Relatório Bázico

Matheus Martins Santos

2023-04-22

Contents

Desafio

A partir de um conjunto de dados de compras passadas de clientes, desenvolva um modelo capaz de prever quais clientes são mais prováveis de realizar uma recompra nas próximas duas semanas. O modelo pode levar em conta diversos fatores como histórico de compras, dados demográficos e preferências de produtos.

A partir de dois conjuntos de dados, onde o primeiro é uma base é de vendas da empresa, com as seguintes colunas:

O desafio porposto é:
* Desenvolver um modelo que prediga quais clientes são mais prováveis de recomprar nas próximas 2 semanas.
* Listar os clientes de mais prováveis a menos prováveis em termos de propensão a recompra.
* Proveja uma breve explicação dos fatores que o modelo tem levado em conta e como eles influenciam nas previsões.
* Proveja uma avaliação da performance do modelo.

Passos iniciais

Bibliotecas

```
library(readxl) # importar dados do Excel para o R
library(dplyr) # pacote para manipulação de dados
library(tidyverse) # para visualizações
library(stats) # pacote para realização de testes de normalidade
library(nortest) # pacote para realização de testes de normalidade
library(rpart) # pacote para construção de árvores de decisão
library(rpart.plot) # pacote para visualização de árvores de decisão construídas com o rpa
library(knitr) # Usado para produzir relatórios dinâmicos e reprodutíveis
library(kableExtra) # pacote é usado para produzir tabelas atraentes e personalizadas
library(DT) # para imprimir a tabeça
library(corrplot) # para calcular as correlações
library(reshape2) # para auxiliar com data.frame
```

Importando dados

```
setwd("F:\\bazico") # Diretorio de trabalho
clientes <- read_excel("clientes.xlsx")
vendas_de_produtos <- read_excel("vendas_de_produtos.xlsx")
```

Preparação dos Dados

```
## convertendo as variaveis em numerico e fatores, númericas e Data
vendas_de_produtos$ID_Cliente = as.factor(vendas_de_produtos$ID_Cliente)
vendas_de_produtos$ID_Produto = as.factor(vendas_de_produtos$ID_Produto)
vendas_de_produtos$ID_Pedido = as.factor(vendas_de_produtos$ID_Pedido)

vendas_de_produtos$Quantidade = as.numeric(vendas_de_produtos$Quantidade)
```

```r
vendas_de_produtos$Desconto = as.numeric(vendas_de_produtos$Desconto)
vendas_de_produtos$Frete = as.numeric(vendas_de_produtos$Frete)
vendas_de_produtos$Total_do_Pedido = as.numeric(vendas_de_produtos$Total_do_Pedido)
vendas_de_produtos$Data <- as.Date(vendas_de_produtos$Data)
```

Foi preciso converter os dados originais em fatores para as variaveis de (ID_Cliente,ID_Produto,ID_Pedido), da mesma forma foi convertido em variáveis númericas (Quantidade,Desconto,Frete,Total_do_Pedido) e em formato de Data a variavel (Data).

Organizando base Para trabalhar

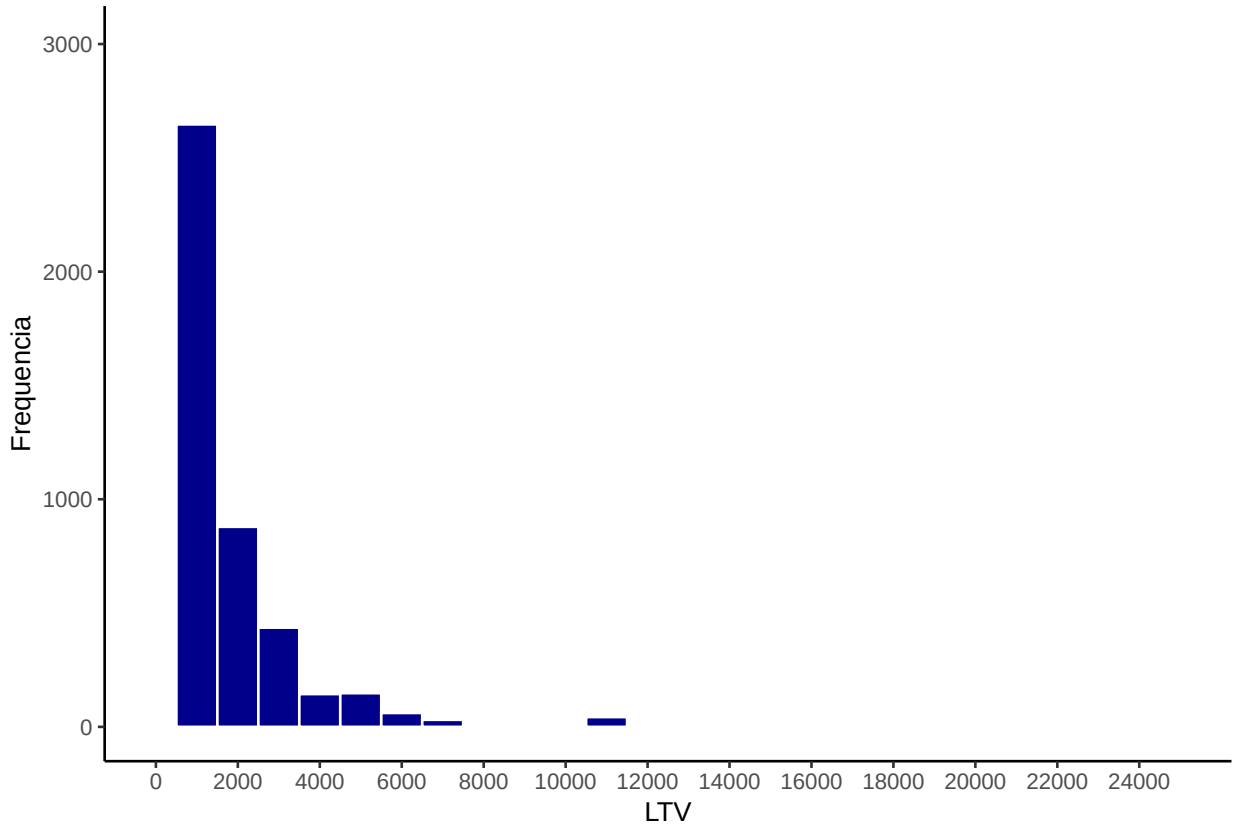
Em seguida, tive o objetivo de consolidar a base, onde foi agrupada por cliente e por Data de compra, com isso teria uma visão maior sobre o quanto o cliente comprou e o quanto pagou pelas compras, a quantidade de visitas na loja (Variável importante para definir se o cliente já fez um recompra na loja). Por fim, buscou juntar as informações de cada cliente com a localidade que o cliente reside, com base na tabela "Cliente".

```r
# Agrupando base por cliente e Data de compra;
# em seguinda, soma a quantidade de produto que o cliente comprou
# Soma o valor unitario de cada produto
# Calcula os descontos e frete
base1 <- vendas_de_produtos %>%
  mutate(valor_compra = Quantidade * Preço_Unitário) %>%
  group_by(ID_Cliente, Data) %>%
  summarise(Quantidade = sum(Quantidade),
            valor_compra = sum(valor_compra),
            Desconto = mean(Desconto),
            Frete = mean(Frete))

# Calculando o valor final da compra por cliente e data
# criando a variavel LifeTime value (LTV)
base1 <- base1 %>%
  mutate(valor_final_compra = (valor_compra - Desconto + Frete)) %>%
  group_by(ID_Cliente) %>%
  mutate(LTV = sum(valor_final_compra))

# Conta a quantidade de vezes que o cliente comprou na loja
base1 <- base1 %>%
  group_by(ID_Cliente) %>%
  mutate(quantidade_compras = n())

## Criando a variavel 'Recompra" que representa seo cliente já comprou ou não na loja
base1 <- base1 %>%
  group_by(ID_Cliente) %>%
  mutate(Recompra_Loja = ifelse(quantidade_compras >= 2, "Sim","Não"))

## Juntando a base organizada com a base de clientes
base <- merge(clientes,base1 ,by = "ID_Cliente")
base <- na.omit(base) # retira os valores faltantes.

# Convertendo variaveis em fator
base$Bairro <-  as.factor(base$Bairro)
base$Cidade <-  as.factor(base$Cidade)
```

```
base$Estado <-  as.factor(base$Estado)
base$Recompra_Loja = factor(base$Recompra_Loja, levels = c("Sim","Não"))
```

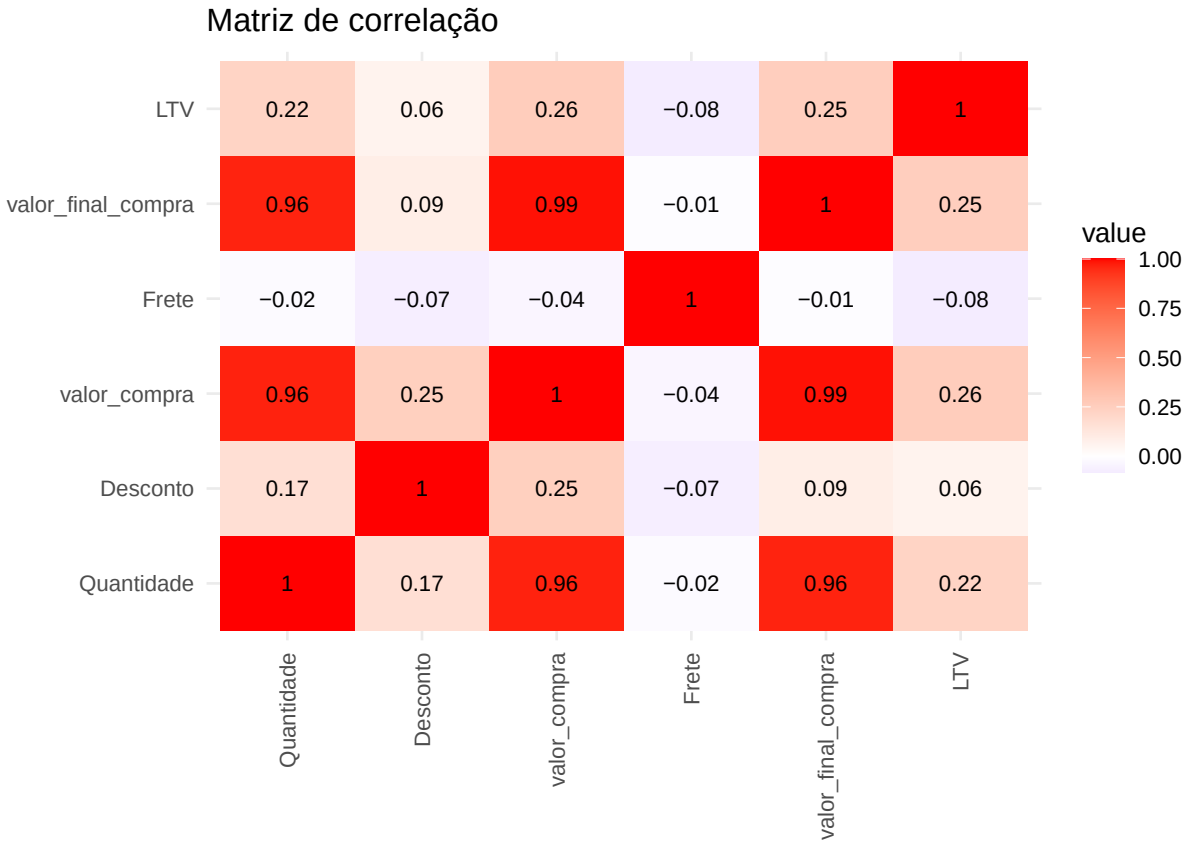
Obtendo Insights

Distribuição do LifeTime Value (LTV)



Conforme o gráfico acima, onde representa a distribuição do LifeTime Value - LTV, muito cliente gastaram um total ao longo da sua vida um valor acumulado menor que R$ 2.000,00, fazendo com que existe uma grande frequência nesse grupo, quando o LTV é maior que R$ 2.000,00, o número de cliente vai diminuir consideravelmente, mas existe cliente que compram muitos, podendo ter valores superiores a R$ 20.000,00

Matriz de correlação das variáveis númericas



Com base na matriz de correlação apresentada, podemos fazer as seguintes observações:

- A quantidade de itens comprados (“Quantidade”) tem uma forte correlação positiva com o valor da compra final (“valor_final_compra”), indicando que quanto mais itens são comprados, maior é o valor final da compra.
- A quantidade de itens comprados também tem uma correlação positiva forte com o valor da compra (“valor_compra”).
- O desconto dado na compra (“Desconto”) tem uma correlação positiva fraca com o valor da compra final (“valor_final_compra”), indicando que quanto maior o desconto, menor é o valor final da compra, mas essa correlação não é muito forte.
- O frete (“Frete”) tem uma correlação negativa fraca com o valor da compra final (“valor_final_compra”), o que sugere que o aumento do frete pode levar a uma diminuição no valor final da compra, mas essa correlação não é muito forte.
- O “LTV” (Lifetime Value) tem uma correlação positiva fraca com todas as outras variáveis, indicando que os clientes com maior LTV tendem a comprar mais itens, a gastar mais e a receber mais descontos.

avaliando a base de cliente por Cidade/Estado

Com base na Tabela 1, cerca de 6340 (85,46%) do cliente são residentes do Estados de Sergipe, os clientes gastaram em média R$ 1.193,17 em ao longo da sua vida com os produtos da Bázico, além disso, cerca de 50% dos cliente de Sergipe gastaram até R$ 683,80. A Tabela 2, mostra as top 15 Cidades que apresentaram com mais clientes da Bázico, a capital de Sergipe, Aracaju possui cerca de 6098 (82,34%), os clientes gastaram

Table 1: Estatistica por estado

Estado	Quantidade de Cliente	% Cliente	Média LTV	Mediana LTV
SE	6340	85.61%	1193.17	683.80
BA	262	3.54%	1197.10	855.53
SP	171	2.31%	750.72	452.00
PE	103	1.39%	706.42	478.21
CE	66	0.89%	746.00	417.01
PB	62	0.84%	2041.84	1100.79
AL	53	0.72%	699.57	582.50
RJ	53	0.72%	559.70	357.00
RN	42	0.57%	811.41	731.91
MG	41	0.55%	1182.56	462.07
DF	37	0.5%	737.19	491.97
PR	35	0.47%	906.64	1114.00
SC	26	0.35%	633.67	528.91
RS	25	0.34%	776.95	681.60
GO	18	0.24%	703.37	398.94
ES	14	0.19%	915.14	1053.55
MA	14	0.19%	350.48	351.90
MS	10	0.14%	1641.10	1797.00
AC	8	0.11%	607.15	676.58
PI	8	0.11%	1127.59	756.86
AM	5	0.07%	1121.43	526.42
MT	4	0.05%	953.65	997.17
PA	3	0.04%	304.03	302.11
RO	3	0.04%	1200.87	1200.87
TO	2	0.03%	301.86	301.86
RR	1	0.01%	205.60	205.60

em média R$ 1.205,97 em ao longo da sua vida com os produtos da Bázico, além disso, cerca de 50% dos cliente de Sergipe gastaram até R$ 692,00.

Table 2: Top 10 Cidades: Estatistica

Cidade	Quantidade de Cliente	% Cliente	Média LTV	Mediana LTV
Aracaju	6098	82.34%	1205.97	692.00
Salvador	184	2.48%	1337.16	928.92
São Paulo	93	1.26%	765.22	547.20
Barra dos Coqueiros	62	0.84%	1479.65	1190.00
Recife	48	0.65%	648.73	544.69
João Pessoa	47	0.63%	2386.71	833.00
Fortaleza	46	0.62%	849.25	417.01
Maceió	45	0.61%	740.43	710.33
Brasília	37	0.5%	737.19	491.97
Rio de Janeiro	36	0.49%	614.37	395.62

Árvore de Decisão - Decision Tree

dividindo a base em treino e teste

Foi considerando uma proporção de 80% para a base de treinamento, assim o modelo terá mais dados para poder aprender.

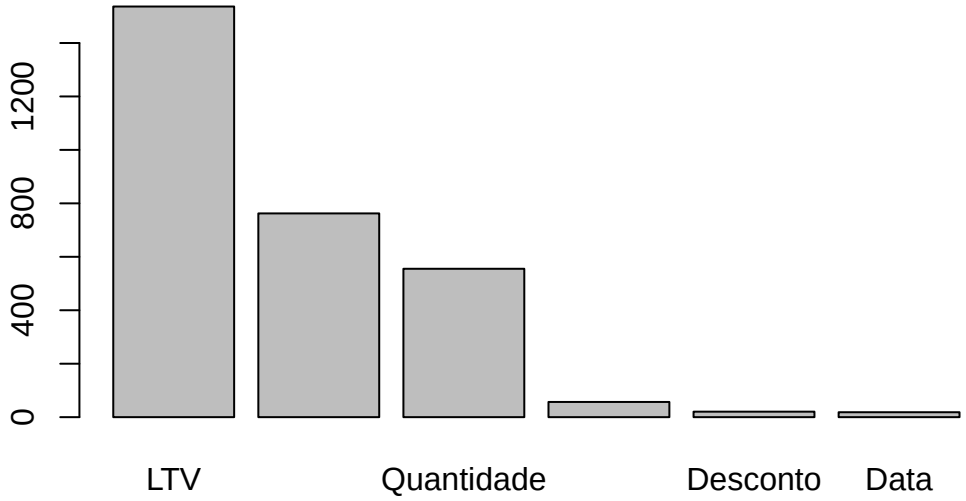
```
# definir a proporção do conjunto de treinamento
prop_treino <- 0.8

# definir uma semente aleatória para reprodutibilidade
set.seed(124)
# criar um vetor de índices aleatórios para dividir a base em treino e teste
indices <- sample(nrow(base), nrow(base)*prop_treino)

# selecionar as observações para o conjunto de treinamento
dados_treino <- base[indices,]

# selecionar as observações para o conjunto de teste
dados_teste <- base[-indices,]
```

Gerando o modelo com a base de treinamento

```
# MODELO DE arvore de Decisão ====
fit = rpart(Recompra_Loja ~ Data+Quantidade+Desconto+Frete+valor_final_compra+LTV,
            method = "class",
            data = dados_treino)
```

Esse é um modelo de árvore de decisão construído com base nos dados de treinamento da variável “Recompra_Loja” em função das seguintes variáveis: “Cidade”, “Estado”, “Quantidade”, “Desconto”, “Frete”, “valor_final_compra” e “LTV”. O modelo final contém apenas duas variáveis: “valor_final_compra” e “LTV”. Isso sugere que essas duas variáveis são as mais importantes na predição da variável “Recompra_Loja”. Abaixo é mostrado um gráfico da variáveis importante considerados no modelo

```
barplot(fit$variable.importance)
```



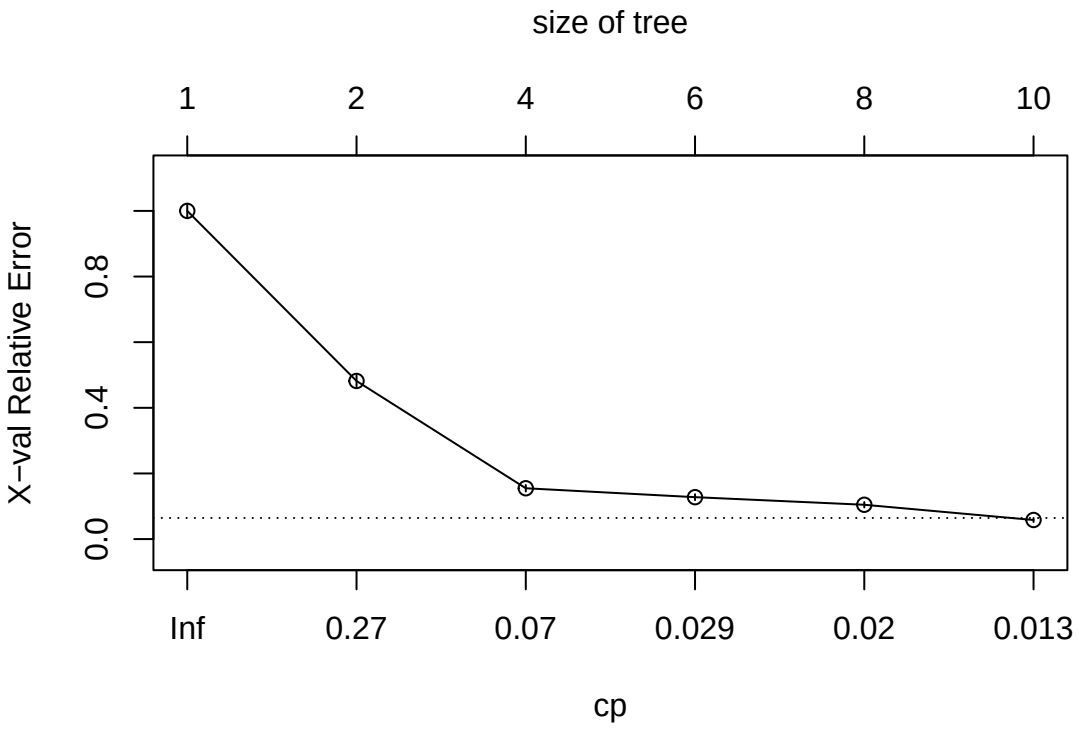
A taxa de erro do nó raiz é de 0,25506, o que significa que o modelo tem uma taxa de acerto de cerca de 74,5%. O modelo foi podado usando validação cruzada com um parâmetro de complexidade (CP) de 0,506287, o que resultou em um modelo com cinco nós terminais.

Table 3: Parametros Complexos do modelo

CP	**nsplit**	**rel error**	**xerror**	**xstd**
0.50628723	0	1.00000000	1.00000000	0.022203795
0.14791529	1	0.49371277	0.46988749	0.016544083
0.03342158	3	0.19788220	0.15155526	0.009819573
0.02481800	5	0.13103905	0.12905361	0.009088343
0.01654533	7	0.08140304	0.10655195	0.008282573
0.01000000	9	0.04831238	0.06618134	0.006562037

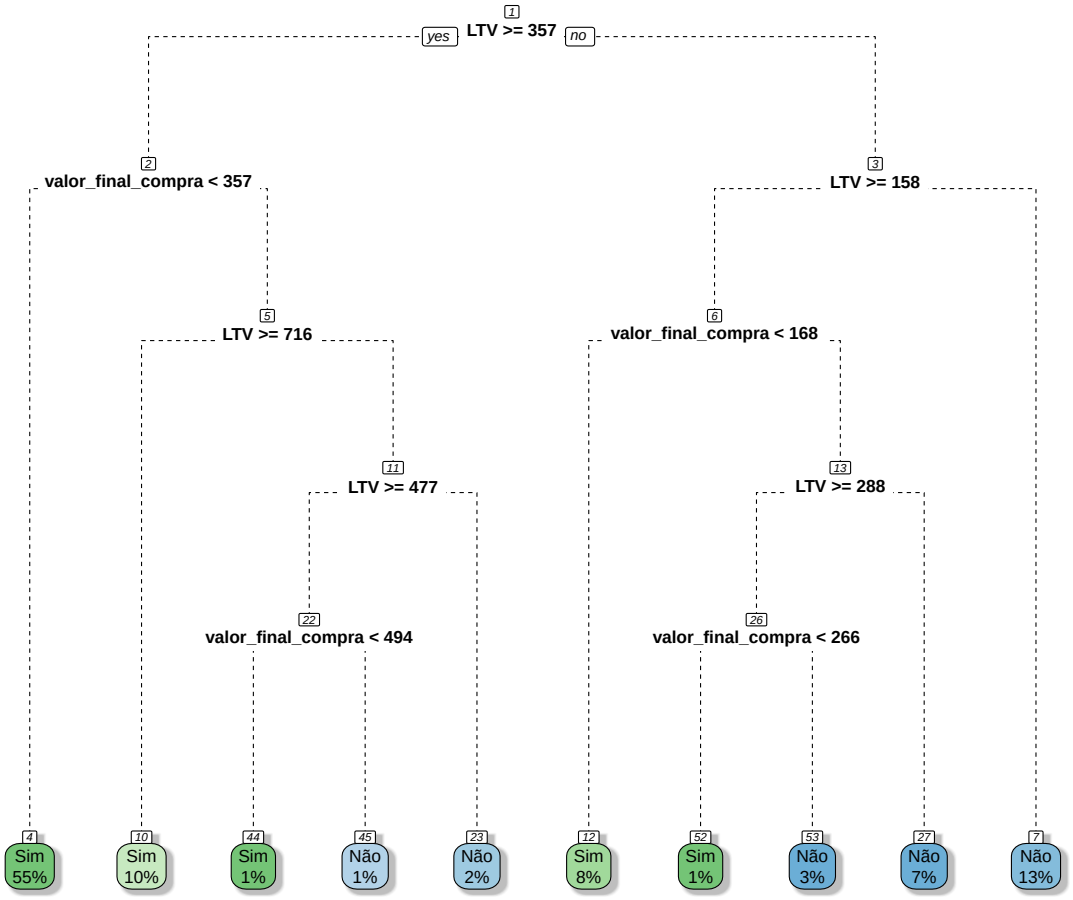
A tabela de CPs mostra os valores do parâmetro de complexidade que foram testados durante a poda do modelo. A medida que o CP diminui, mais nós são adicionados à árvore, aumentando a complexidade do modelo. A medida que o CP aumenta, mais nós são podados, diminuindo a complexidade do modelo e evitando o overfitting. O valor de CP que produziu o modelo final foi de 0,01.. Abaixo é apresetado um gráfico dos CP e o Erro relativo

```
plotcp(fit)
```



Plotando o modelo

```
rpart.plot(fit, # method graph
           type = 0,
           extra = 100,
           box.palette = 'GnBu',
           branch.lty = 2,
           shadow.col = "gray",
           nn = TRUE,
           cex = 1)
```



Interpretação do modelo

Esse modelo foi modelado para prever se um cliente fará uma recompra ou não com base em duas variáveis: LTV (Lifetime Value, valor de vida útil) e valor final da recompra. A árvore tem um nó raiz, que contém o total de observações usadas para construir o modelo (5924), a proporção de observações que fizeram uma recompra (0.745) e a proporção que não fizeram uma recompra (0.255).

Se a primeira decisão for verdadeira, o modelo segue para o segundo nó, que testa se o valor final da compra é menor que 357, se for verdadeira, o modelo prevê que o cliente fará uma recompra(prob = 100%), caso contrário, se a valor final da compra for maior 357 (2º nó) o modelo parte para o %º nó, onde vai testar se o LTV é maior ou igual que 719, se for, o modelo prevê que o cliente fará uma recompra(prob = 94,71%), se não, o modelo segue para o 11º nó onde vai testar o LTV é maior ou igual que 477, se for falsa, ,o modelo prevê que o cliente não fará uma recompra (prob = 94,82), mas se for verdadeira, o modelo segue para o 22º nó, onde vai testar se o valor final da compra é menor que 494, se for verdadeira, o modelo prevê que o cliente fará uma recompra(prob = 100%), caso contrário prevê que o cliente não fará uma recompra (prob = 83,09%).

Se LTV for menor que 357, a árvore segue para o terceiro nó, que testa se o LTV é maior ou igual a 158. Se for menor, a árvore prevê que o cliente não fará uma compra (prob = 51,7%), caso contrário, a árvore segue para o 6º nó, que testa se o valor final da compra é menor que 168, se for o caso, o árvore prevê que

o cliente fará uma recompra (prob = 99,5), caso contrário, se o valor final da compra for maior que 168, o modelo segue para o 13º nó, onde testa se o LTV é maior ou igual a 288, se for menor, a árvore prevê que o cliente não fará uma recompra (prob = 99,5%), se for maior, o modelo segue para o 26º nó, que testa se o valor final da compra é menor que 266, se for o caso, a árvore prevê que o cliente fará uma recompra (prob = 100%), caso contrário o modelo prevê que o cliente não fará uma recompra (prob = 100%).

Predição

Utilizado o modelo treinado com a base teste para testar se o modelo está fazendo boa previsões.

```
predicao_teste <- predict(fit, newdata = dados_teste, type = "class")
```

Matriz de confusão

```
matriz_confusao <- table(predicao_teste, dados_teste$Recompra_Loja)
```

Table 4: Matriz de confusão

Valores Previstos	Sim	Não
Sim	1082	10
Não	18	372

Interpretando a matriz de confusão, temos que:

- O modelo classificou corretamente 1082 clientes como verdadeiros positivos, ou seja, previu corretamente que esses clientes farão uma recompra.
- O modelo classificou corretamente 372 clientes como falso verdadeiro, ou seja, previu corretamente que esses clientes não farão uma recompra.
- O modelo classificou incorretamente 10 clientes como falsos positivo, ou seja, previu incorretamente que esses clientes farão uma recompa, quando na verdade eles não fizeram.r
- O modelo classificou incorretamente 18 clientes como falsos negativos, ou seja, previu incorretamente que esses clientes não farão uma recompra, quando na verdade eles fizeram.s.

Acurácia

```
acuracia <- sum(diag(matriz_confusao))/sum(matriz_confusao)
round(acuracia,4)*100
```

```
## [1] 98.11
```

A acurácia é uma métrica de avaliação de modelos de classificação que mede a proporção de observações classificadas corretamente pelo modelo em relação ao total de observações. Em outras palavras, a acurácia representa a capacidade do modelo de classificar corretamente as observações em todas as classes. Desse modo, o modelo apresentou uma acertividade de 98,11% das observações.

Table 5: Top 10 Cliente mais propensos de fazer uma recompra

ID_Cliente	probabilidade
12268398905	1
12383202291	1
12383340726	1
12383774078	1
12384988715	1
12385092320	1
12419503756	1
12428854910	1
12429082030	1
12429263068	1

Prevendo as probabilidades dos clientes fazer uma recompra

```
# fazer a previsão das probabilidades de recompra para o conjunto de teste
probabilidade_teste <- predict(fit, newdata = dados_teste, type = "prob")
```

Juntando as probabilidades ao conjuntos de teste e imprimindo a lista do cliente mais provavéis

```
# juntar as probabilidades previstas ao conjunto de teste
dados_teste_com_prob <- cbind(dados_teste, probabilidade_teste[,1])

# renomeando nome da coluna
colnames(dados_teste_com_prob)[ncol(dados_teste_com_prob)] <- "probabilidade_recompra"

# imprimindo a tabela
df <- dados_teste_com_prob %>%
  select(ID_Cliente, probabilidade_recompra) %>%
  group_by(ID_Cliente) %>%
  summarise(probabilidade = round(mean(probabilidade_recompra),4)) %>%
  arrange(desc(probabilidade))

kable(head(df,10),caption = 'Top 10 Cliente mais propensos de fazer uma recompra',
      align = 'cc') %>%
  kable_styling(full_width = F,
                bootstrap_options = c("striped", "hover", "condensed", "responsive"),
                fixed_thead = T)
```

Na tabela 5 foi impresso apenas os top 10 clientes mais propensos a fazer uma recompra na loja, para poder ter acesso a todos os clientes é só seguir clincando no link: https://rpubs.com/Matheusmartin04/1034064. Ao clicar nesse link, leva para uma versão desse documento em HTML, e no final do documento, tem todos os cliente impressos.