Árvore de Decisão

Douglas Cardoso

FEA.dev

April 2021

Árvores

Definição

É um grafo conexo, acíclico onde um dos vértices é diferenciado dos deamis, organizando os dados em um conjunto de **registros que satisfaz certas condições**.

- Registros serão chamados de **nós** (node);
- O nó por qual conseguimos acessar qualquer outro nó, é chamado de **raiz** (root);
- O nó que não tem ancestralidade de outros nós, é chamado de **folha** (leaf).

Árvore de decisão

Definição

Metodologia de representação não paramétrica, supervisionada que leva a resultados facilmente interpretáveis, construída por particionamentos dos espaços recursivamente em retângulos.

- Cada particionamento é **nó**;
- Cada resultado final é **folha**.

Representação gráfica de uma árvore

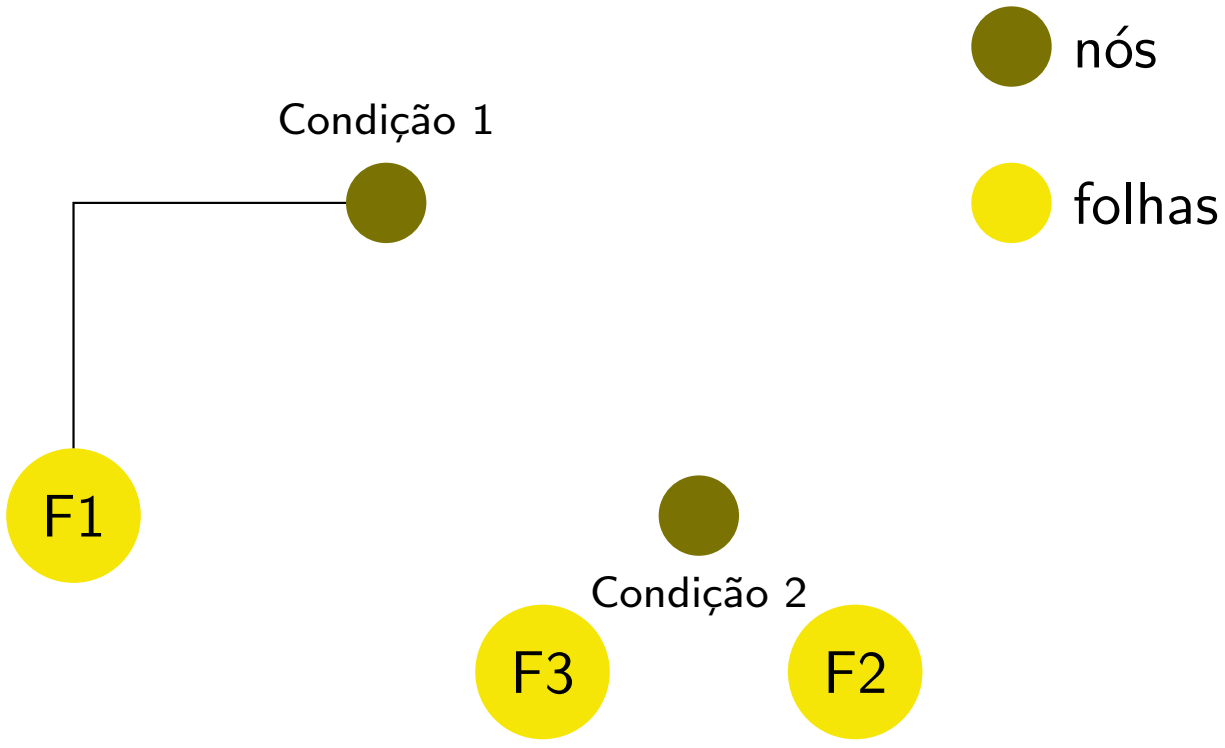


Figure: Exemplo de estrutura de uma árvore.

Representação gráfica de uma árvore

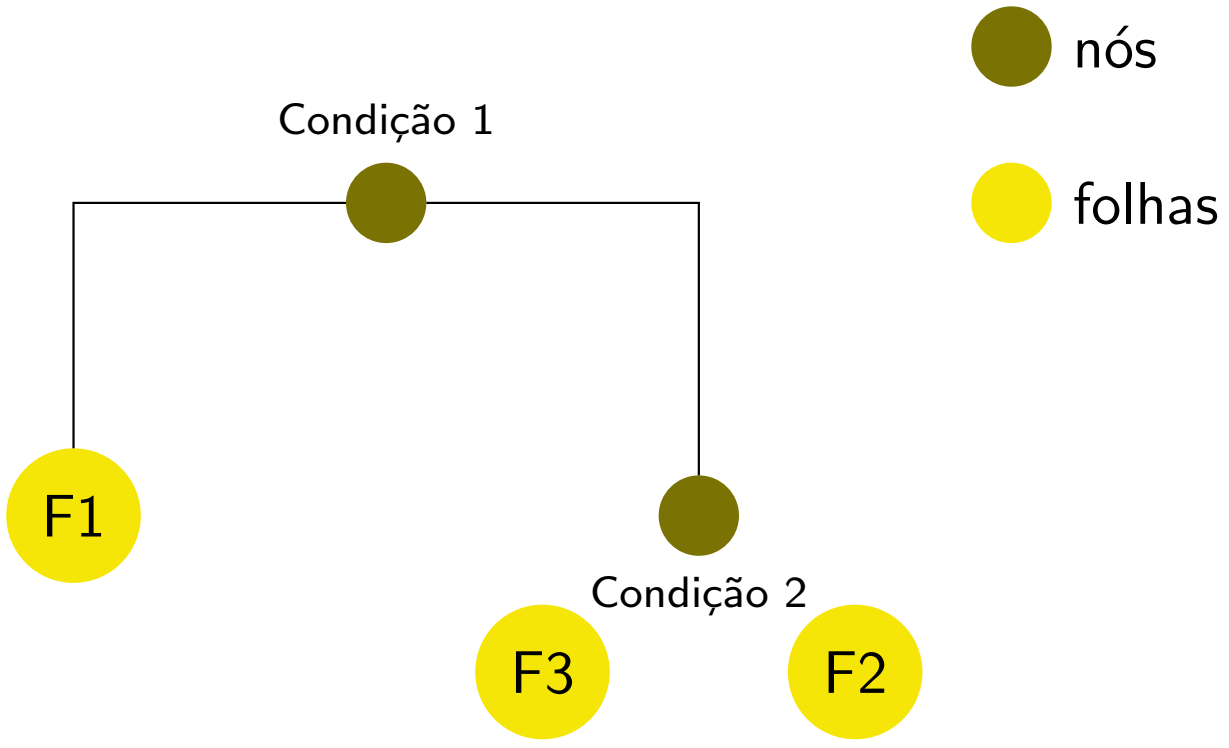


Figure: Exemplo de estrutura de uma árvore.

Representação gráfica de uma árvore

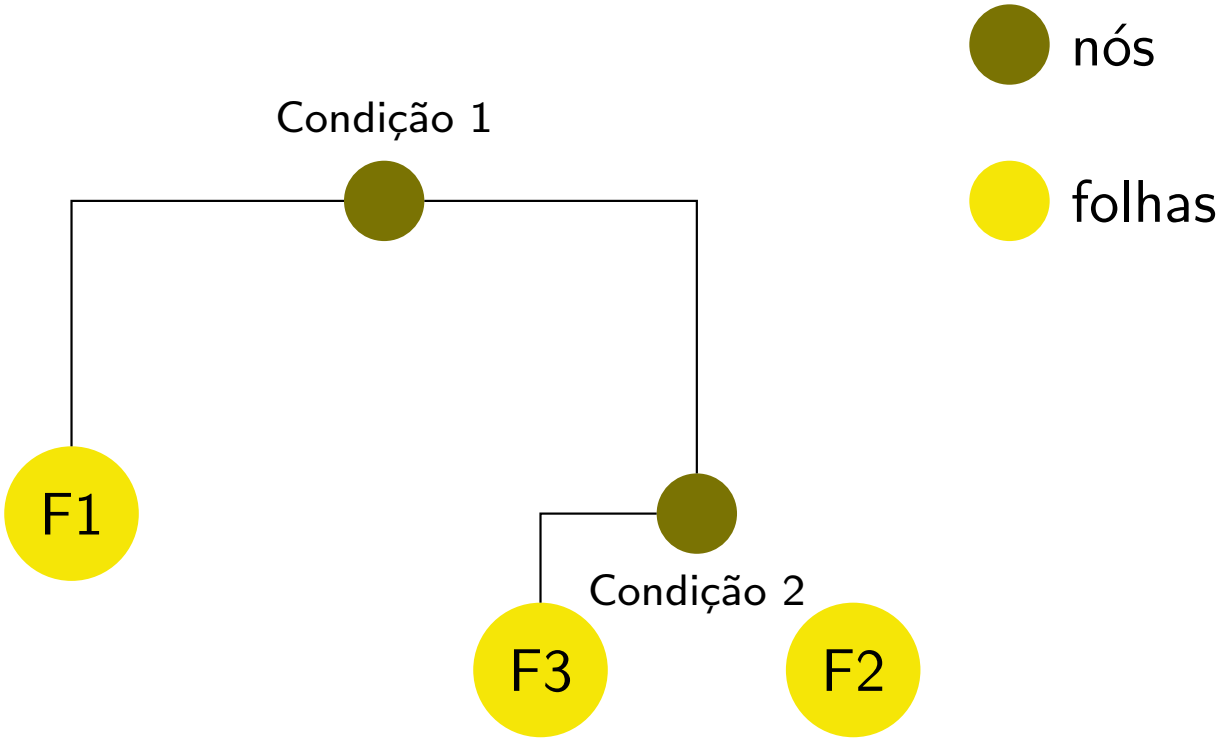


Figure: Exemplo de estrutura de uma árvore.

Representação gráfica de uma árvore

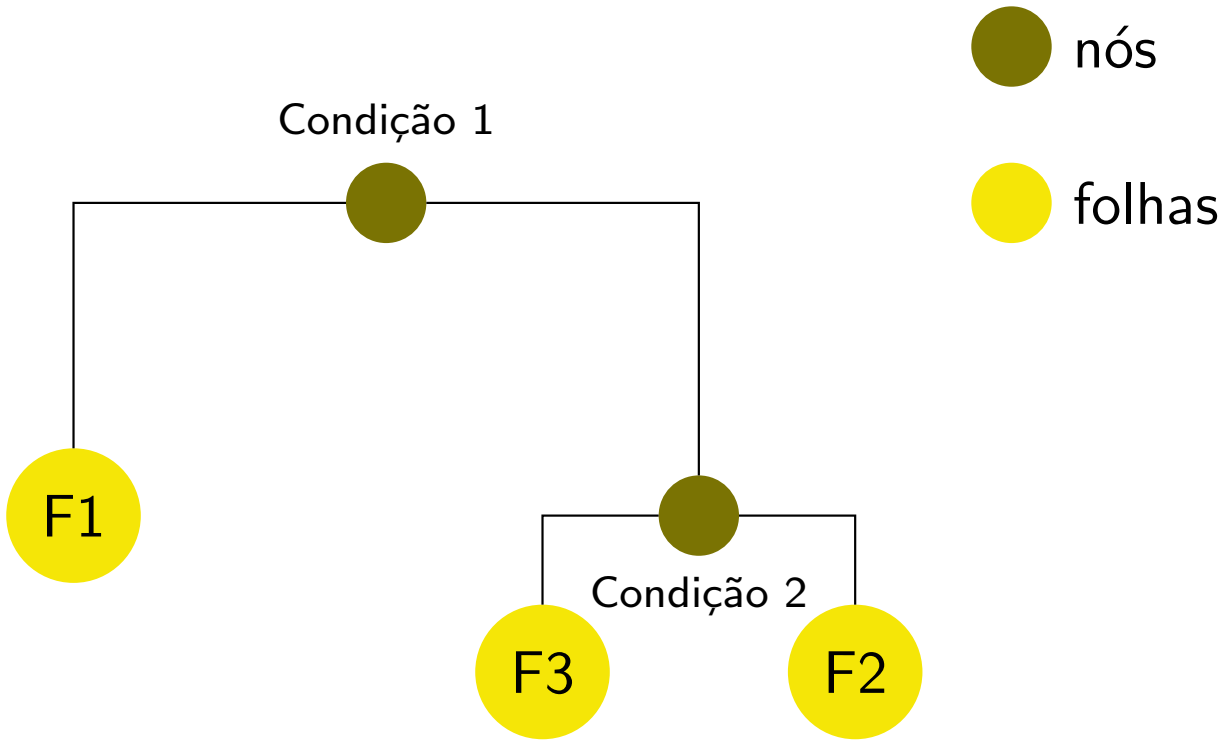


Figure: Exemplo de estrutura de uma árvore.

Criação da estrutura

1. Expansão da árvore: partições no conjunto de treino até que a condição de parada seja satisfeita **(caso base)**;

2. Poda da árvore: reagrumento dos subconjuntos da partição, com finalidade de evitar o overfitting.

Propriedades da expansão da árvore

- As partições são feitas de com cortes ortoganais que maximiza a pureza das sub-regiões resultantes;
- A pureza é o quão homogêneo são os rótulos naquela região;
- O melhor corte é o que mais tem o maior **ganho de informação**, isto é, o quanto se ganha em "pureza" ao se dividir um conjunto segundo um atributo;
- O atributo escolhido depende do tipo de modelo (regressão ou classificação).

Regression and Classification trees

Uma árvore cria uma **repartição** do espaço das covariáveis em **regiões distintas e disjuntas**: $R_1, ..., R_j$. A predição para a resposta Y de uma observação com covariáveis x que estão em R_k é dada por

$$g(x) = \frac{1}{\mid \{i : x_i \in R_k\} \mid} \sum_{i:x_i \in R_k} y_i \tag{1}$$

O cálculo feito para prever o valor da resposta de x é a **média** dos valores da variável resposta da amostra do conjunto de treinamento pertencentes àquela mesma região.
Em contraste com o modelo de regressão, a única mudança que o modelo de classificação terá, será que o cálculo para prever o valor da resposta de x é a **moda** dos valores da variável resposta.

$$g(x) = moda\{y_i : x_i \in R_k\} \tag{2}$$

Atributos para regression trees

A avaliação do quão razoável uma dada árvore T é através do seu erro quadrático médio,

$$P(T) = \sum_{R} \sum_{i:x_i \in R} \frac{(y_i - \hat{y}_R)^2}{n} \tag{3}$$

Porém, é computacionamente inviável calcular todas as possibilidades. Assim, utiliza-se uma heurística de **divisão binária recursiva**

1. Seleciona o preditor X_j e o ponto de corte s de modo a dividir o espaço do preditor nas regiões $\{X \mid X_j < s\}$ e $\{X \mid X_j \geq s\}$, buscando o valor de j e s que minimiza a equação

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \tag{4}$$

2. A melhor divisão feita considera apenas a **etapa específica**, sem se preocupar com uma futura divisão que levará a uma árvore melhor e o processo é repetido recursivamente até o **caso base**.

Exemplificação gráfica

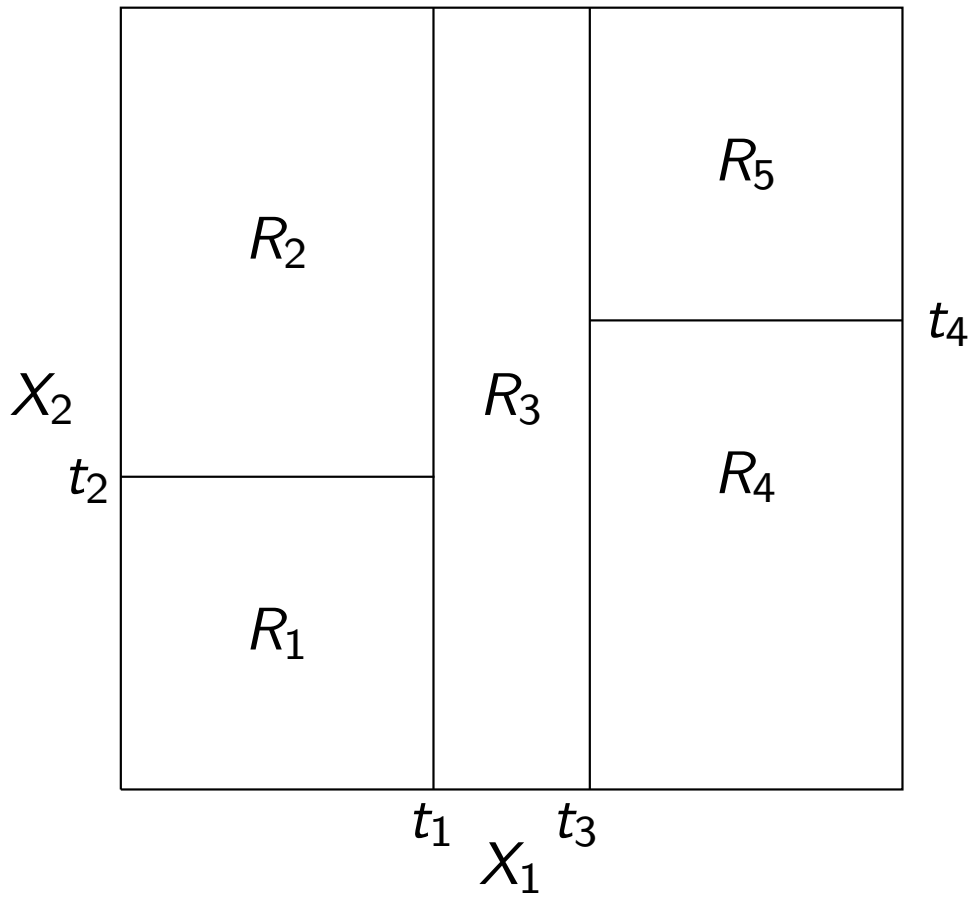


Figure: Partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some fake data.

Exemplificação gráfica

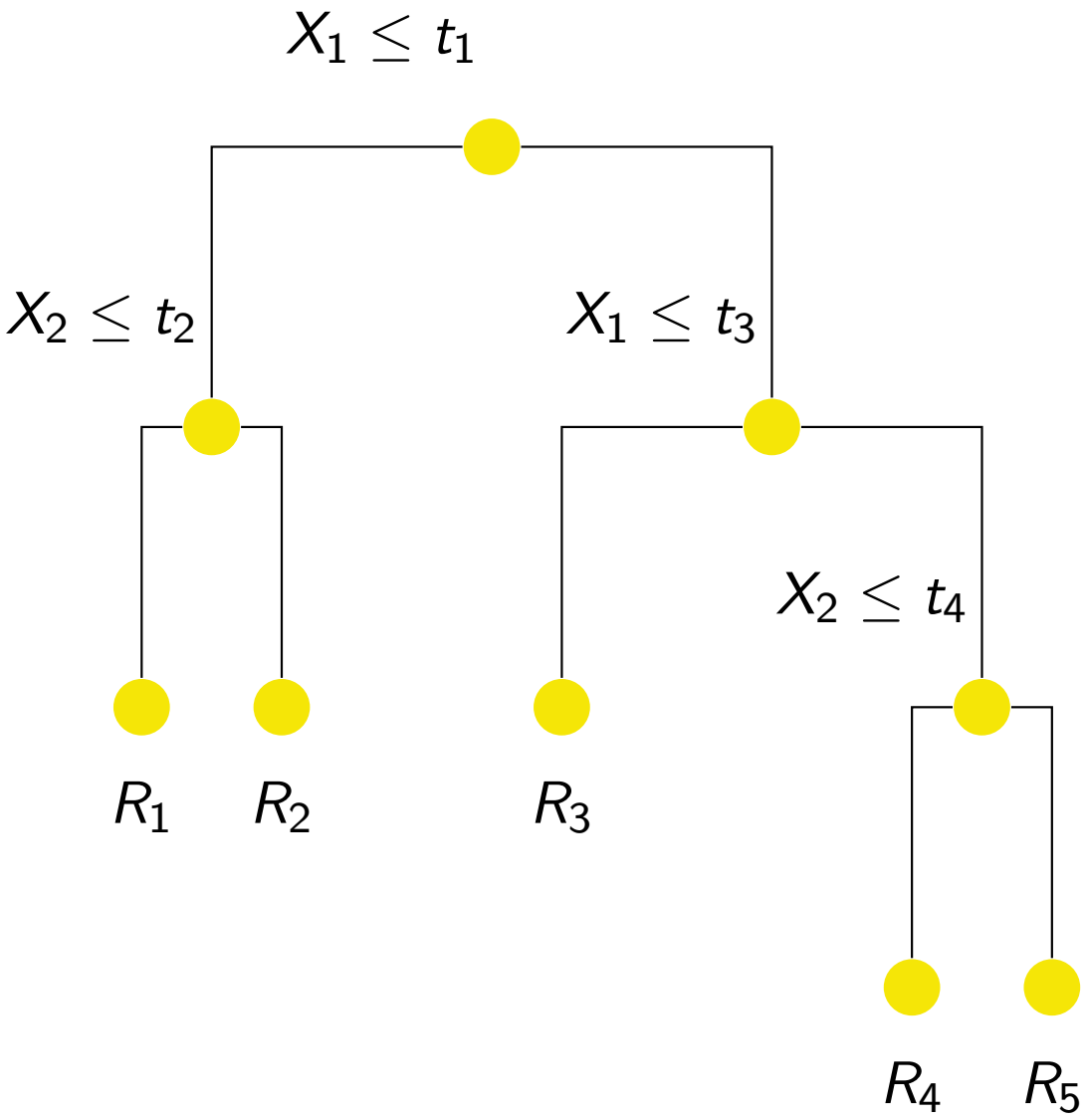


Figure: Tree corresponding to the partition in the previous slide.

Atributos para classification trees

Um dos principais atributos utilizados para definir o corte a ser feito em um modelo de classificação é a **Entropia de Shannon**, que mede o grau de inversibilidade (G_i) de um sistema. Na figura abaixo, o primeiro sistema tem um G_i grande, visto que a quantidade de informação necessária para descrever com precisão qual a cor da bolinha a ser retirada é zero.

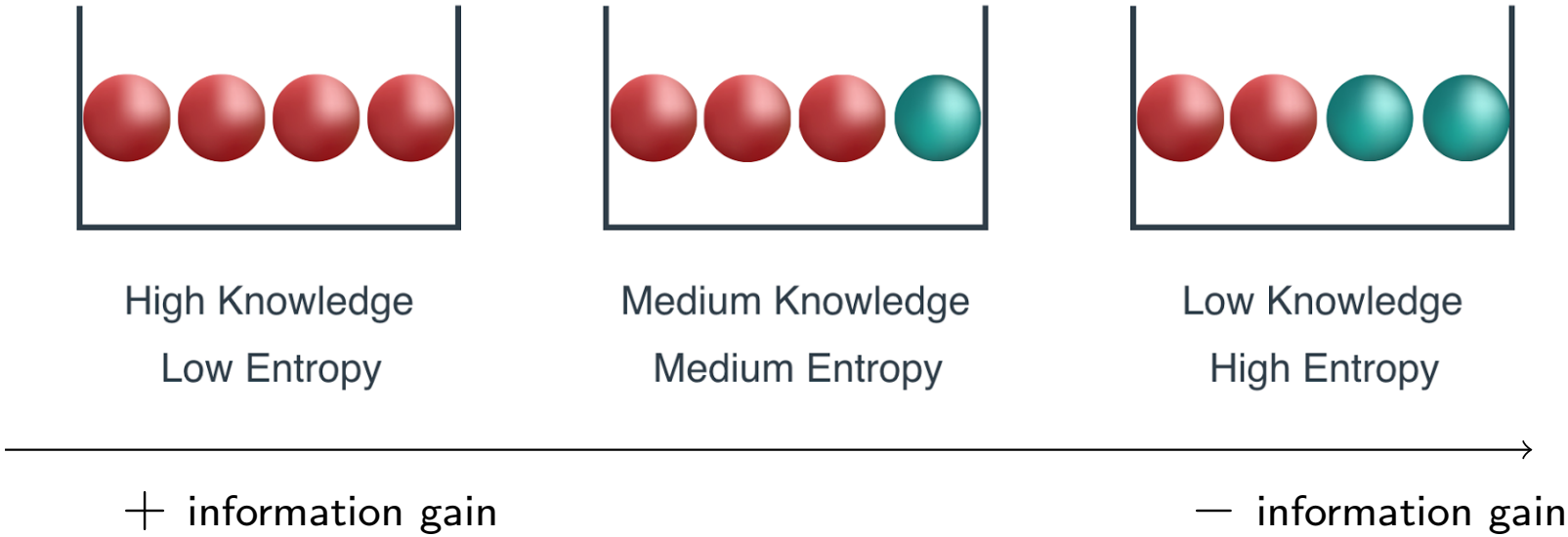


Figure: Representação da entropia

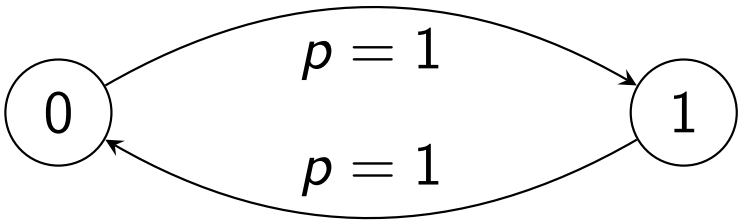
Entropia de Shannon

A equação matemática que define entropia é

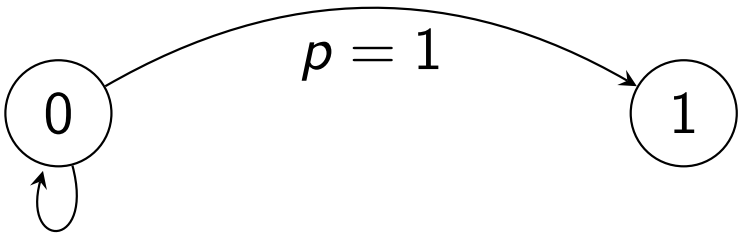
$$Entropy = -\sum_{i=1}^{n} p_i \log 2 p_i, \tag{5}$$

onde há n classes, e p_i é a probabilidade de um objeto da classe i "aparecer". Usamos $\log 2$ pois estamos tratando, intrisicamente, de *bits*.

Exemplo



$$H(S_1) = -(1 \log_2 1 + 1 \log_2 1) = 0$$



$$H(S_2) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

Gini Impurity

Outro atributo para cortes é o de Gini. Para entender, suponha o seguinte:

1. Escolhemos aleatoriamente um ponto de dados em nosso conjunto de dados e, em seguida,
2. Classificamos aleatoriamente de acordo com a distribuição da classe no conjunto de dado;
3. A probabilidade dessas observações terem classificação diferentes, é a impureza de Gini.
   O índice de Gini é definido por

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}), \tag{6}$$

onde $\hat{p}_{mk}$ é a proporção de observações classificadas como sendo da categoria m que caem na região K.

Exemplo de Gini Impurity

Event	Probability
Pick Blue, Classify Blue ✓	25%
Pick Blue, Classify Green ✖	25%
Pick Green, Classify Green ✖	25%
Pick Green, Classify Blue ✓	25%

Table: Descrição probabilística de um dataset fictício.

$$
\begin{aligned}
G &= p(1) \times (1 - p(1)) + p(2) \times (1 - p(2)) \\
&= 0.5 \times (1 - 0.5) + 0.5 \times (1 - 0.5) \\
&= \mathbf{0.5}
\end{aligned}
$$

Análise dos atributos de impureza

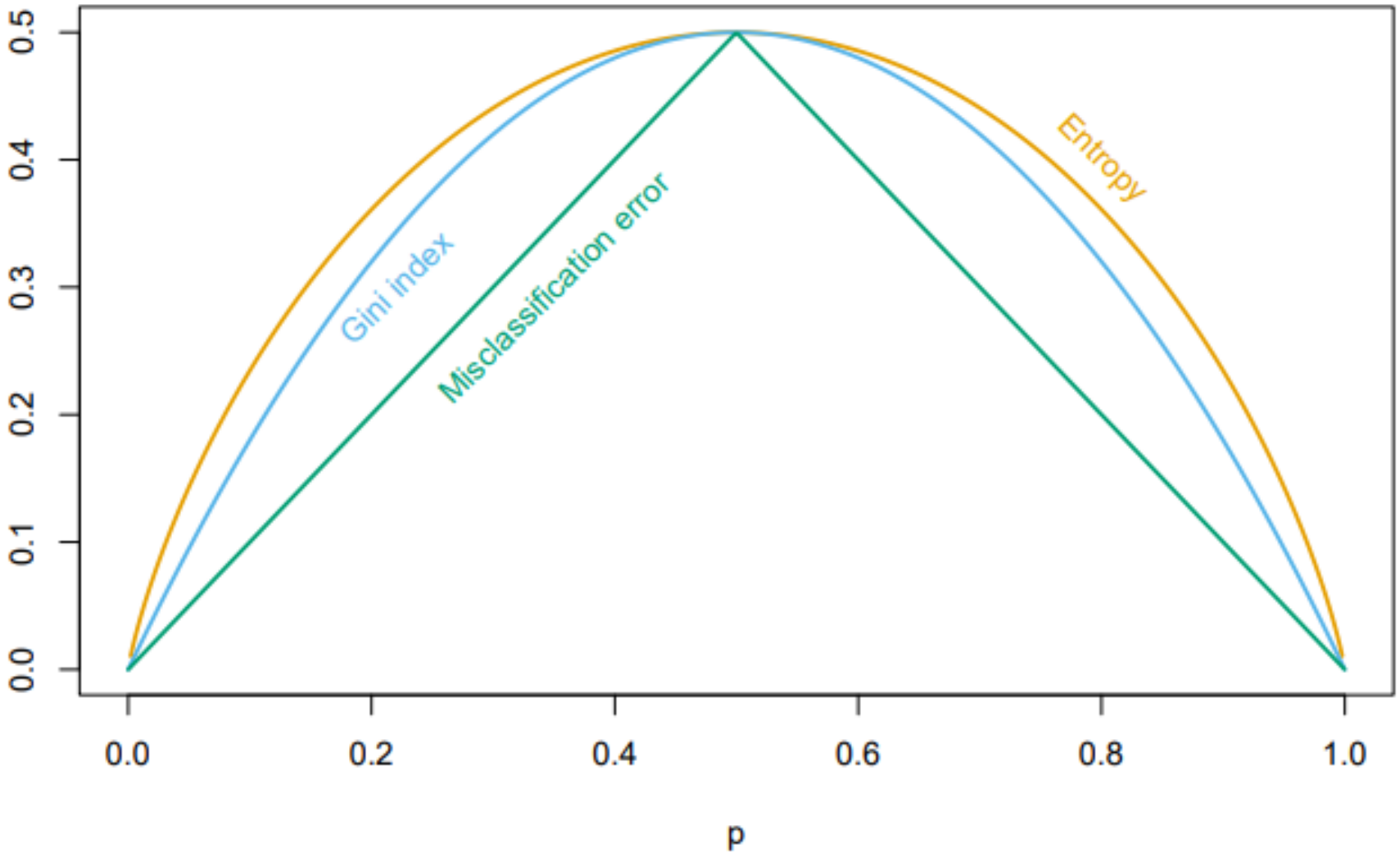
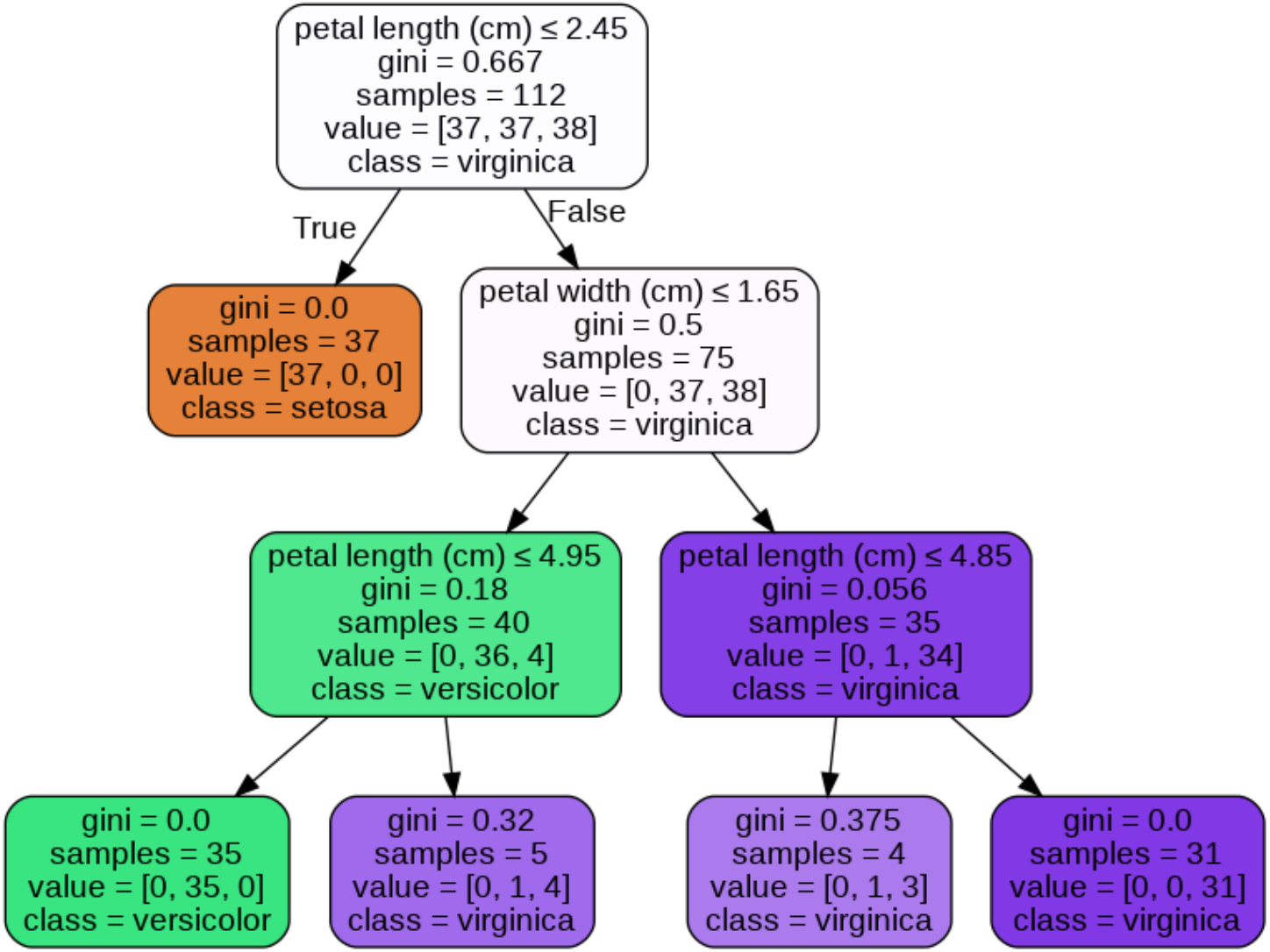


Figure: Medida de impureza para uma caso de classificação binária

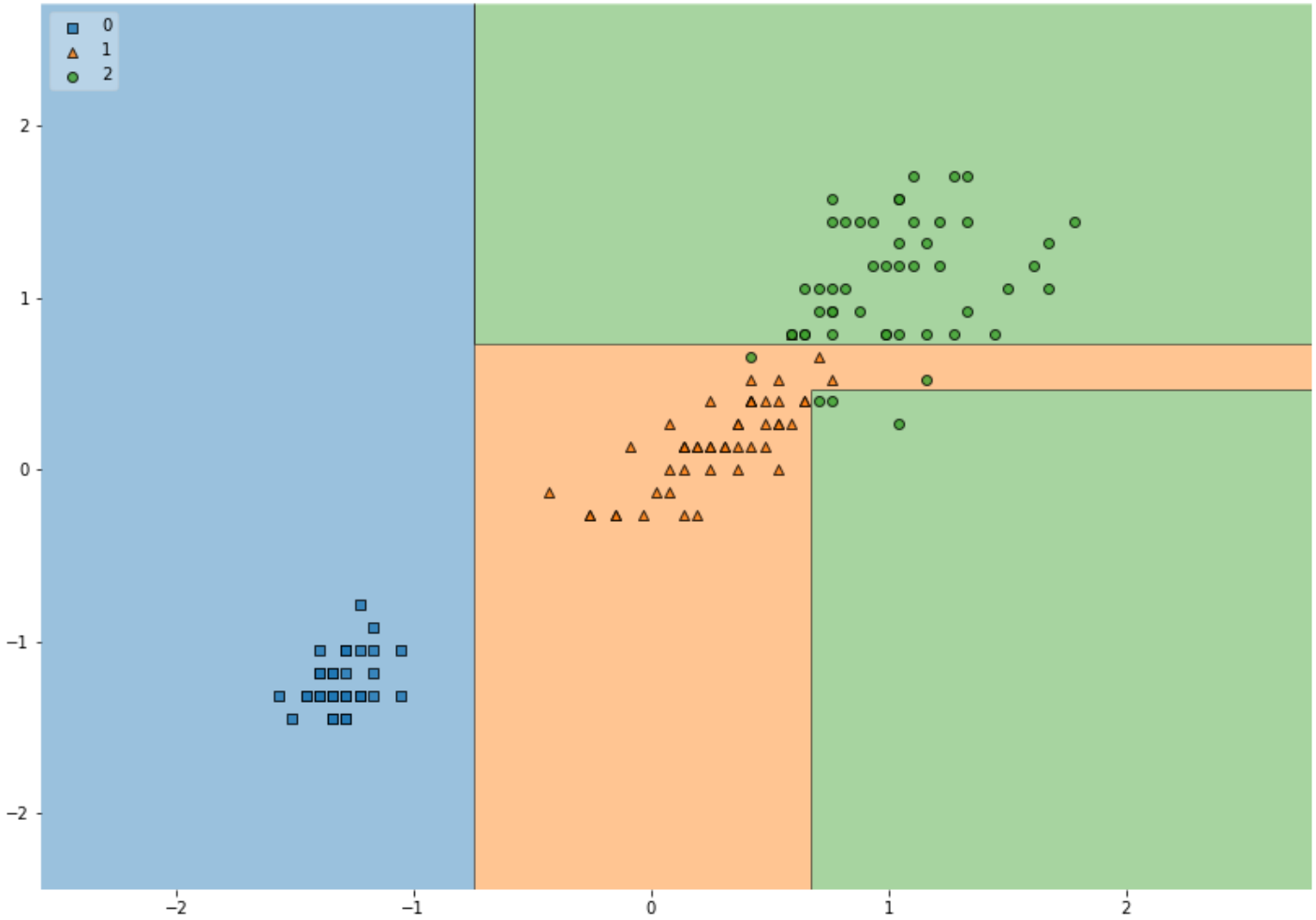
Aplicação em Python

```
# Load do dataset iris
data = load_iris()
X = data.data
y = data.target

# Separar em dados de treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                   random_state=36, test_size=0.25)

# Instanciar, treinar e ver a acurácia do modelo
clf_gini = DecisionTreeClassifier(min_samples_leaf=3,
                        criterion='gini', random_state=34)

clf_gini.fit(X_train, y_train)
y_pred_clfg = clf_gini.predict(X_test)
print("Acc with Gini:" ,accuracy_score(y_test, y_pred_clfg))
```

Tree


petal length (cm) ≤ 2.45
gini = 0.667
samples = 112
value = [37, 37, 38]
class = virginica
True
False
gini = 0.0
samples = 37
value = [37, 0, 0]
class = setosa
petal width (cm) ≤ 1.65
gini = 0.5
samples = 75
value = [0, 37, 38]
class = virginica
petal length (cm) ≤ 4.95
gini = 0.18
samples = 40
value = [0, 36, 4]
class = versicolor
petal length (cm) ≤ 4.85
gini = 0.056
samples = 35
value = [0, 1, 34]
class = virginica
gini = 0.0
samples = 35
value = [0, 35, 0]
class = versicolor
gini = 0.32
samples = 5
value = [0, 1, 4]
class = virginica
gini = 0.375
samples = 4
value = [0, 1, 3]
class = virginica
gini = 0.0
samples = 31
value = [0, 0, 31]
class = virginica

Decision Boundary



Frame Title