR

Trabalho apresentado como requisito para aprovação na disciplina de Linguagem R, ministrada pelo professor Razer Anthom Nizer Rojas Montano, na especialização em Inteligência Artificial Aplicada da Universidade Federal do Paraná - UFPR.

CURITIBA
2021

ÍNDICE DE ILUSTRAÇÕES

TRABALHO DA DISCIPLINA

Este trabalho pode ser realizado em equipes de no máximo 5 integrantes

O que deve ser entregue:

- Um arquivo compactado com os documentos e arquivos
- A lista de comandos R que foi executada, com suas respectivas saídas
- Um texto com o resultado e justificativa do porquê
- Outros arquivos pedidos (ex, modelo gerado)

1 Pesquisa com Dados de Satélite (Satellite)

O banco de dados consiste nos valores multi-espectrais de pixels em vizinhanças 3x3 em uma imagem de satélite, e na classificação associada ao pixel central em cada vizinhança. O objetivo é prever esta classificação, dados os valores multi-espectrais.

Um quadro de imagens do Satélite Landsat com MSS (Multispectral Scanner System) consiste em quatro imagens digitais da mesma cena em diferentes bandas espectrais. Duas delas estão na região visível (correspondendo aproximadamente às regiões verde e vermelha do espectro visível) e duas no infravermelho (próximo). Cada pixel é uma palavra binária de 8 bits, com 0 correspondendo a preto e 255 a branco. A resolução espacial de um pixel é de cerca de 80m x 80m. Cada imagem contém 2340 x 3380 desses pixels. O banco de dados é uma subárea (minúscula) de uma cena, consistindo de 82 x 100 pixels. Cada linha de dados corresponde a uma vizinhança quadrada de pixels 3x3 completamente contida dentro da subárea 82x100. Cada linha contém os valores de pixel nas quatro bandas espectrais (convertidas em ASCII) de cada um dos 9 pixels na vizinhança de 3x3 e um número indicando o rótulo de classificação do pixel central.

As classes são: solo vermelho, colheita de algodão, solo cinza, solo cinza úmido, restolho de vegetação, solo cinza muito úmido.

Os dados estão em ordem aleatória e certas linhas de dados foram removidas, portanto você não pode reconstruir a imagem original desse conjunto de dados. Em cada linha de dados, os quatro valores espectrais para o pixel superior esquerdo são dados primeiro, seguidos pelos quatro valores espectrais para o pixel superior central e, em seguida, para o pixel superior direito, e assim por diante, com os pixels lidos em sequência, da esquerda para a direita e de cima para baixo. Assim, os quatro valores espectrais para o pixel central são dados pelos atributos 17, 18, 19 e 20. Se você quiser, pode usar apenas esses quatro atributos, ignorando os outros. Isso evita o problema que surge quando uma vizinhança 3x3 atravessa um limite.

O banco de dados se encontra no pacote **mlbench** e é completo (não possui dados faltantes).

1.1 TAREFAS

- Treine modelos RandomForest, SVM e RNA para predição destes dados.

```
# Date: 24 May 2021
# Authors: Wellinthon Kiiller and Samuel Kojicovski

# Load libraries
library("caret")
library("mlbench")
library("randomForest")

# Set directory
setwd("D:/Documentos/UFPR/Disciplinas/Linguagem R/4.GIT/iaa-ufpr-applied-r-
language/satellite-data-search/")

# Dataset from Satellite base
data(Satellite)
dataset <- Satellite

# Separe datasets
index <- createDataPartition(dataset$classes, p=0.8, list = FALSE)
train_data <- dataset[index, ]
test_data <- dataset[-index, ]

# Train models
random_forest <- train(classes~., data=train_data, method="rf")
svm <- train(classes~., data=train_data, method="svmRadial")
rna <- train(classes~., data=train_data, method="nnet", trace=FALSE)

# Make predictions
predictions_random_forest <- predict(random_forest, test_data)
predictions_svm <- predict(svm, test_data)
predictions_rna <- predict(rna, test_data)

# Generate confusion matrix
confusionMatrix(predictions_random_forest, test_data$classes)
confusionMatrix(predictions_svm, test_data$classes)
confusionMatrix(predictions_rna, test_data$classes)
```

- Escolha o melhor modelo com base em suas matrizes de confusão.

  Para encontrar o melhor modelo de classificação dos solos, 3 modelos de *machine learning* foram submetidos a testes com bases de dados de treino de 75% e 80%, são eles: Random Forest, Rede Neural e SVM. Dentre os modelos testados, o mais eficaz foi o Random Forest, com uma acurácia de 90,41% para uma base de treino de 75% e 91,56% para uma base de treino de com 80% dos dados.

```
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil                373           0         1              2                   8                   0
  cotton crop               2         170         0              0                   2                   0
  grey soil                 8           1       322             39                   1                   6
  damp grey soil            0           2        11             93                   0                  24
  vegetation stubble        0           0         0              2                 156                   9
  very damp grey soil       0           2         5             20                   9                 338

Overall Statistics

               Accuracy : 0.9041
                 95% CI : (0.8887, 0.9181)
    No Information Rate : 0.2385
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8813
```

Figura 1 - Matriz de confusão do modelo Random Forest (treino = 75% e teste = 25%)

```
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil                374           0         1              2                   4                   0
  cotton crop               1         171         1              1                   3                   0
  grey soil                 8           0       323             41                   1                   8
  damp grey soil            0           3        12             88                   0                  32
  vegetation stubble        0           0         0              3                 159                  12
  very damp grey soil       0           1         2             21                   9                 325

Overall Statistics

               Accuracy : 0.8966
                 95% CI : (0.8807, 0.9111)
    No Information Rate : 0.2385
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8722
```

Figura 2 - Matriz de confusão do modelo SVM (treino = 75% e teste = 25%)

```
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil                364           6         2              1                   3                   0
  cotton crop               0           0         0              0                   0                   0
  grey soil                19         156       302             89                  46                  74
  damp grey soil            0           0         0              0                   0                   0
  vegetation stubble        0           0         0              0                   0                   0
  very damp grey soil       0          13        35             66                 127                 303

Overall Statistics

               Accuracy : 0.6034
                 95% CI : (0.579, 0.6274)
    No Information Rate : 0.2385
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4879
```

Figura 3 - Matriz de confusão do modelo RNA (treino = 75% e teste = 25%)

```
> # Generate confusion matrix
> confusionMatrix(predictions_random_forest, test_data$classes)
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil                302           0         3              1                 4                   0
  cotton crop               0         136         0              1                 0                   0
  grey soil                 1           1       259             27                 0                   7
  damp grey soil            0           0         4             77                 0                  11
  vegetation stubble        3           1         1              1               127                   8
  very damp grey soil       0           2         4             18                10                 275

Overall Statistics

               Accuracy : 0.9159
                 95% CI : (0.8993, 0.9305)
    No Information Rate : 0.2383
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8958

 Mcnemar's Test P-Value : NA
```

Figura 4 - Matriz de confusão do modelo Random Forest (treino = 80% e teste = 20%)

```
> confusionMatrix(predictions_svm, test_data$classes)
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil                303           0         3              0                 3                   0
  cotton crop               1         137         1              1                 1                   1
  grey soil                 1           0       259             29                 0                   8
  damp grey soil            0           0         5             76                 0                  18
  vegetation stubble        1           2         0              1               128                  10
  very damp grey soil       0           1         3             18                 9                 264

Overall Statistics

               Accuracy : 0.9089
                 95% CI : (0.8918, 0.9241)
    No Information Rate : 0.2383
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8873

 Mcnemar's Test P-Value : NA
```

Figura 5 - Matriz de confusão do modelo SVM (treino = 80% e teste = 20%)

```
> confusionMatrix(predictions_rna, test_data$classes)
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil                302           9       271            124                115                 301
  cotton crop               4         131         0              1                 23                   0
  grey soil                 0           0         0              0                  0                   0
  damp grey soil            0           0         0              0                  0                   0
  vegetation stubble        0           0         0              0                  3                   0
  very damp grey soil       0           0         0              0                  0                   0

Overall Statistics

               Accuracy : 0.3396
                 95% CI : (0.3137, 0.3662)
    No Information Rate : 0.2383
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1511

 Mcnemar's Test P-Value : NA
```

Figura 6 - Matriz de confusão do modelo RNA (treino = 80% e teste = 20%)

- Treine o modelo final com todos os dados e faça a predição na base completa.

```
# Train the best model according
best_model <- randomForest(classes~., data=dataset, type="regression",
importance=TRUE, mtry=19)
final_predict_random_forest <- predict(best_model, dataset)
confusionMatrix(final_predict_random_forest, dataset$classes)
```

```
Confusion Matrix and Statistics

                    Reference
Prediction           red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil               1533           0         0              0                 0                    0
  cotton crop               0         703         0              0                 0                    0
  grey soil                 0           0      1358              0                 0                    0
  damp grey soil            0           0         0            626                 0                    0
  vegetation stubble        0           0         0              0               707                    0
  very damp grey soil       0           0         0              0                 0                 1508

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9994, 1)
    No Information Rate : 0.2382
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA
```

Figura 7 - Matriz de confusão do modelo Random Forest (treino = 100% e teste = 100%)

- Analise o resultado.

  O modelo final utilizado foi o *Random Forest*, por sua maior acurácia de 91,79%. Após o treino do modelo final com toda o *dataset,* obteu-se uma acurácia de 100%, que levou a um *overfitting* do modelo. Embora tenha uma ótima acurácia com os dados de treino, a classificação com dados desconhecidos pode conter erros, sendo incapaz de prever novos dados.

- Salve este modelo final

```
# Save model
saveRDS(best_model, "sattellite_random_forest.rds")
```

2 Estimativa de Volumes de Árvores

Modelos de aprendizado de máquina são bastante usados na área da engenharia florestal (mensuração florestal) para, por exemplo, estimar o volume de madeira de árvores sem ser necessário abatê-las.

O processo é feito pela coleta de dados (dados observados) através do abate de algumas árvores, onde sua altura, diâmetro na altura do peito (dap), etc, são medidos de forma exata. Com estes dados, treina-se um modelo de AM que pode estimar o volume de outras árvores da população.

Os modelos, chamados **alométricos**, são usados na área há muitos anos e são baseados em regressão (linear ou não) para encontrar uma equação que descreve os dados. Por exemplo, o modelo de Spurr é dado por:

$$Volume = b_0 + b_1 * dap^2 * H_t$$

Onde **dap** é o diâmetro na altura do peito (1,3metros), $\mathbf{H_t}$ é a altura total. Tem-se vários modelos alométricos, cada um com uma determinada característica, parâmetros etc. Um modelo de regressão envolve aplicar os dados observados e encontrar b_0 e b_1 no modelo apresentado, gerando assim uma equação que pode ser usada para prever o volume de outras árvores.

Dado o arquivo **Volumes.csv**, que contém os dados de observação, escolha um modelo de aprendizado de máquina com a melhor estimativa, a partir da estatística de correlação.

2.1 TAREFAS

- Carregar o arquivo **Volumes.csv (http://www.razer.net.br/datasets/Volumes.csv)**

```
# Date: 25 May 2021
# Authors: Wellinthon Kiiller and Samuel Kojicovski

# Load libraries
library("caret")
library("mlbench")
library("randomForest")
library("neuralnet")

# Set directory
setwd("D:/Documentos/UFPR/Disciplinas/Linguagem R/4.GIT/iaa-ufpr-applied-r-language/tree-volumes/")

# Read dataset
dataset <- read.csv2("volumes.csv")
```

- Eliminar a coluna NR, que só apresenta um número sequencial

```
# Removed unused columns
dataset$NR <- NULL
```

- Criar partição de dados: treinamento 80%, teste 20%

```
# Separe datasets
index <- createDataPartition(dataset$VOL, p=0.8, list = FALSE)
train_data <- dataset[index, ]
test_data <- dataset[-index, ]
```

- Usando o pacote "caret", treinar os modelos: Random Forest (rf), SVM (svmRadial), Redes Neurais (neuralnet) e o modelo alométrico de SPURR

```
# Train models
set.seed(7)
start.rf <- Sys.time()
rf <- train(VOL~., data=train_data, method="rf", linout=TRUE)
end.rf <- Sys.time()

start.svm <- Sys.time()
svm <- train(VOL~., data=train_data, method="svmRadial", linout=TRUE)
end.svm <- Sys.time()

start.nnet <- Sys.time()
nnet <- train(VOL~., data=train_data, method="neuralnet")
end.nnet <- Sys.time()

start.allom <- Sys.time()
allometric <- nls(VOL ~ b0 + b1*DAP*DAP*HT, train_data, start=list(b0=0.5,
b1=0.5))
end.allom <- Sys.time()
```

- O modelo alométrico é dado por: Volume = $b_0 + b_1 * dap^2 * H_t$

```
alom <- nls(VOL ~ b0 + b1*DAP*DAP*HT, dados, start=list(b0=0.5,
b1=0.5))
```

- Efetue as predições nos dados de teste

```
predictions.rf <- predict(rf, test_data)
predictions.svm <- predict(svm, test_data)
predictions.nnet <- predict(nnet, test_data)
predictions.allometric <- predict(allometric, test_data)
```

- Crie funções e calcule as seguintes métricas entre a predição e os dados observados
  - Coeficiente de determinação: R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \underline{y})^2}$$

onde y_i é o valor observado, $\hat{y}_i$ é o valor predito e $\underline{y}$ é a média dos valores y_i observados. Quanto mais perto de 1 melhor é o modelo;

```
determination_coefficient <- function (observed, predicted)
{
  num = sum((observed - predicted) ^ 2)
  den = sum((observed - mean(observed)) ^ 2)

  return (1 - (num/den))
}
```

  - Erro padrão da estimativa: S_{yx}

$$S_{yx} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n-2}}$$

esta métrica indica erro, portanto quanto mais perto de 0 melhor é o modelo;

- $S_{yx}\%$

$$S_{yx}\% = \frac{S_{yx}}{\underline{y}} * 100$$

esta métrica indica porcentagem de erro, portanto quanto mais perto de 0 melhor é o modelo;

```
standard_error <- function (observed, predicted)
{
  num = sum((observed - predicted) ^ 2)
  den = (length(observed)) - 2

  return (sqrt(num/den))
}
```

- Escolha o melhor modelo

```
# Estimate metrics
# Random Forest
r2.rf <- determination_coefficient(test_data$VOL, predictions.rf)
syx.rf <- standard_error(test_data$VOL, predictions.rf)

# SVM
r2.svm <- determination_coefficient(test_data$VOL, predictions.svm)
syx.svm <- standard_error(test_data$VOL, predictions.svm)

# Neuralnet
r2.nnet <- determination_coefficient(test_data$VOL, predictions.nnet)
syx.nnet <- standard_error(test_data$VOL, predictions.nnet)

# Allometric
r2.allometric <- determination_coefficient(test_data$VOL,
predictions.allometric)
syx.allometric <- standard_error(test_data$VOL, predictions.allometric)

# Prints
cat(paste("Method |   R   |  Syx",
    "\nRF     |", round(rf.rf, 5), "|", round(syx.rf, 5),
    "\nSVM    |", round(r2.svm, 5), "|", round(syx.svm, 5),
    "\nNNET   |", round(r2.nnet, 5), "|", round(syx.nnet, 5),
    "\nALOM   |", round(r2.allometric, 5), "|", round(syx.allometric, 5),
sep=""))
```

Method	R^2	Syx
RF	0.92783	0.11197
SVM	0.77078	0.19955
NNET	0.87506	0.14732
ALOM	0.89599	0.13442

Figura 8 - Métricas para os dados observados e preditos para cada um dos modelos

Escrever texto de justificativa aqui

Para encontrar o melhor modelo de estimativa de volume de árvores, 4 modelos de *machine learning* foram submetidos a testes com bases de dados de treino de 80%, são eles: Random Forest, Rede Neural, SVM e Alométrico. Dentre os modelos testados, o mais eficaz foi o Random Forest, com um coeficiente de determinação de 92,78% e menor erro padrão de 11,19%, mostrando que os dados estão próximos da linha de regressão gerada pelo modelo.

```
# Train the best model according
best_model <- randomForest(VOL~., data=dataset, type="regression",
importance=TRUE)
final_predict.rf <- predict(best_model, dataset)

# Get the R and Syx
r2.final <- determination_coefficient(dataset$VOL, final_predict.rf)
```

```
syx.final <- standard_error(dataset$VOL, final_predict.rf)

# Prints
cat(paste("     Random Forest \n |    R    |  Syx    |\n",
          "|", round(r2.final, 4), "|", round(syx.final, 4), "|" ))
```

Random Forest	
R^2	Syx
0.9418	0.0992

Figura 9 - Métricas para toda a base de dados no modelo Random Forest

```
# Save model
saveRDS(best_model, "tree_volumes_random_forest.rds")
```