# Árvores de Decisão

## Definição Geral

A Árvore de Decisão é um algoritmo de aprendizado de máquina utilizado para resolver problemas de classificação e regressão. Ele funciona criando uma estrutura em forma de árvore, onde cada nó representa uma decisão a ser tomada e cada ramo representa uma possível conclusão a partir da decisão.

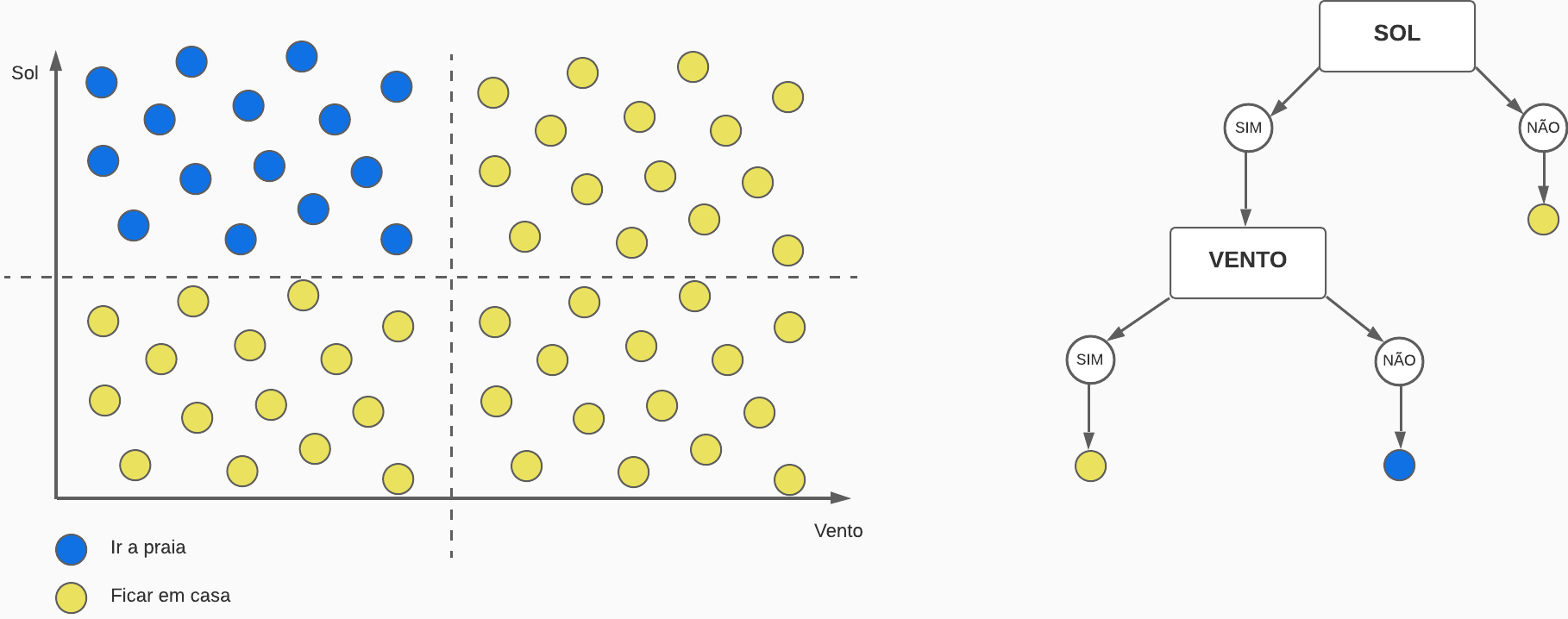
A construção da árvore começa com a seleção de um atributo do conjunto de dados como raiz da árvore. Em seguida, é feita uma divisão dos dados de acordo com os valores desse atributo. Esse processo é repetido para cada subconjunto de dados gerado, escolhendo-se sempre o atributo que melhor divide os dados.

A construção da árvore continua até que sejam satisfeitas determinadas condições pré-estabelecidas, como por exemplo, todos os exemplos em um nó pertençam a mesma classe ou todos os atributos tenham sido utilizados. Quando essas condições são satisfeitas, o nó é transformado em uma folha e a classe prevista é atribuída a ele.

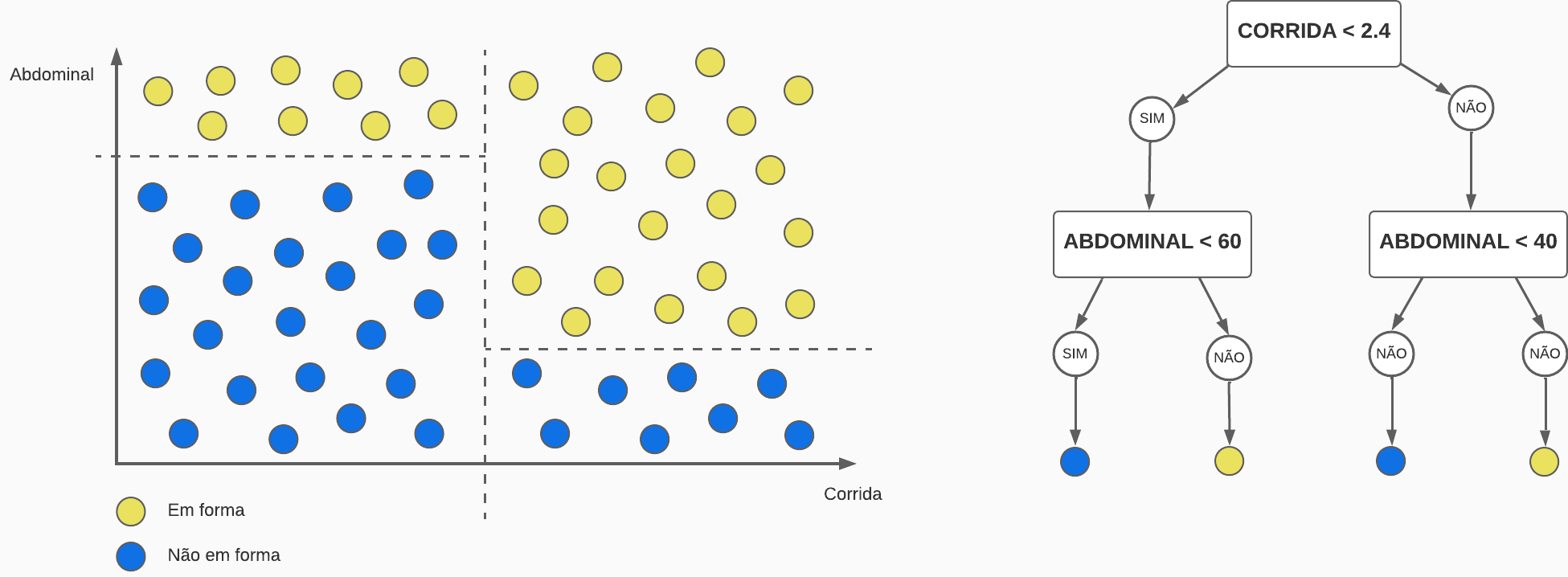
Para fazer uma previsão para um novo exemplo, basta seguir a árvore, respondendo às perguntas presentes nos nós, até chegar a uma folha. A classe prevista para o exemplo é então a classe atribuída à folha.

A Árvore de Decisão é uma técnica de aprendizado simples e eficiente, que permite ao usuário visualizar e entender as regras de decisão utilizadas pelo algoritmo. Além disso, ela é capaz de lidar com dados com atributos categóricos e numéricos, e pode ser utilizada em uma ampla variedade de aplicações.

**exemplo**:

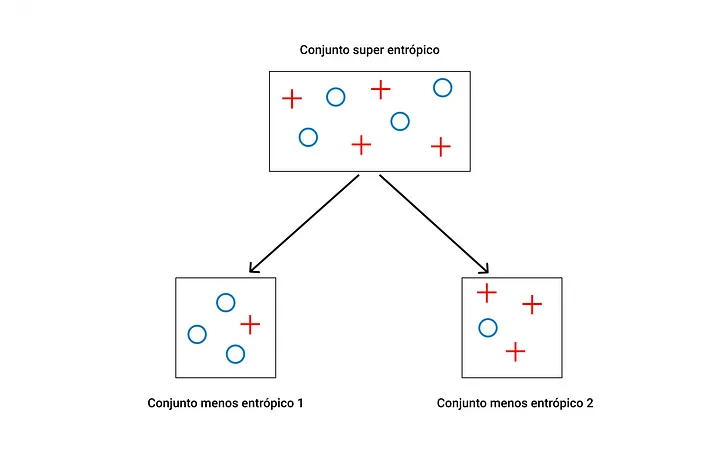


**exemplo**:



## Entropia

A Entropia pode ser definida como a medida que nos diz o quanto nossos dados estão desorganizados e misturados. Quanto maior a entropia, menor o ganho de informação e vice-versa. Nossos dados ficam menos entrópicos conforme dividimos os dados em conjuntos capazes de representar apenas uma classe do nosso modelo.



A partir desse momento, nosso objetivo se torna construir nossa árvore tendo o conjunto de dados inteiro como raiz e criar ramificações baseadas em condições que minimizem a entropia e aumentem o ganho de informação.

A criação de uma Árvore de Decisão usando entropia segue os seguintes passos:

1. Calcular a entropia inicial: a entropia é uma medida da impureza de um conjunto de dados. Para o problema de classificação, a entropia é dada por:

onde é a proporção de amostras da classe i.

1. Escolher o atributo que melhor divide os dados: para escolher o atributo, é necessário calcular o ganho de informação para cada atributo. Ela é dada por:

onde é a proporção de amostras do filho e é sua entropia.

O atributo que resultar em maior ganho de informação será escolhido como o próximo nó da árvore.

1. Dividir o conjunto de dados: o conjunto de dados é dividido em subgrupos (filhos), de acordo com os valores do atributo escolhido.
2. Repetir o processo para cada subgrupo: os passos 2 e 3 são repetidos para cada subgrupo gerado, até que todos os exemplos em um nó pertençam à mesma classe, todos os atributos tenham sido utilizados ou até que o critério de parada seja atingido.

Esse é um processo iterativo que continua até que a entropia de um nó seja 0 (todos os exemplos pertencem à mesma classe) ou não haja mais atributos para serem utilizados como testes de divisão. A árvore resultante representa uma série de testes de divisão que levam à classificação de um novo exemplo.

Como exemplo prático, vamos considerar a seguinte tabela:

| Salário | Localização | Função | Decisão |
| --- | --- | --- | --- |
| alto | longe | interessante | SIM |
| baixo | perto | desinteressante | NÃO |
| baixo | longe | interessante | SIM |
| alto | longe | desinteressante | NÃO |
| alto | perto | interessante | SIM |
| baixo | longe | desinteressante | NÃO |

1. Calcular a entropia inicial:

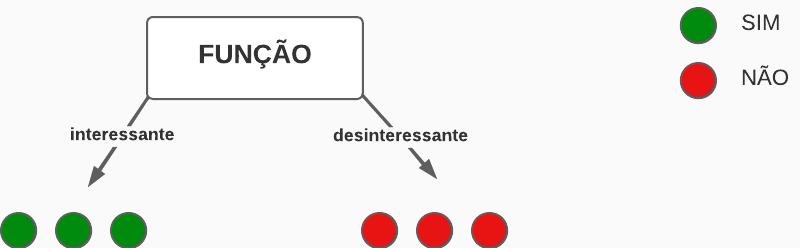
1. Cálculo do ganho de informação:

**SALÁRIO**

**LOCALIZAÇÃO**

**FUNÇÃO**

1. A primeira feature a ser escolhida é a Função, pois ela possui o maior ganho de informação (1). Como os elementos dos nós filhos possuem uma só classe, a árvore está finalizada.



## 

## Índice GINI

O índice de Gini é uma medida de impureza usada em algoritmos de árvore de decisão para medir a heterogeneidade de uma classe ou a proporção de itens que pertencem a uma determinada classe. Assim como a entropia, quanto menor o índice de Gini, maior é a homogeneidade dos dados e, portanto, maior é a certeza sobre a classe dos itens.

A seguir, está o passo a passo geral para desenvolver uma árvore de decisão usando o índice Gini:

1. O primeiro passo é calcular o índice Gini para cada atributo a fim de determinar qual deles é o melhor para separar as classes. A fórmula para isso é dada pela seguinte fórmula:

Onde é o índice Gini e é a proporção para cada subconjunto do atributo escolhido.

onde é a proporção de exemplos da classe j no subconjunto de dados do atributo.

1. Seleção do atributo com o menor índice Gini: O atributo com o menor índice Gini é selecionado como o melhor atributo para separar as classes.
2. Separação dos dados: O conjunto de dados é separado em subconjuntos baseado no atributo selecionado. Cada subconjunto é uma folha da árvore.
3. Continuação da construção da árvore: O processo é repetido para cada subconjunto, começando pelo cálculo do índice Gini para a classe em cada subconjunto, até que todas as classes estejam separadas ou o critério de parada seja atingido (por exemplo, todas as folhas têm a mesma classe ou todos os atributos já foram usados).

Esse é um resumo geral do processo para construir uma árvore de decisão usando o índice. Agora, vamos analisar em um exemplo prático:

| varA | varB | Classe |
| --- | --- | --- |
| 1 | 13 | 1 |
| 0 | -2 | 1 |
| 0 | 27 | 1 |
| 1 | 9 | 1 |
| 0 | 67 | 0 |
| 0 | 45 | 0 |
| 0 | 21 | 0 |
| 0 | 50 | 0 |

1. Cálculo do índice GINI (varA):

**varA = 1:**

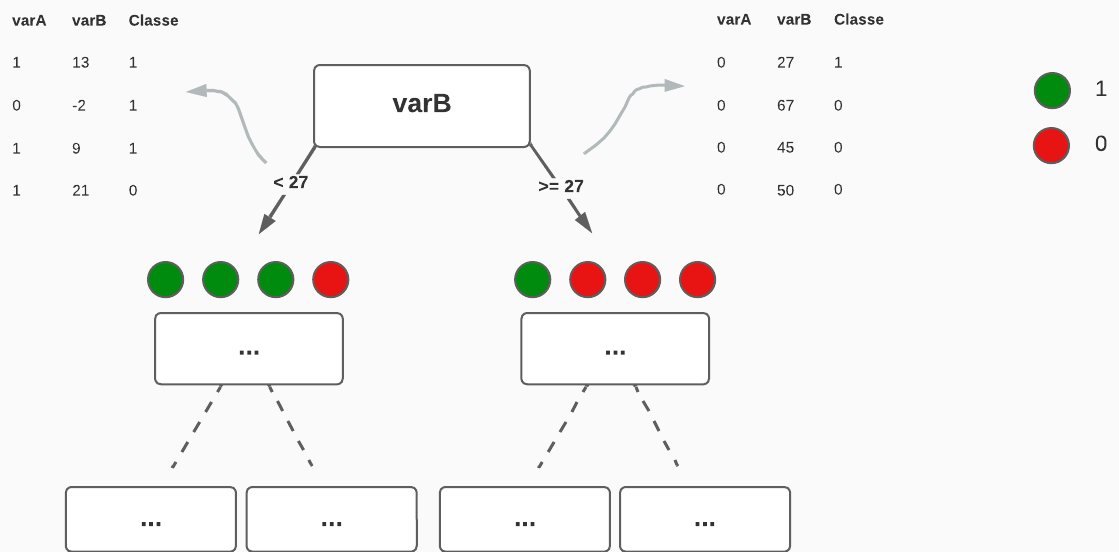
**varA = 0:**

1. Cálculo do índice GINI (varB):

**varB < 27:**

**varB >= 27:**

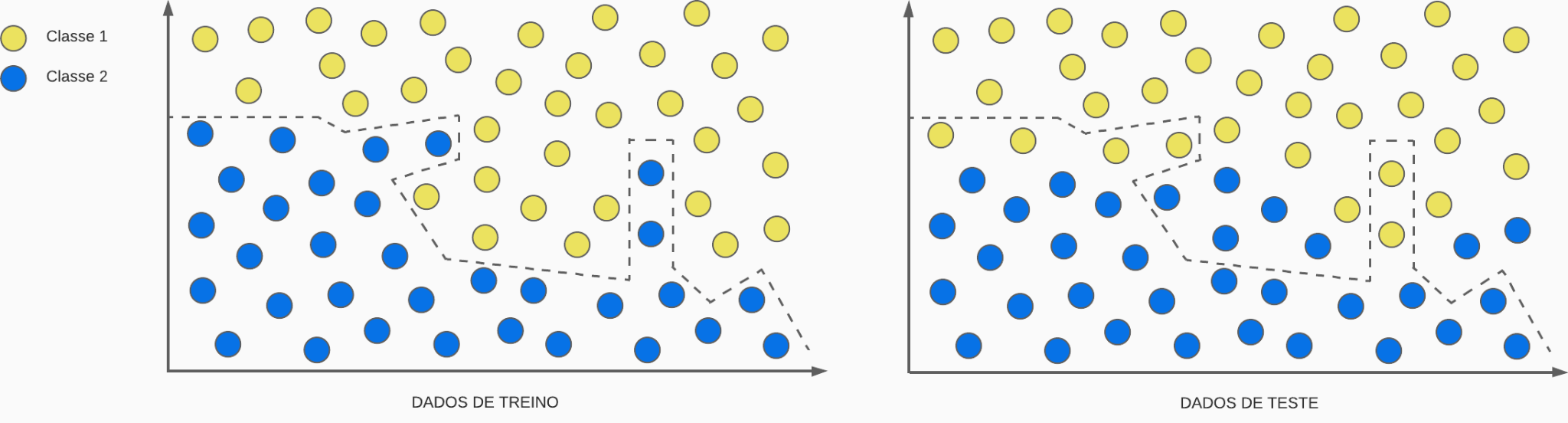
1. Para começar a criar nossa árvore de decisão, devemos escolher o atributo/variável com o menor índice GINI. Como , nossa árvore é inicialmente estruturada da seguinte forma:



## Evitando o Overfitting

Overfitting em árvores de decisão ocorre quando a árvore é treinada muito especificamente aos dados de treinamento, a ponto de começar a capturar ruídos ou variações aleatórias nos dados em vez de identificar padrões verdadeiros. Como resultado, a árvore pode ser muito complexa, com muitas folhas e ramificações, e não generaliza bem para novos dados.

Quando a árvore de decisão é treinada com overfitting, ela pode ter uma acurácia muito alta nos dados de treinamento, mas uma baixa acurácia nos dados de teste ou em dados novos. Isso indica que a árvore aprendeu os dados de treinamento muito bem, mas não foi capaz de generalizar as informações para dados desconhecidos.



Há algumas técnicas que podem ser usadas para evitar o overfitting em árvores de decisão, incluindo limitar a profundidade da árvore, adicionar penalidades para árvores complexas ou usar técnicas de poda. Além disso, é importante fornecer a árvore com uma boa amostra de dados de treinamento que sejam representativos do conjunto completo de dados, a fim de evitar o overfitting.

## Regressão

A árvore de decisão é uma técnica de aprendizado de máquina supervisionado que é amplamente utilizada não só para resolver problemas de classificação, mas também de regressão. Aqui está um passo a passo para desenvolver uma árvore de decisão para problemas de regressão usando a técnica com desvio padrão:

1. Calcular o desvio padrão inicial da nossa variável target ()
2. Escolha o critério de divisão: Para resolver problemas de regressão, o critério de divisão geralmente é o desvio padrão. Para calcular o desvio padrão de cada atributo/variável, deve-se utilizar a seguinte fórmula:

onde é a proporção e é o desvio padrão de cada subconjunto pertencente ao atributo.

Já a redução de desvio padrão é dada por:

1. Criar a raiz da árvore: A raiz da árvore é a primeira divisão que você fará em seus dados. Escolha a característica que tem a maior redução de desvio padrão e use-a para criar a raiz da sua árvore.
2. Criar as folhas da árvore: As folhas representam as saídas da árvore. Para resolver problemas de regressão, as saídas são valores contínuos. Para cada folha, calcule a média dos dados que foram alocados naquela folha.
3. Repetir os passos 1 a 4: Repita os passos 1 e 4 até que você alcance a profundidade desejada da sua árvore. Ou, você pode definir uma condição de parada, como o número mínimo de amostras por folha ou o erro mínimo.

Agora, vamos analisar em um exemplo prático:

| Temperatura | Domingo | Vendas |
| --- | --- | --- |
| QUENTE | SIM | 286 |
| FRIO | NÃO | 147 |
| AMENO | NÃO | 169 |
| FRIO | SIM | 172 |
| AMENO | NÃO | 176 |
| QUENTE | NÃO | 253 |
| QUENTE | NÃO | 238 |
| FRIO | NÃO | 151 |
| FRIO | SIM | 168 |
| QUENTE | NÃO | 264 |
| AMENO | SIM | 207 |
| QUENTE | SIM | 309 |
| QUENTE | NÃO | 245 |

1. Desvio padrão inicial:

1. Desvio padrão para cada atributo

**Temperatura**

| Temperatura |  | Amostras |
| --- | --- | --- |
| QUENTE | 24.65 | 6 |
| FRIO | 10.69 | 4 |
| AMENO | 16.51 | 3 |

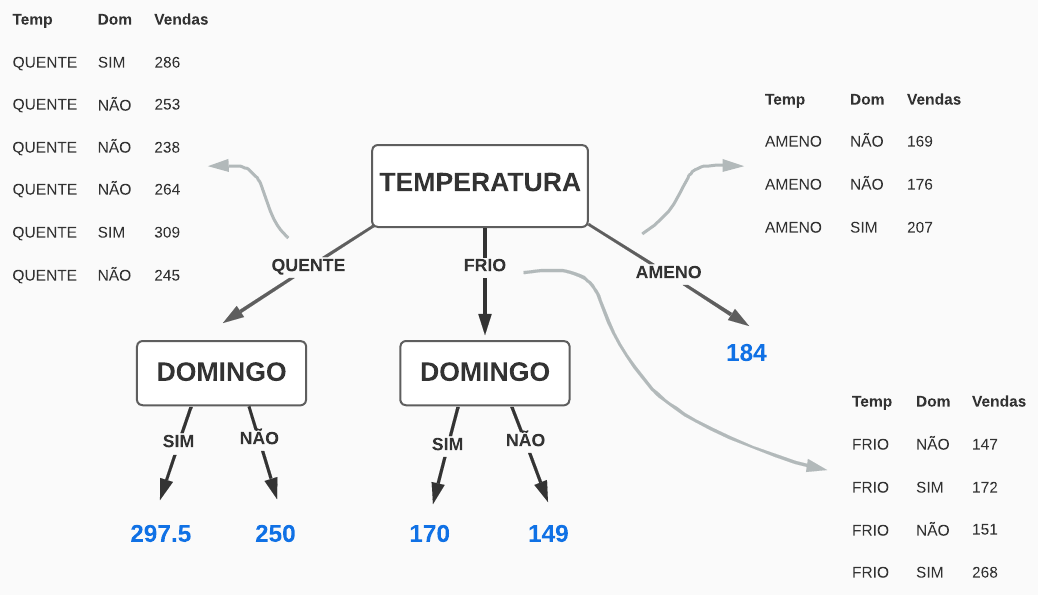
**Domingo**

| Domingo |  | Amostras |
| --- | --- | --- |
| QUENTE | 58.48 | 5 |
| FRIO | 45.94 | 8 |

1. O atributo temperatura tem a maior redução de desvio padrão. Nosso critério de parada consiste no número de amostras sendo 3. Com isso, podemos iniciar nossa árvore:

Como o subconjunto AMENO possui 3 amostras, o critério de parada já é aplicado, sendo calculada a média final desses dados.

Como os subconjuntos QUENTE e FRIO ainda não se aplicam ao critério de parada, eles ainda serão divididos entre a última variável e posteriormente calculadas suas médias.



## A escolha do melhor split com algoritmos CART e C4.5

CART (Classification and Regression Tree) e C4.5 são dois algoritmos de aprendizado de máquina que são usados para construir árvores de decisão. A escolha do melhor split é um processo crítico nestes algoritmos, pois afeta diretamente a performance e a precisão da árvore resultante.

A escolha do melhor split é feita comparando várias opções de divisão dos dados, a fim de encontrar a que resulta na menor impureza. A impureza é medida por diferentes métricas, como a entropia ou a Gini impurity, e representa a incerteza ou a probabilidade de classificação incorreta.

No CART, o split é escolhido pela divisão dos dados em dois subconjuntos de modo a minimizar a soma dos erros quadráticos (para a regressão) ou a impureza (para a classificação). O processo é repetido recursivamente em cada subconjunto até que uma condição de parada seja atingida, como um número mínimo de exemplos em um nó ou uma impureza mínima.

No C4.5, a escolha do melhor split é feita de maneira semelhante, mas a impureza é medida pela entropia e a escolha dos splits é baseada em uma heurística conhecida como information gain, que mede o quão informativo é o split em relação a uma classificação correta. O algoritmo também suporta a pruning, que é uma técnica para remover nós com pouco impacto na performance.

Em resumo, a escolha do melhor split é uma parte importante na construção de árvores de decisão e é feita comparando diferentes opções de divisão dos dados com base em métricas como a impureza ou a entropia, com o objetivo de encontrar a divisão que resulta em menor incerteza ou probabilidade de erro.