

Power BI com R para Visualização de Modelos

The background features a dark blue grid. A white line chart with circular markers is positioned horizontally across the middle. Below the line chart, there is a series of vertical bars of varying heights, creating a bar chart effect. The overall aesthetic is clean and technical, suitable for a data science presentation.

Olá!

Mikaele Costa Mendonca

Professor: Methanias Colaço

Disciplina: Mineração de
Dados

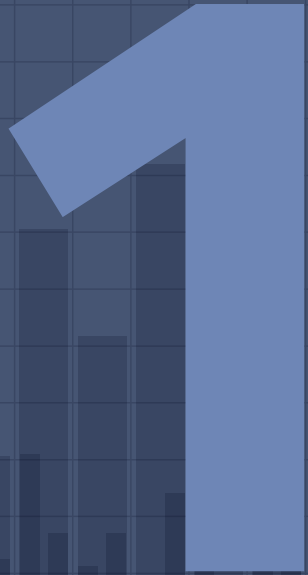



Roteiro

- ▣ Linguagem R
- ▣ Power BI
- ▣ Integração
- ▣ Dataset
- ▣ Pré-processamento
- ▣ Algoritmo
- ▣ Visualização do Modelo


Linguagem R

R é uma linguagem e ambiente para computação estatística e gráficos. É um projeto GNU.





R está disponível como Software Livre sob os termos da Free Software Foundation 's GNU General Public License em forma de código fonte.



Ele compila e roda em uma ampla variedade de plataformas UNIX e sistemas similares (incluindo FreeBSD e Linux), Windows e MacOS.

Ambiente R

R é um conjunto integrado de recursos de software para manipulação de dados, cálculo e exibição gráfica. Inclui:

- ▣ Uma eficaz na manipulação e armazenamento de dados
- ▣ Um conjunto de operadores para cálculos em arrays, em particular matrizes,

Ambiente R

- ▣ Uma larga, coerente e integrada coleção de ferramentas intermediárias para análise de dados,
- ▣ Facilidades gráficas para análise de dados e exibição na tela ou em impressa, e
- ▣ Uma linguagem de programação bem desenvolvida, simples e eficaz que inclui condicionais, loops, funções recursivas definidas pelo usuário e recursos de entrada e saída.

Instalação R e do RStudio

- ▣ <https://cran.r-project.org/bin/windows/Rtools/>
- ▣ Instalar na raiz do disco C://
- ▣ <https://www.rstudio.com/products/rstudio/download/>

Power BI

É um serviço de análise de negócios que fornece insights para permitir decisões rápidas e informadas.



Power BI

- ▣ Transforma dados em visuais impressionantes e compartilhe-os com seus colegas em qualquer dispositivo.
- ▣ Explora e analisa visualmente dados locais ou na nuvem em uma única exibição.
- ▣ Possibilita a colaboração em relatórios interativos e dashboards personalizados e compartilhe-os.
- ▣ Dimensiona em sua organização com governança e segurança internas.

Instalação Power BI

- ▣ <https://powerbi.microsoft.com/pt-br/downloads/>
- ▣ Padrão Windows

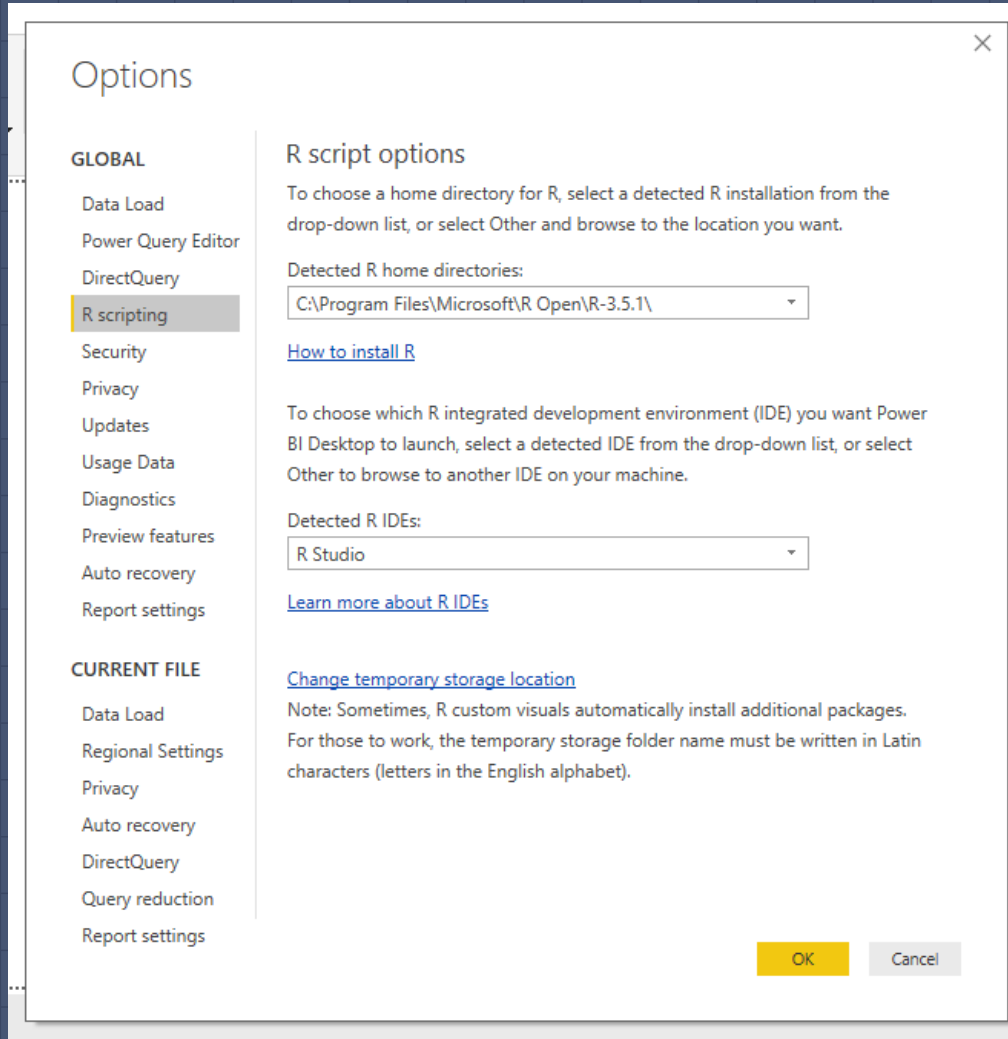
Integração

Integração da Linguagem com o Power BI.



Integração

File > Options > R
scripting



Arquivo Home View Modeling Help Format Data / Drill Sign in

Paste Get Data Recent Sources Enter Data Edit Queries Refresh New Page New Visual Ask A Question Buttons Text box Image Shapes From Marketplace From File Switch Theme Manage Relationships New Measure New Column New Quick Measure Publish

Clipboard External data Insert Custom visuals Themes Relationships Calculations Share

VISUALIZATIONS **FIELDS**

Search

Values

Add data fields here

FILTERS

Visual level filters

Drag data fields here

Page level filters

Drag data fields here

Report level filters

Drag data fields here

DRILLTHROUGH

Keep all filters

Off

R script editor


Drag fields into the Values area in the Visualization pane to start scripting.

Page 1

Dataset

Dados usado para construção do modelo. Dataset: Prevendo a Inadimplência de Clientes





Objetivo: construir um classificador e usá-lo para prever se o cliente do cartão de crédito será inadimplente no próximo pagamento.

Dataset

Os atributos do conjunto de dados são: ID, equilíbrio de crédito, gênero, educação, estado civil e idade. Os Pay_0, Pay_2, Pay_3, Pay_4, Pay_5, Pay_6 são o estado do pagamento nos meses de abril a setembro, respectivamente. O status do pagamento é definido como o atraso no pagamento.



Dataset

Exemplo: se o valor de `Pay_0` for -1, então significa que o cliente foi devidamente pago, se o valor for 2, isso significa que o pagamento está atrasado por dois meses.

`Bill_Amt1` a `Bill_Amt6` são os montantes das faturas do cartão de crédito para o mês de abril até setembro. `Pay_Amt1` a `Pay_Amt6` são o valor que o cliente pagou na conta do cartão de crédito no mês de abril até setembro.

Atributo	Descrição
ID	ID único de cada registro
Credit Balance	Quantidade de crédito no cartão de crédito
Gender	Sexo do cliente (masculino/feminino)
Education	Nível de Escolaridade, I.e. Pos-graduado, Graduado, Ensino Médio, Outros
Marital Status	Estado Civil, I.e. casado, solteiro, outros
Age	Idade do cliente
Pay_0	Status de Pagamento em Setembro
Pay_2	Status de Pagamento em Agosto
Pay_3	Status de Pagamento em Julho
Pay_4	Status de Pagamento em Junho
Pay_5	Status de Pagamento em Maio
Pay_6	Status de Pagamento em Abril

Bill_Amt1	Valor da Conta do cartão em Setembro
Bill_Amt2	Valor da Conta do cartão em Agosto
Bill_Amt3	Valor da Conta do cartão em Julho
Bill_Amt4	Valor da Conta do cartão em Junho
Bill_Amt5	Valor da Conta do cartão em Maio
Bill_Amt6	Valor da Conta do cartão em Abril
Pay_Amt1	Valor pago em Setembro
Pay_Amt2	Valor pago em Agosto
Pay_Amt3	Valor pago em Julho
Pay_Amt4	Valor pago em Junho
Pay_Amt5	Valor pago em Maio
Pay_Amt6	Valor pago em Abril
Default_payment_next_month	Valor 0 ou 1 - 0 significa não-Inadimplência e 1 significa Inadimplência

Pré-processamento

Ajuste nos dados para a utilização dos mesmos na geração do modelo.



Pré-processamento

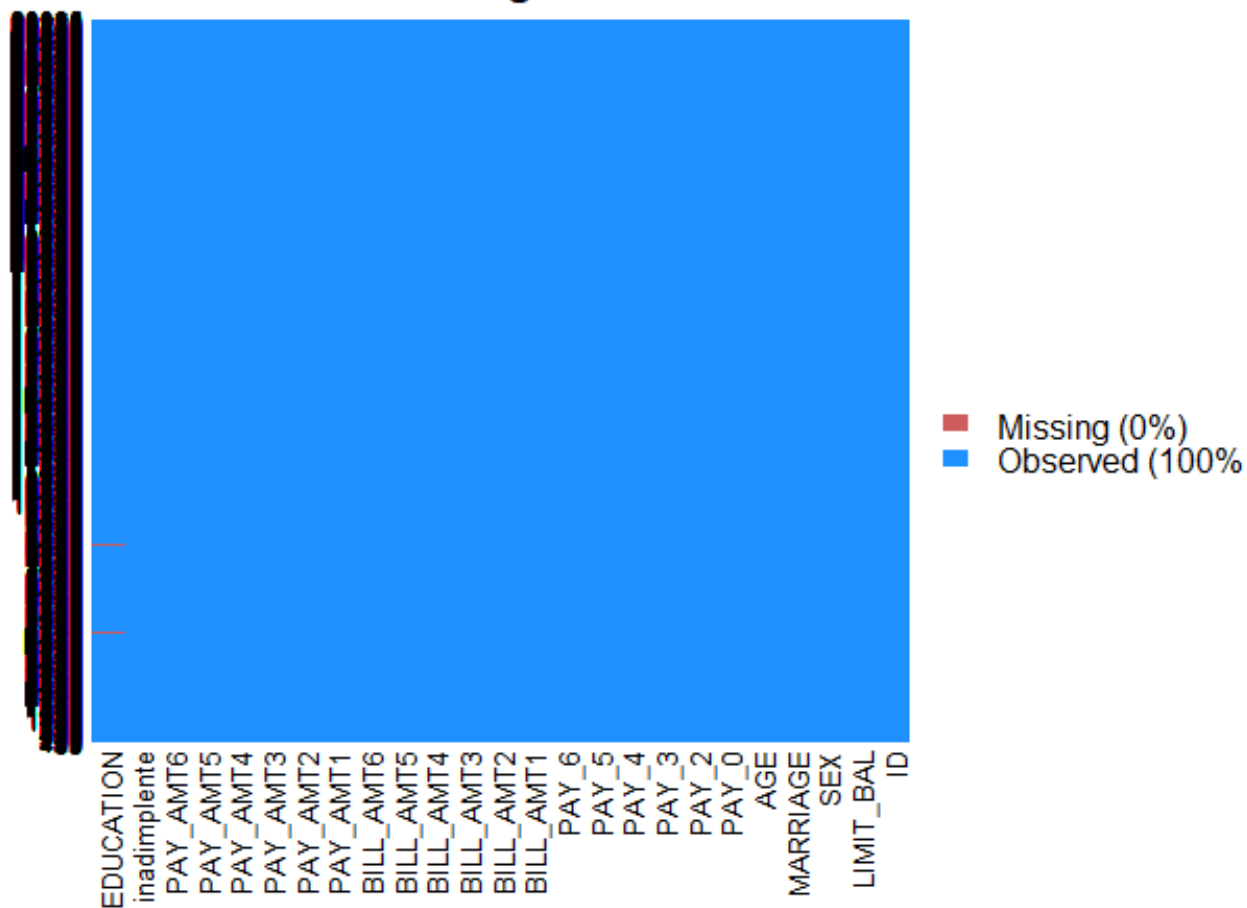
```
47 # Sexo
48 dataset$SEX <- cut(dataset$SEX, c(0,1,2), labels = c("Masculino","Feminino"))
49 head(dataset$SEX)
50
51 # Escolaridade
52 dataset$EDUCATION <- cut(dataset$EDUCATION, c(0,1,2,3,4),
53                               labels = c("Pos Graduated","Graduated","Ensino Medio","Outros"))
54 head(dataset$EDUCATION)
55
56 # Estado Civil
57 dataset$MARRIAGE <- cut(dataset$MARRIAGE, c(-1,0,1,2,3),
58                           labels = c("Desconhecido","Casado","Solteiro","Outros"))
59 head(dataset$MARRIAGE)
60
61 # Convertendo a variavel que indica pagamentos para o tipo fator
62 dataset$PAY_0 <- as.factor(dataset$PAY_0)
63 dataset$PAY_2 <- as.factor(dataset$PAY_2)
64 dataset$PAY_3 <- as.factor(dataset$PAY_3)
65 dataset$PAY_4 <- as.factor(dataset$PAY_4)
66 dataset$PAY_5 <- as.factor(dataset$PAY_5)
67 dataset$PAY_6 <- as.factor(dataset$PAY_6)
```

Valores Vazios

Verificar se há valores não preenchidos no dataset. Para as linhas com valores vazios o tratamento dado foi apenas a remoção.



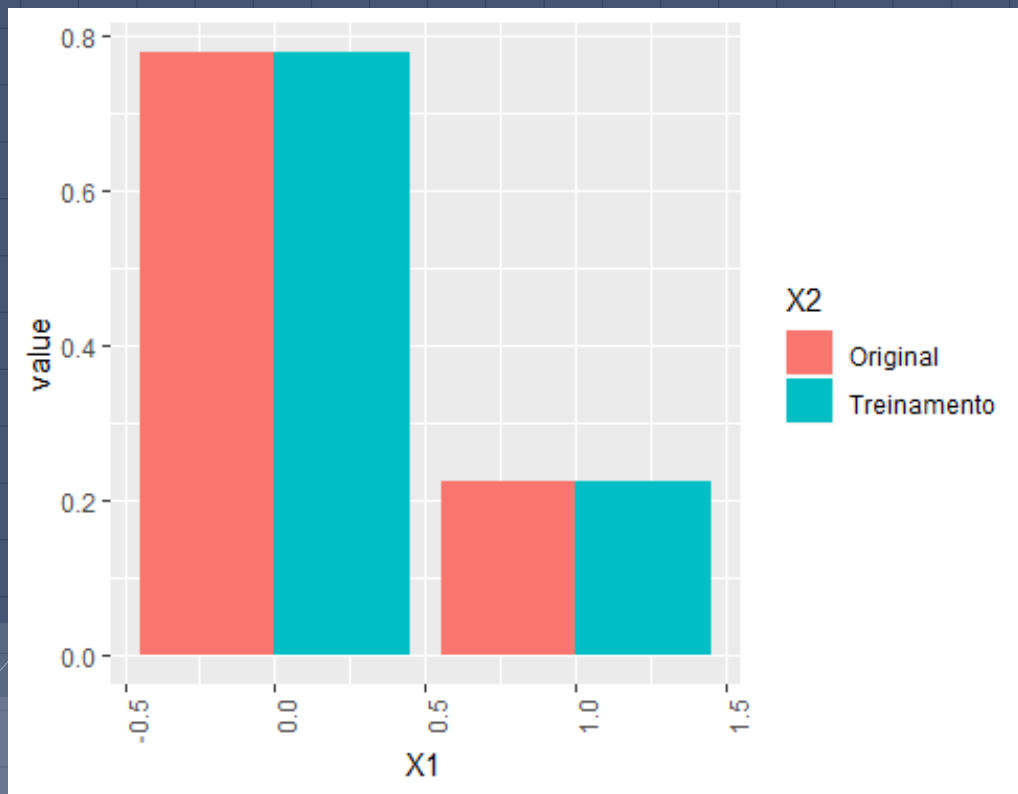
Valores Missing Observados



Dados Treinamento X Teste

```
109 # Amostragem estratificada. selecione as linhas de acordo
110 # com a variable inadimplente como strata
111 TrainingDataIndex <- createDataPartition(dataset$inadimplente,
112                                           p = 0.45, list = FALSE)
113
114 # Criar Dados de Treinamento como subconjunto do conjunto de dados
115 # com numeros de indice de linha conforme identificado acima e todas as colunas
116 trainData <- dataset[TrainingDataIndex,]
117
118 # Tudo o que nao esta no dataset de treinamento esta no dataset de
119 # teste. Observe o sinal - (menos)
120 testData <- dataset[-TrainingDataIndex,]
121
122 # Veja porcentagens entre as classes
123 prop.table(table(trainData$inadimplente))
124
```

Dados Treinamento X Teste



Validação Cruzada

```
142 # Usaremos uma validacao cruzada de 10 folds
143 # para treinar e avaliar modelo
144 TrainingParameters <- trainControl(method = "cv", number = 10)
145
```



Algoritmo

Foram utilizados três algoritmos para geração dos modelos.



Algoritmos

30


Random Forest

Árvore

Naive Bayes

Random Forest

É um algoritmo de aprendizado supervisionado. Como você já pode ver pelo seu nome, ele cria uma floresta e faz de forma aleatória. A "floresta" que constrói, é um conjunto de Árvores de Decisão, na maioria das vezes treinadas com o método "bagging". A ideia geral do método de bagging é que uma combinação de modelos de aprendizagem aumenta o resultado geral.

A decorative background graphic at the bottom of the slide. It features a white line chart with circular markers at various points, showing an overall upward trend. Below the line chart is a series of vertical bars of varying heights, resembling a bar chart or histogram, rendered in a light blue-grey color.

Random Forest

```
146 ##### Random Forest #####
147
148 # Construindo o Modelo
149 rf_model <- randomForest(inadimplente ~ ., data = trainData)
150
151 # Salvando o modelo
152 saveRDS(rf_model, file = "rf_model.rds")
153
154 # Importancia das variaveis preditoras para as previsoes
155 varImpPlot(rf_model)
```


Random Forest

```
176 # Previsoes
177 predictionrf <- predict(rf_model, testData)
178
179 # Confusion Matrix
180 cmrf <- confusionMatrix(predictionrf, testData$inadimplente, positive = "1")
181 cmrf
182
183 # Plotando a Matriz de Confusao
184 Matriz_Confusao <- cmrf$table
185 plot(Matriz_Confusao)
186
```

Árvore

```
205 - ##### Arvore #####
206
207 # Construindo o Modelo
208 arvore = rpart(inadimplente ~ PAY_0 + BILL_AMT1 + BILL_AMT2 + LIMIT_BAL,
209               data = trainData, cp = .02)
210 arvore
211 summary(arvore)
212
213 # salvando o modelo
214 saveRDS(arvore, file = "arvore_model.rds")
215
216 # Previsao
217 predictionarvore = predict(arvore, newdata = testData)
```

Árvore

```
225 # Verificando o resultado da predição
226 test = cbind(testData, predictionarvore)
227
228 # Renomeando a coluna de classe
229 colnames(test)[25] <- "nao"
230 colnames(test)[26] <- "sim"
231
232 # verificando o resultado da previsao
233 test['result'] = ifelse(test$sim >= 0.5, '1', '0')
234
235 # Convertendo a class e o resultado da predicao para fator
236 test$result <- as.factor(test$result)
237 test$inadimplente <- as.factor(test$inadimplente)
238
239 # Confusion Matrix
240 cmrfArvore <- confusionMatrix(test$inadimplente, test$result)
```

Naive Bayes

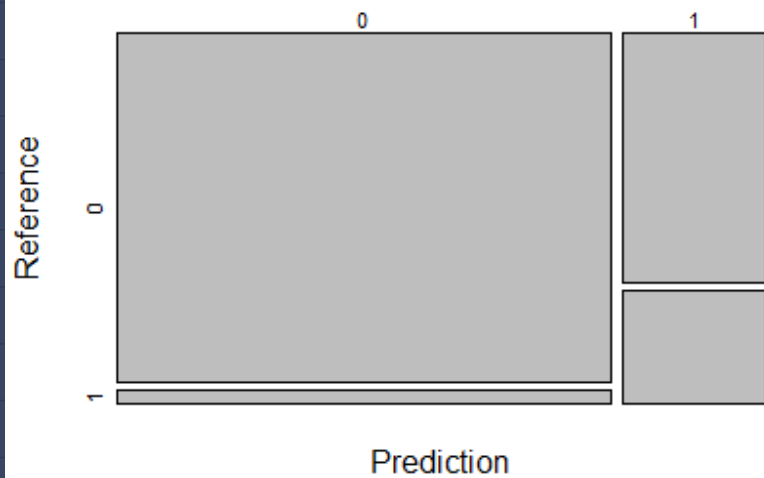
```
282 - ##### Naive Bayes #####
283
284 # criando o Modelo
285 modeloNaiveBayes = NaiveBayes(inadimplente ~ PAY_0 + BILL_AMT1 +
286                               BILL_AMT2 + LIMIT_BAL, trainData)
287 # salvando o modelo
288 saveRDS(modeloNaiveBayes, file = "naivebayes_model.rds")
289
290 # Fazendo as predicoes
291 predicaoNaivesBayes = predict(modeloNaiveBayes, testData)
292
293 # Confusion Matrix
294 cmrfNaive <- confusionMatrix(testData$inadimplente, predicaoNaivesBayes$class)
```

Visualização do Modelo

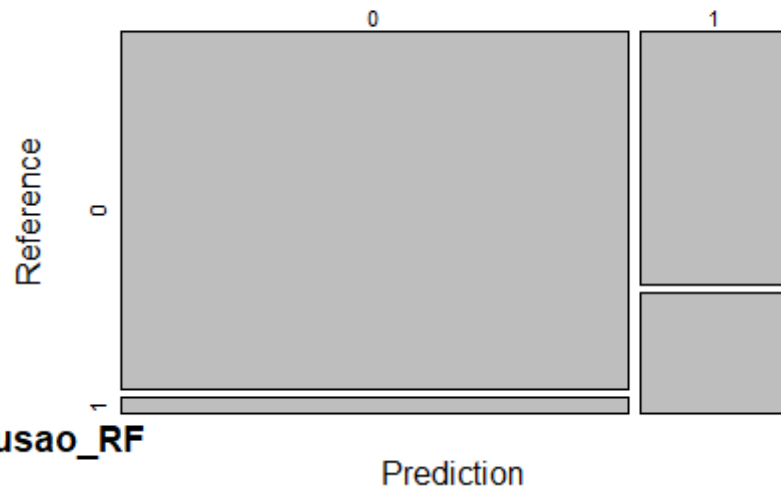
Visualização dos modelos, gerados
pelos algoritmos, no PowerBI



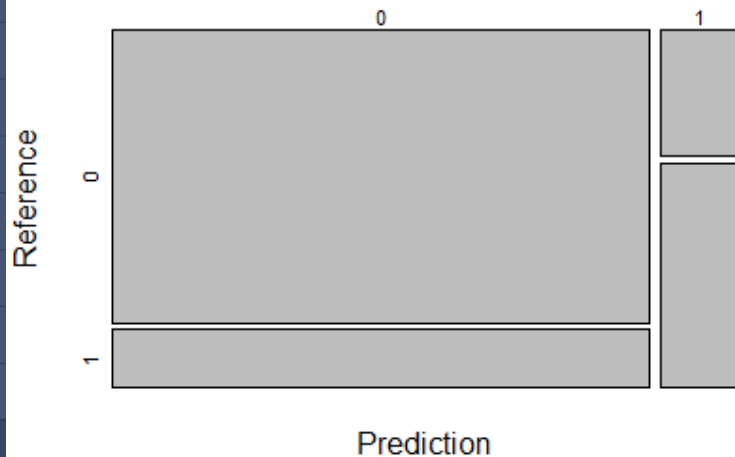
Matriz_Confusao_NaiveBayes



Matriz_Confusao_Arvore



Matriz_Confusao_RF



81,24%

Accuracy de teste para o Random Forest

81,75%

Accuracy de teste para o Árvore

81,65%

Accuracy de teste para o Naive Bayes

Referências

- ▣ R disponível em < <https://www.r-project.org/about.html> >
- ▣ Power BI < <https://powerbi.microsoft.com/pt-br/> >
- ▣ Dataset disponível em < <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#> >
- ▣ Random Forest disponível em < <https://machinelearning-blog.com/2018/02/06/the-random-forest-algorithm/> >
- ▣ Curso de Power BI disponível em < www.datascienceacademy.com.br >

Obrigado!
Perguntas?

