Pesquisa com Dados de Satélite (Satellite)

```
> # Carregar os pacotes
> library(caret)
> library(mlbench)
>
> # Carregando os dados
> data("Satellite")
>
> # Criando as partições
> set.seed(7)
> indices <- createDataPartition(Satellite$classes, p = 0.8, list = FALSE)
> treino <- Satellite[indices, c(17, 18, 19, 20, 37)]
> teste  <- Satellite[-indices, c(17, 18, 19, 20, 37)]
>
> # Treinar RF, SVM e RNA com a base de Treino
> print("Treinando modelos..")
[1] "Treinando modelos.."
> print("Treinando rf")
[1] "Treinando rf"
> rf <- train(classes ~ ., data = treino, method = "rf")
>
> print("Treinando svm")
[1] "Treinando svm"
> svm <- train(classes ~ ., data = treino, method = "svmRadial")
>
> print("Treinando rna")
[1] "Treinando rna"
> rna <- train(classes ~ ., data = treino, method = "nnet", trace = FALSE)
```

```
>
> # Aplicar modelos treinados na base de Teste
> predict.rf <- predict(rf, teste)
> predict.svm <- predict(svm, teste)
> predict.rna <- predict(rna, teste)
>
> # Criar as matrizes de confusão e comparar os resultados
> print("Matriz de confusão RF")
[1] "Matriz de confusão RF"
> print(confusionMatrix(predict.rf, teste$classes))
Confusion Matrix and Statistics

                    Reference
Prediction          red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil
  red soil             296         1       5          2              8              0
  cotton crop            0       123       0          0              2              1
  grey soil              3         0     238         29              1             10
  damp grey soil         1         0      21         62              2             40
  vegetation stubble     6        10       0          0            120              8
  very damp grey soil    0         6       7         32              8            242

Overall Statistics

               Accuracy : 0.8419
                 95% CI : (0.8208, 0.8614)
    No Information Rate : 0.2383
    P-Value [Acc > NIR] : < 2.2e-16
```

Kappa : 0.8047

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: red soil	Class: cotton crop	Class: grey soil	Class: damp grey soil	Class: vegetation stubble
Sensitivity	0.9673	0.87857	0.8782	0.49600	0.85106
Specificity	0.9836	0.99738	0.9576	0.94478	0.97900
Pos Pred Value	0.9487	0.97619	0.8470	0.49206	0.83333
Neg Pred Value	0.9897	0.98532	0.9671	0.94560	0.98158
Prevalence	0.2383	0.10903	0.2111	0.09735	0.10981
Detection Rate	0.2305	0.09579	0.1854	0.04829	0.09346
Detection Prevalence	0.2430	0.09813	0.2188	0.09813	0.11215
Balanced Accuracy	0.9755	0.93797	0.9179	0.72039	0.91503

	Class: very damp grey soil
Sensitivity	0.8040
Specificity	0.9461
Pos Pred Value	0.8203
Neg Pred Value	0.9403
Prevalence	0.2344
Detection Rate	0.1885
Detection Prevalence	0.2298

Balanced Accuracy 0.8750

>

> print("Matriz de confusão SVM")

[1] "Matriz de confusão SVM"

> print(confusionMatrix(predict.svm, teste$classes))

Confusion Matrix and Statistics

Reference

Prediction red soil cotton crop grey soil damp grey soil vegetation stubble very damp grey soil

red soil	298	1	4	2	7	0
cotton crop	1	120	0	0	4	0
grey soil	4	0	260	29	1	12
damp grey soil	0	1	7	69	2	32
vegetation stubble	3	14	0	2	117	3
very damp grey soil	0	4	0	23	10	254

Overall Statistics

Accuracy : 0.8707

95% CI : (0.8511, 0.8886)

No Information Rate : 0.2383

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.8399

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: red soil	Class: cotton crop	Class: grey soil	Class: damp grey soil	Class: vegetation stubble
Sensitivity	0.9739	0.85714	0.9594	0.55200	0.82979
Specificity	0.9857	0.99563	0.9546	0.96376	0.98075
Pos Pred Value	0.9551	0.96000	0.8497	0.62162	0.84173
Neg Pred Value	0.9918	0.98274	0.9888	0.95226	0.97904
Prevalence	0.2383	0.10903	0.2111	0.09735	0.10981
Detection Rate	0.2321	0.09346	0.2025	0.05374	0.09112
Detection Prevalence	0.2430	0.09735	0.2383	0.08645	0.10826
Balanced Accuracy	0.9798	0.92639	0.9570	0.75788	0.90527

	Class: very damp grey soil
Sensitivity	0.8439
Specificity	0.9624
Pos Pred Value	0.8729
Neg Pred Value	0.9527
Prevalence	0.2344
Detection Rate	0.1978
Detection Prevalence	0.2266
Balanced Accuracy	0.9031

>

> print("Matriz de confusão RNA")

[1] "Matriz de confusão RNA"

> print(confusionMatrix(predict.rna, teste$classes))

Confusion Matrix and Statistics

	Reference					
Prediction	red soil	cotton crop	grey soil	damp grey soil	vegetation stubble	very damp grey soil
red soil	292	2	11	2	12	1
cotton crop	7	124	0	0	13	0
grey soil	3	0	259	67	1	25
damp grey soil	0	0	0	0	0	0
vegetation stubble	3	5	0	0	100	12
very damp grey soil	1	9	1	56	15	263

Overall Statistics

Accuracy : 0.8084

95% CI : (0.7858, 0.8296)

No Information Rate : 0.2383

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7595

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: red soil	Class: cotton crop	Class: grey soil	Class: damp grey soil	Class: vegetation stubble
Sensitivity	0.9542	0.88571	0.9557	0.00000	0.70922

Specificity 0.9714 0.98252 0.9052 1.00000 0.98250

Pos Pred Value 0.9125 0.86111 0.7296 NaN 0.83333

Neg Pred Value 0.9855 0.98596 0.9871 0.90265 0.96478

Prevalence 0.2383 0.10903 0.2111 0.09735 0.10981

Detection Rate 0.2274 0.09657 0.2017 0.00000 0.07788

Detection Prevalence 0.2492 0.11215 0.2765 0.00000 0.09346

Balanced Accuracy 0.9628 0.93412 0.9305 0.50000 0.84586

Class: very damp grey soil

Sensitivity 0.8738

Specificity 0.9166

Pos Pred Value 0.7623

Neg Pred Value 0.9595

Prevalence 0.2344

Detection Rate 0.2048

Detection Prevalence 0.2687

Balanced Accuracy 0.8952

>

>

Estimativa de Volumes de Árvores

library("caret")

> library(RSNNS)

> #Carregabdo a base de Dados

> df <- read.csv("http://www.razer.net.br/datasets/Volumes.csv",sep = ";", dec = ",")

```
>
> #Eliminando a coluna NR
> df$NR<- NULL
>
> #Criando as Partições 80/20
> set.seed(7)
> indices <- indices <- createDataPartition(df$VOL,p=0.80, list=FALSE)
> treino <- df[indices,]
> teste <- df[-indices,]
>
> print("Treinando os modelos...")
[1] "Treinando os modelos..."
> #Treinar um modelo Random Forest
> print("Random Forest")
[1] "Random Forest"
> rf <- caret::train(VOL~., data=treino,method="rf")
note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
print("SVM Radial")
[1] "SVM Radial"
svm <- caret::train(VOL~., data=treino,method="svmRadial")
#Treinar um modelo RNA (NNET)
>print("RNA")
[1] "RNA"
rna <- caret::train(VOL~., data=treino,method="nnet")

#Treinar mmodelo Alométrico alom <- nls(VOL ~ b0 + b1DAPDAP*HT, df,
start=list(b0=0.5, b1=0.5))
#predições predicoes.svm <- predict(svm, teste) predicoes.rf <- predict(rf, teste)
predicoes.rna <- predict(rna, teste) predicoes.alom <- predict(alom, teste)
```

```
coefR2 <- function(y_real, y_pred){
```

- residual <- sum((y_real - y_pred)^2) # Soma dos quadrados dos resíduos
- total <- sum((y_real - mean(y_real))^2) # Soma dos quadrados totais
- r2 <- 1 - (residual / total) # Fórmula do R^2
- return(r2)
- }

```
> coefSYX <- function(y_real,y_pred,percent=F){
+   n <- length(y_real)
+   residual <- sum((y_real - y_pred)^2)
+   syx <- sqrt(residual / (n-2))
+   if(percent){
+     syx <- (syx / mean(y_real)) * 100  # Cálculo do Syx %
+   }
+   return(syx)
+ }
>
> #Validando resultados RF
> result.rf.r2 <- coefR2(teste$VOL,predicoes.rf)
> result.rf.syx <- coefSYX(teste$VOL,predicoes.rf)
> result.rf.syxPerc <- coefSYX(teste$VOL,predicoes.rf,percent = T)
>
> #validando resultados SVM
> result.svm.r2 <- coefR2(teste$VOL,predicoes.svm)
> result.svm.syx <- coefSYX(teste$VOL,predicoes.svm)
> result.svm.syxPerc <- coefSYX(teste$VOL,predicoes.svm,percent = T)
>
> #validando resultados RNA
> result.rna.r2 <- coefR2(teste$VOL,predicoes.rna)
> result.rna.syx <- coefSYX(teste$VOL,predicoes.rna)
> result.rna.syxPerc <- coefSYX(teste$VOL,predicoes.rna,percent = T)
>
> #validando resultados SVM
> result.alom.r2 <- coefR2(teste$VOL,predicoes.alom)
> result.alom.syx <- coefSYX(teste$VOL,predicoes.alom)
> result.alom.syxPerc <- coefSYX(teste$VOL,predicoes.alom,percent = T)
>
> cat("RF-> R2: ", result.rf.r2,"Syx: ", result.rf.syx, "Syx%",result.rf.syxPerc,"\n" )
RF-> R2:  0.8535647 Syx:  0.1445527 Syx% 11.07658
```

```
> cat("SVM-> R2: ", result.svm.r2,"Syx: ", result.svm.syx, "Syx%",result.svm.syxPerc,"\n"
)
SVM-> R2:  0.8324292 Syx:  0.1546331 Syx% 11.84901
> cat("RNA-> R2: ", result.rna.r2,"Syx: ", result.rna.syx, "Syx%",result.rna.syxPerc,"\n" )
RNA-> R2:  -0.7244946 Syx:  0.49606 Syx% 38.01139
> cat("Alometrico-> R2: ", result.alom.r2,"Syx: ", result.alom.syx,
"Syx%",result.alom.syxPerc,"\n" )
Alometrico-> R2:  0.8356895 Syx:  0.1531214 Syx% 11.73318
```