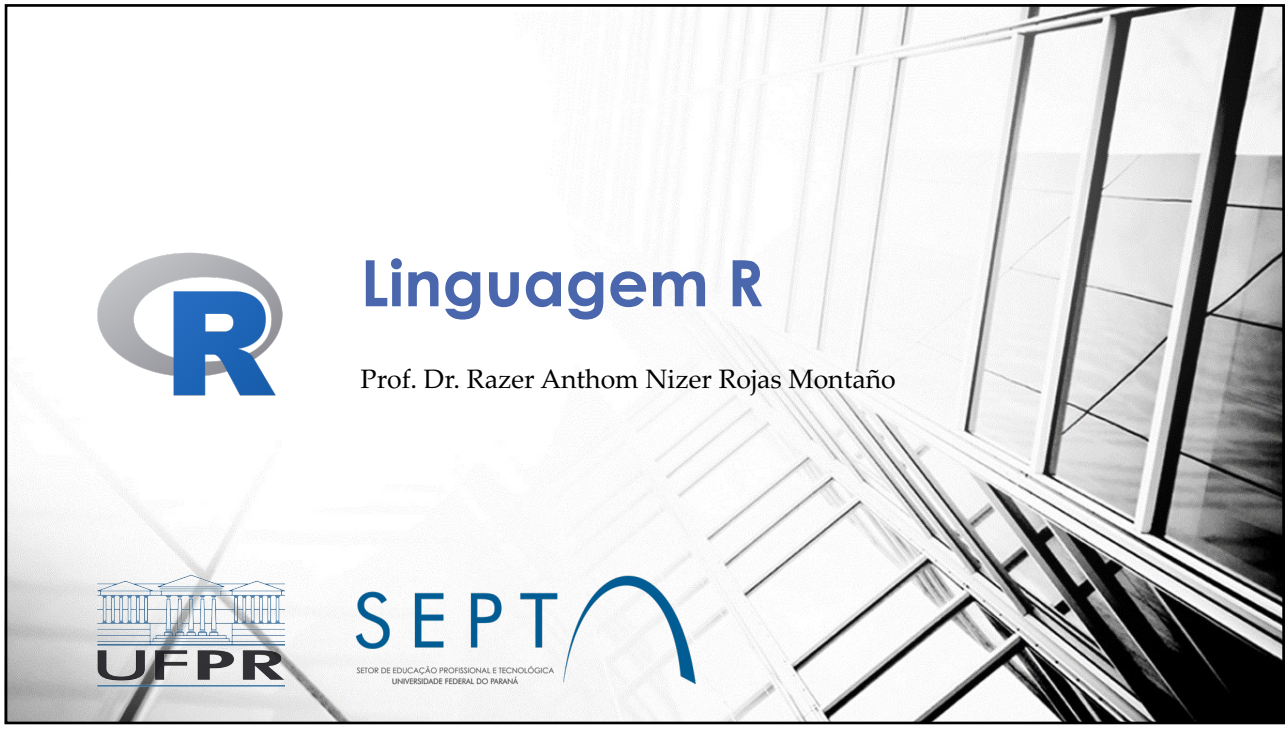
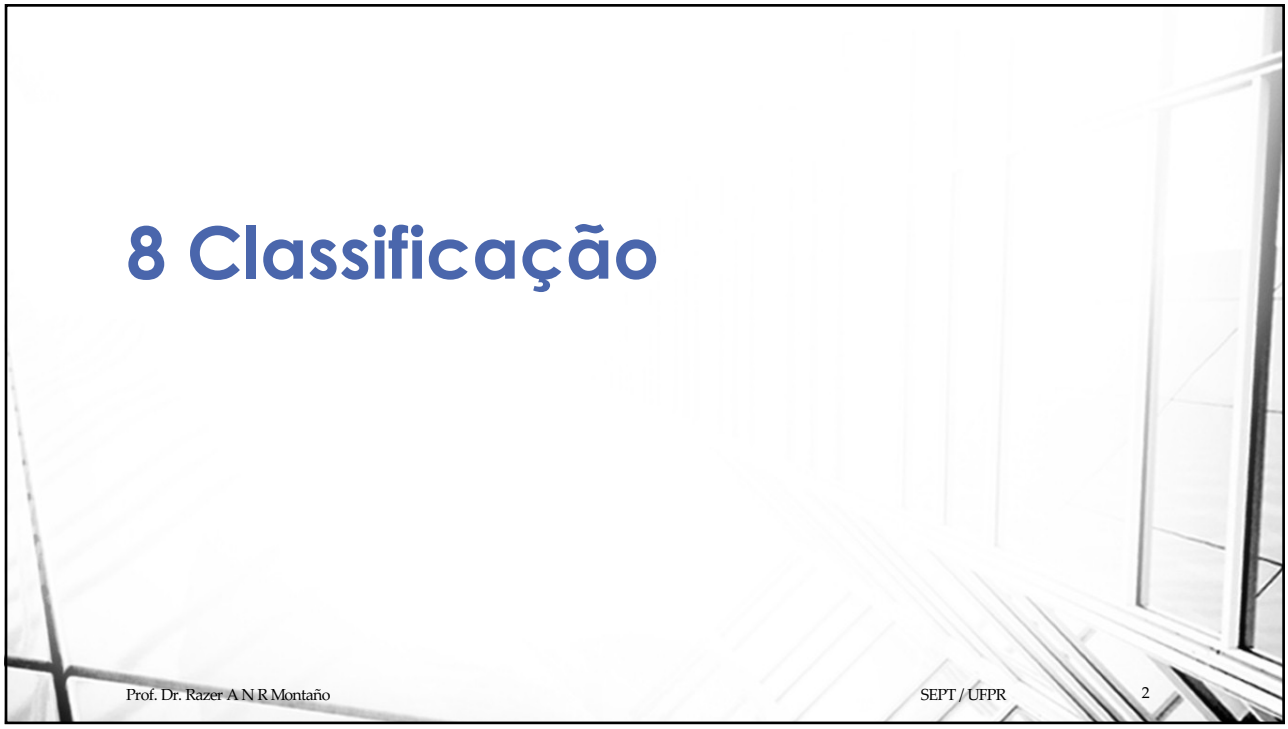

Linguagem R
Prof. Dr. Razer Anthom Nizer Rojas Montaño
UFPR
SEPT
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ

1


8 Classificação
Prof. Dr. Razer A N R Montaño
SEPT / UFPR
2

2

Classificação

- Dada uma base de dados **ROTULADA**
  - Exemplo:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
83	5.8	2.7	3.9	1.2	versicolor
23	4.6	3.6	1.0	0.2	setosa
113	6.8	3.0	5.5	2.1	virginica
123	7.7	2.8	6.7	2.0	virginica
48	4.6	3.2	1.4	0.2	setosa
78	6.7	3.0	5.0	1.7	versicolor

- Quer-se treinar um modelo que **APRENDE** esses rótulos
  - De forma que se um dado desconhecido seja apresentado, ele o classifique
  - Ex:
    - Sepal.Length = 4.2
    - Sepal.Width = 3.2
    - Petal.Length = 1.1
    - Petal.Width = 0.3
  - QUAL É A ESPÉCIE???

3

Classificação

- Dada uma base de dados
  - Aprender para conseguir distinguir entre duas ou mais categorias (classes)
- Ex: IRIS
  - https://archive.ics.uci.edu/ml/datasets/iris
- Dados
  - Comprimento Sépala
  - Largura Sépala
  - Comprimento Pétala
  - Largura Pétala
- Classes
  - Iris Setosa
  - Iris Versicolour
  - Iris Virginica

Iris Versicolor | Iris Setosa | Iris Virginica

4

Classificação

- Base de Dados: iris
- Dados
  - 150 dados
  - 4 atributos : medidas da sépala e pétala
  - 1 atributo de classe

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

- Objetivo: treinar um modelo de classificação
  - Usar 80% da base para treino do modelo
  - Usar 20% da base para teste de acurácia do modelo treinado

5

Classificação

- No R
  - Os dados já estão disponíveis, basta usá-los
  - Usaremos o pacote "caret" para treinar modelos

1. Instalação e uso dos pacotes necessários

```
> install.packages("e1071")
> install.packages("randomForest")
> install.packages("kernlab")
> install.packages("caret")
> library("caret")
```

2. Carga da base de dados

```
> data(iris)
> dataset <- iris
> dataset
```


6

Classificação

3. Particionar a bases em treino (80%) e teste (20%)

```
> indices <- createDataPartition(dataset$Species, p=0.80, list=FALSE)
> treino <- dataset[indices,]
> teste <- dataset[-indices,]
```

4. Treinar um modelo Random Forest com a base de treino

```
> set.seed(7)
> rf <- train(Species~., data=treino, method="rf")
```

5. Efetuar as predições na base de teste

```
> predicoes.rf <- predict(rf, teste)
```


7

Classificação

6. Gerar e analisar a matriz de confusão

```
> confusionMatrix(predicoes.rf, teste$Species)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0          9         1
  virginica       0          1         9

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.7793, 0.9918)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 8.747e-12
```


8

Classificação

7. Gerar um novo modelo usando SVM, predições e matriz de confusão

```
> set.seed(7)
> svm <- train(Species~., data=treino, method="svmRadial")
> predicoes.svm <- predict(svm, teste)
> confusionMatrix(predicoes.svm, teste$Species)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa          9          0         0
  versicolor      0          9         1
  virginica       1          1         9

Overall Statistics

               Accuracy : 0.9
                 95% CI : (0.7347, 0.9789)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.665e-10
```


9

Classificação

8. Comparando as duas matrizes de confusão

Random Forest

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         10          0         0
  versicolor      0          9         1
  virginica       0          1         9

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.7793, 0.9918)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 8.747e-12
```

SVM

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa          9          0         0
  versicolor      0          9         1
  virginica       1          1         9

Overall Statistics

               Accuracy : 0.9
                 95% CI : (0.7347, 0.9789)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.665e-10
```


10

Classificação.

9. Salvamento e Carga dos modelos para uso posterior

```
> getwd()
[1] "/Users/razer"
> save(rf, file="rf.RData")
> save(svm, file="svm.RData")
> load("rf.RData")
```


11

Exercícios.

1. Efetuar o exercício de classificação apresentado, usando a base IRIS.
   a) Apresente os resultados dos modelos
   b) Apresente o modelo que deu o melhor resultado

12

Câncer de Mama

- Wisconsin Breast Cancer Database
  - Dr. William H Wolberg
- Dados
  - 699 amostras
  - Informações sobre as amostras e a classe: benígno / maligno

1. Instalar Pacotes necessários e carregar as bibliotecas

```
> install.packages("caret")
> install.packages("e1071")
> install.packages("mlbench")
> install.packages("mice")
> library(mlbench)
> library(caret)
> library(mice)
```


Câncer de Mama

2. Obter os dados

```
> data(BreastCancer)
> temp_dados <- BreastCancer
```

3. Tratar o ID e *Missing Values*

```
> temp_dados$Id <- NULL
> imp <- mice(temp_dados)
> dados <- complete(imp, 1)
```

Analisa os dados e gera dados plausíveis a serem colocados nos NAs

Completa os dados com o 1º conjunto plausível de dados

4. Criar bases de Treino e Teste

```
> indices <- createDataPartition(dados$Class, p=0.80,
list=FALSE)
> treino <- dados[indices,]
> teste <- dados[-indices,]
```

Câncer de Mama

5. Treinar RF, SVM e RNA com a base de Treino

```
> set.seed(7)
> rf <- train(Class~., data=treino, method="rf")
> svm <- train(Class~., data=treino, method="svmRadial")
> rna <- train(Class~., data=treino, method="nnet",
trace=FALSE)
```

6. Aplicar modelos treinados na base de Teste

```
> predict.rf <- predict(rf, teste)
> predict.svm <- predict(svm, teste)
> predict.rna <- predict(rna, teste)
```


15

Câncer de Mama

7. Verificar a quantidade de amostrar de cada classe na base de Teste

```
> table(teste$Class)
```

8. Criar as matrizes de confusão e comparar os resultados

```
> confusionMatrix(predict.rf, teste$Class)
> confusionMatrix(predict.svm, teste$Class)
> confusionMatrix(predict.rna, teste$Class)
```

Nos interessa (por enquanto) a ACURÁRIA!!!!

16

Câncer de Mama

Random Forest

```
> confusionMatrix(predict.rf, teste$Class)
Confusion Matrix and Statistics

          Reference
Prediction  benign malignant
  benign        87         2
  malignant      4        46

               Accuracy : 0.9568
                 95% CI : (0.9084, 0.984)
    No Information Rate : 0.6547
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9055
 Mcnemar's Test P-Value : 0.6831

            Sensitivity : 0.9560
            Specificity : 0.9583
         Pos Pred Value : 0.9775
         Neg Pred Value : 0.9200
             Prevalence : 0.6547
         Detection Rate : 0.6259
   Detection Prevalence : 0.6403
      Balanced Accuracy : 0.9572

       'Positive' Class : benign
```

SVM

```
> confusionMatrix(predict.svm, teste$Class)
Confusion Matrix and Statistics

          Reference
Prediction  benign malignant
  benign        87         0
  malignant      4        48

               Accuracy : 0.9712
                 95% CI : (0.928, 0.9921)
    No Information Rate : 0.6547
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9376
 Mcnemar's Test P-Value : 0.1336

            Sensitivity : 0.9560
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9231
             Prevalence : 0.6547
         Detection Rate : 0.6259
   Detection Prevalence : 0.6259
      Balanced Accuracy : 0.9780

       'Positive' Class : benign
```

RNA

```
> confusionMatrix(predict.rna, teste$Class)
Confusion Matrix and Statistics

          Reference
Prediction  benign malignant
  benign        87         2
  malignant      4        46

               Accuracy : 0.9568
                 95% CI : (0.9084, 0.984)
    No Information Rate : 0.6547
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9055
 Mcnemar's Test P-Value : 0.6831

            Sensitivity : 0.9560
            Specificity : 0.9583
         Pos Pred Value : 0.9775
         Neg Pred Value : 0.9200
             Prevalence : 0.6547
         Detection Rate : 0.6259
   Detection Prevalence : 0.6403
      Balanced Accuracy : 0.9572

       'Positive' Class : benign
```


17

Câncer de Mama.

- Comparação da Acurácia
  - **RF** : 0,9568
  - **SVM** : 0,9712
  - **RNA** : 0,9568
- Assim, o modelo usando **SVM** tem melhor desempenho, por esta métrica

18

Exercícios.

1. Efetuar o exercício de classificação apresentado, usando a base Câncer de Mama.
   a) Apresente os resultados dos modelos
   b) Apresente o modelo que deu o melhor resultado

19

Salvando e Finalizando o Modelo

Para uso posterior

20

Criar/Salvar/Carregar o Modelo Final

- Uma vez que o modelo foi treinado, testado e escolhido
  - Treina-se um novo modelo com os parâmetros escolhidos
  - Usa-se todos os dados de teste disponíveis
  - Este é o modelo que será disponibilizado
- Então é feito o salvamento do modelo, para posterior uso
  - `saveRDS()`
  - `readRDS()`

21

Criar/Salvar/Carregar o Modelo Final

- A função `train()` do pacote `caret` usa:
  - SMV: função `ksvm()` do pacote `kernlab`
  - RF: função `randomForest()` do pacote `randomForest`
  - RNA: pacote `nnet`, `neuralnet`, etc

22

Criar/Salvar/Carregar o Modelo Final

- No exemplo de Câncer de Mama

```
> print(svm)
Support Vector Machines with Radial Basis Function Kernel

560 samples
  9 predictor
  2 classes: 'benign', 'malignant'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 560, 560, 560, 560, 560, 560, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.9592903  0.9096852
  0.50  0.9640970  0.9198458
  1.00  0.9655193  0.9229535

Tuning parameter 'sigma' was held constant at a value of 0.01173596
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.01173596 and C = 1.
```


Criar/Salvar/Carregar o Modelo Final

- Treina-se um novo modelo, com todos os dados, com os seguintes parâmetros
  - **Sigma = 0.01173596**
  - **C = 1**
- Treina-se o novo modelo completo

```
> final_model <- ksvm(type="C-svc", Class~., data=dados, kernel="rbfdot",
C=1.0, kpar=list(sigma=0.01173596))
> final_predict.svm <- predict(final_model, dados)
> confusionMatrix(final_predict.svm, dados$Class)
```

- Salva-se o modelo treinado

```
> saveRDS(final_model, "cancer_mama_svm.rds")
```

Criar/Salvar/Carregar o Modelo Final.

- Posteriormente pode-se carregar o modelo salvo para novas predições

```
> meu_modelo <- readRDS("./cancer_mama_svm.rds")
> novas_predicoes <- predict(meu_modelo, teste)
> confusionMatrix(novas_predicoes, teste$Class)
```


25

Exercícios.

1. Sobe os exercícios da base Iris e Câncer de Mama
   a) Salve os modelos de gerados
   b) Em um script, carregue os modelos e execute uma predição com cada

26