

# Árvore de Decisão

Gabriel Rodrigues

2025-11-11

## Table of contents

<b>Introdução</b>	<b>1</b>
O que é Aprendizado de Máquina? . . . . .	1
Tipos de Aprendizado de Máquina . . . . .	2
<b>Árvore de Decisão</b>	<b>2</b>
Estrutura e funcionamento . . . . .	2
<b>Construindo Árvores de Decisão no R</b>	<b>3</b>
Bloco 1 - Instalando pacotes e importando dados . . . . .	3
Bloco 2 - Implementando o Modelo . . . . .	4
Bloco 3 - Gerando Visualização . . . . .	5
Bloco 4 - Classificando Novo Item . . . . .	5

## Introdução

### O que é Aprendizado de Máquina?

Em termos gerais, *Machine Learning* (ML) ou **Aprendizado de Máquina** é um subconjunto da Inteligência Artificial (IA) que permite que sistemas de computador aprendam e melhorem automaticamente a partir da experiência (dados), sem serem explicitamente programados para cada tarefa específica.

O emprego de métodos de Aprendizado de Máquina pode variar desde recomendações de filmes e músicas para usuários baseado em seu histórico de consumo, até na análise de imagens médicas para a detecção de doenças e diagnóstico.

## Tipos de Aprendizado de Máquina

Existem diversas abordagens para o aprendizado, mas as duas categorias principais são:

- **Aprendizado Supervisionado** (Supervised Learning): O modelo é treinado usando dados rotulados. É como aprender com um professor que sempre fornece as respostas corretas. **Exemplo:** Treinar um modelo com imagens já marcadas como “SPAM” ou “NÃO SPAM” para que ele aprenda a filtrar novos e-mails.
- **Aprendizado Não Supervisionado** (Unsupervised Learning): O modelo é treinado usando dados não rotulados e tem que encontrar a estrutura, agrupamentos (clustering) e padrões por conta própria. É como agrupar itens por cor sem que ninguém diga o nome das cores. **Exemplo:** Agrupar clientes de e-commerce com base em seus hábitos de compra para segmentação de marketing.

## Árvore de Decisão

Uma Árvore de Decisão (*Decision Tree*) é um algoritmo de aprendizado de máquina supervisionado usado tanto para tarefas de classificação quanto de regressão. Ela modela decisões sequenciais baseadas em características dos dados, de forma que se assemelha a uma estrutura de fluxograma.

### Estrutura e funcionamento

O principal objetivo de uma Árvore de Decisão é criar um modelo que prevê o valor-alvo (o target) ao aprender regras de decisão simples inferidas das características dos dados (features).

- **Nó Raiz (Root Node):** É o ponto inicial, representa todo o conjunto de dados.
- **Nós de Decisão Internos (Internal Decision Nodes):** Representam testes em um atributo (característica) dos dados. A partir deles, o conjunto de dados é dividido em subconjuntos com base no resultado do teste. Por exemplo, “A idade é maior que 30?”
- **Ramos (Branches/Edges):** São as possíveis respostas ou caminhos resultantes do teste em um nó de decisão.
- **Nós Folha (Leaf Nodes):** Representam os resultados finais ou a decisão. Em um problema de classificação, a folha contém o rótulo da classe (ex: “aprovado” ou “negado”). Em um problema de regressão, contém um valor previsto (ex: a média dos valores daquele subconjunto).

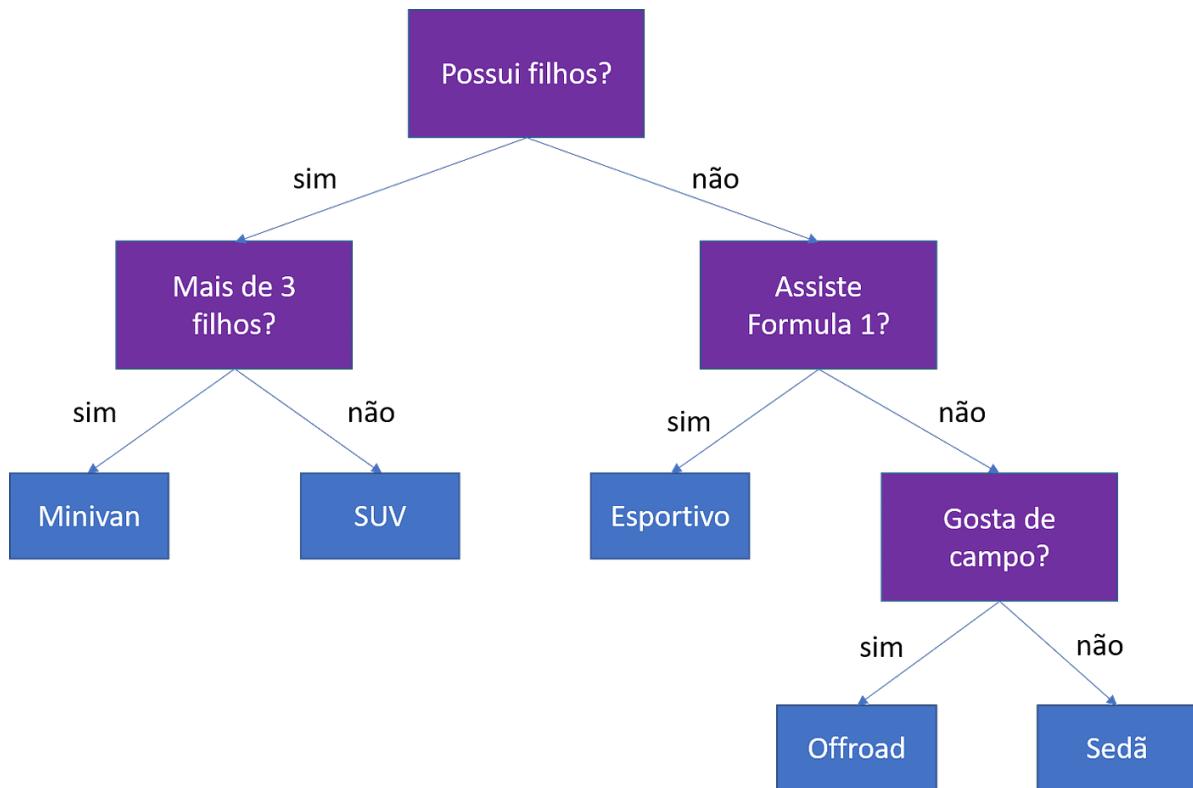


Figure 1: Exemplo de Árvore de Decisão

Nesse exemplo, a árvore de decisão se baseia em informações pessoais dos itens (das pessoas), para a partir delas prever o tipo de carro mais provável para compra.

## Construindo Árvores de Decisão no R

Para criar uma árvore de decisão no R, deve-se utilizar o pacote `rpart`, ele é um pacote em R para construir árvores de decisão que podem ser usadas para classificação ou regressão. A função principal `rpart()` cria o modelo da árvore de decisão, e pacotes adicionais como `rpart.plot` podem ser usados para visualização.

### Bloco 1 - Instalando pacotes e importando dados

```
# 1. Instale os pacotes (rode apenas uma vez)
# install.packages("rpart")
```

```

# install.packages("rpart.plot")

# 2. Carregue os pacotes
library(rpart)
library(rpart.plot)

# 3. Carregue o dataset "iris" (que já vem com o R)
# Vamos chamá-lo de "dados" para ficar mais claro
dados <- iris

# Dê uma olhada rápida nos dados
# Queremos prever "Species" usando as outras 4 colunas
head(dados)

```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

## Bloco 2 - Implementando o Modelo

```

# Treina o modelo da árvore
# method = "class" informa ao rpart que é um problema de CLASSIFICAÇÃO
modelo_arvore <- rpart(
  Species ~ .,
  data = dados,
  method = "class"
)

# Para ver um resumo simples das regras em texto
print(modelo_arvore)

```

```

n= 150

node), split, n, loss, yval, (yprob)
  * denotes terminal node

```

```

1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)
2) Petal.Length< 2.45 50    0 setosa (1.00000000 0.00000000 0.00000000) *
3) Petal.Length>=2.45 100   50 versicolor (0.00000000 0.50000000 0.50000000)
6) Petal.Width< 1.75 54    5 versicolor (0.00000000 0.90740741 0.09259259) *
7) Petal.Width>=1.75 46    1 virginica (0.00000000 0.02173913 0.97826087) *

```

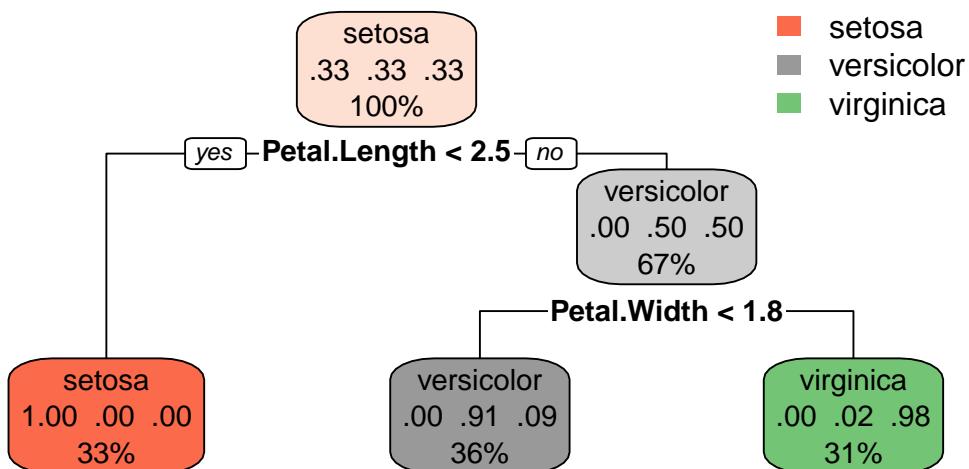
### Bloco 3 - Gerando Visualização

```

# Criar a visualização da árvore
# Esta é a "saída" do algoritmo!
rpart.plot(modelo_arvore, main = "Árvore de Decisão para Prever Espécie de Flor Iris")

```

## Árvore de Decisão para Prever Espécie de Flor Iris



### Bloco 4 - Classificando Novo Item

```

# Vamos criar uma "nova flor" que medimos no jardim
# Ela tem pétalas longas e largas.
nova_flor <- data.frame(
  Sepal.Length = 5.1,

```

```
Sepal.Width = 3.2,  
Petal.Length = 4.8,  
Petal.Width = 1.9  
)  
  
# Usar o modelo para prever a qual classe (espécie) ela pertence  
previsao <- predict(modelo_arvore, nova_flor, type = "class")  
  
# Mostrar o resultado  
print(paste("A previsão do modelo para a nova flor é:", previsao))
```

[1] "A previsão do modelo para a nova flor é: virginica"

A partir da inserção do novo item, seguindo os mesmos parâmetros do dataset original, é possível classificar um novo item em uma das categorias estabelecidas.