



**INSTITUTO
FEDERAL**

Norte de Minas Gerais

Introdução a Sistemas Inteligentes

Modelo Árvore de Decisão

Prof^a. Suzana Mota



Modelos de Machine Learning

Um modelo é uma representação matemática encapsulada em formato computacional que captura padrões a partir de um conjunto de dados.

Essa representação é utilizada para realizar previsões, classificações ou tomar decisões com base em novos dados



Modelos de Machine Learning



Análise de Risco de Crédito: Qual o limite que você pode possuir no cartão de crédito?

Recomendação de Produtos: Comprou este item? Então você vai se interessar por este outro

Previsão de Demanda: Quantos itens devo comprar para não deixar faltar no estoque?

Perfil de Cliente: Qual a probabilidade deste perfil de cliente comprar tal coisa? Ou cancelar a assinatura?

Detecção de Fraude: Este cliente é um potencial fraudador?

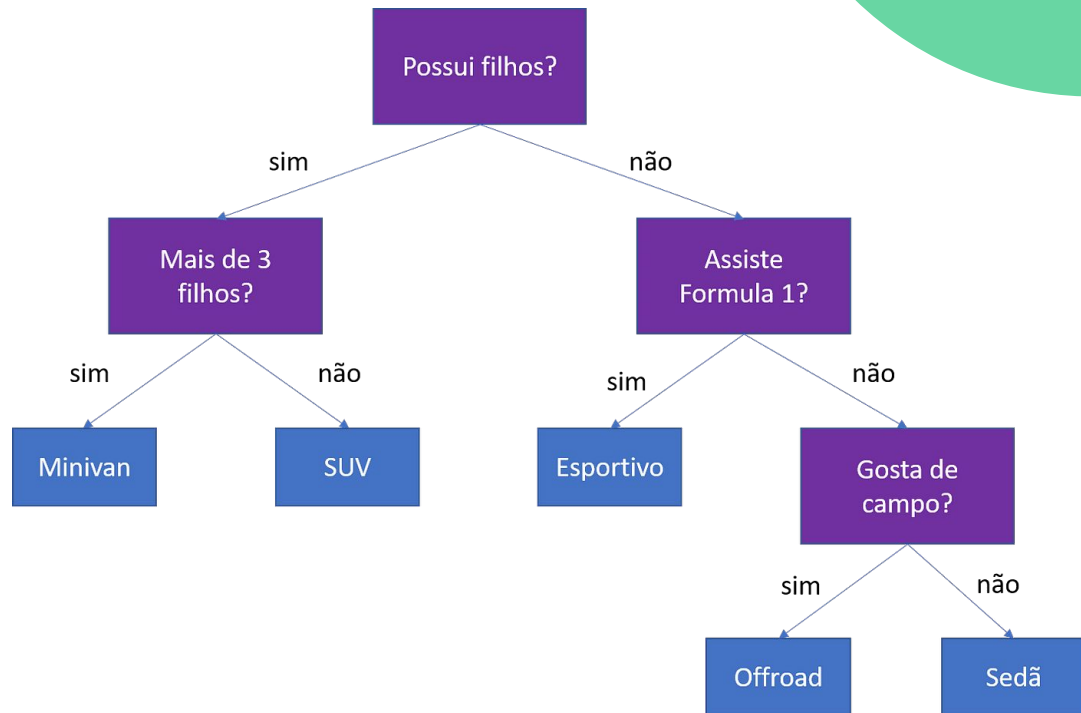
Árvores de Decisão

Decision Tree



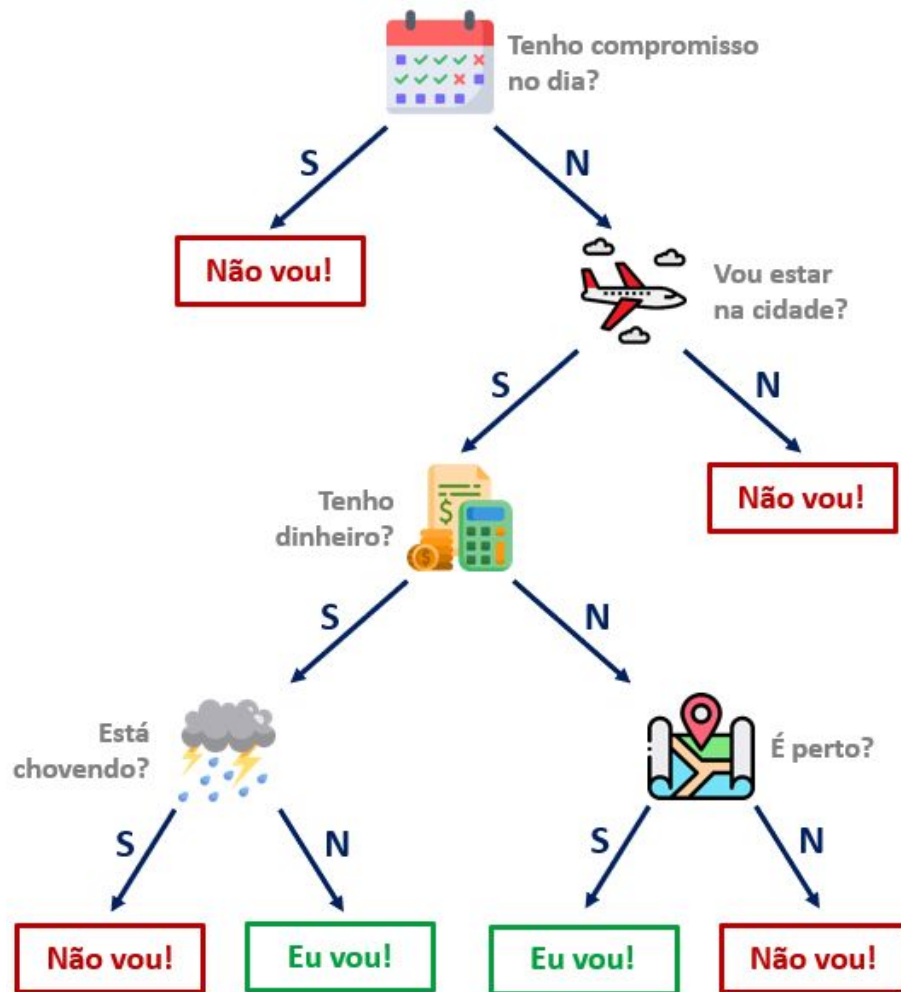
Árvore de Decisão

Uma árvore de decisão funciona como um conjunto de regras hierárquicas, onde as **decisões** são feitas em "**nós**" e as **saídas** são dadas nas "**folhas**".



Árvore de Decisão

Devo ir naquela festa de Halloween?



Como funciona

Divisão Recursiva dos Dados: A árvore é construída ao dividir repetidamente os dados em subconjuntos com base nas características que melhor separam as classes.

Nós Internos: Cada nó interno representa uma pergunta ou condição sobre uma característica (por exemplo, "Possui filhos?").

Ramos: Cada ramo a partir de um nó é uma possível resposta à pergunta (por exemplo, "Sim" ou "Não").

Folhas: As folhas representam a previsão final de classe ou valor.

Vantagens

Interpretabilidade: Fácil de visualizar e entender.

Não Requer Normalização de Dados: Pode lidar com dados sem a necessidade de normalizá-los

Resistente a Dados Faltantes: Pode funcionar bem mesmo com alguns dados ausentes.

Desvantagens

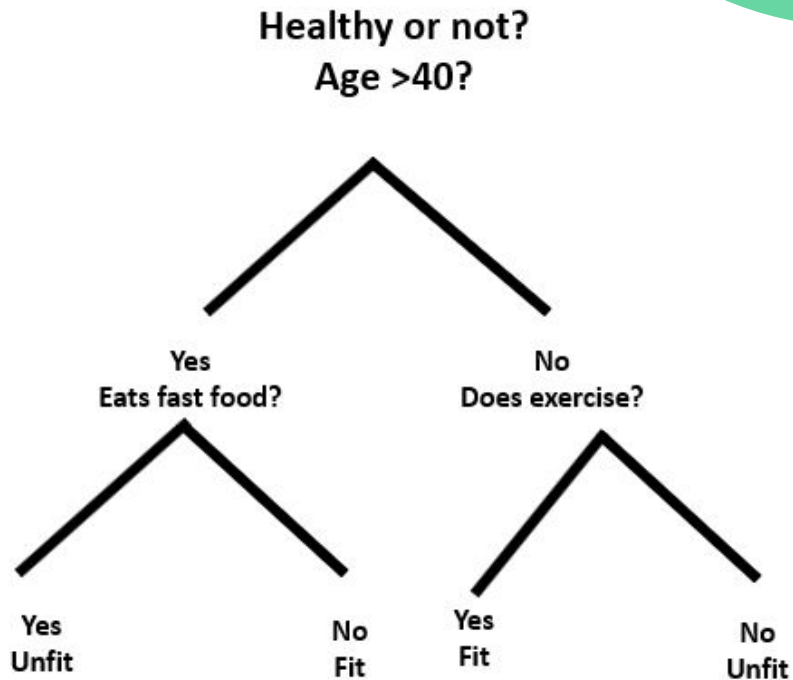
Propensão ao Overfitting: Árvores muito profundas podem se ajustar excessivamente aos dados de treino.

Suscetível a Pequenas Variações nos Dados: Pequenas mudanças nos dados podem gerar uma árvore completamente diferente.

Interpretabilidade

As árvores de decisão são fáceis de visualizar, permitindo que os usuários entendam com bastante clareza como as predições do modelo são feitas.

As árvores podem mostrar quais características são mais importantes para a decisão, ajudando na interpretação dos dados.



Etapas de Treinamento

Configurações de ambiente

- ❑ Instalar Bibliotecas
- ❑ Importar Bibliotecas

Conhecimento dos Dados

- ❑ Ler dados
- ❑ Fazer análise exploratória para conhecer os dados

Pré-Processamento

- ❑ Tratar dados ausentes
- ❑ Separar labels e features(x e y)
- ❑ Discretizar dados
- ❑ Dividir dados: treinamento e teste

Treinar Modelo

- ❑ Definir Modelo de Machine Learning a ser usado
- ❑ Treinar modelo (.fit)
- ❑ Obter métricas

Avaliar Modelo

- ❑ Oferecer dados de testes e observar as previsões do modelo
- ❑ Obter métricas
- ❑ Avaliar Modelo



Treinando o Modelo

<https://github.com/suzanasvm/SistemasInteligentes/tree/main/datasets>

Utilize um dos datasets e faça o treinamento do Modelo, usando

imports

Configurações
de ambiente

```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import accuracy_score
```

```
from sklearn.preprocessing import LabelEncoder
```

```
#biblioteca para criarmos o modelo
```

```
from sklearn.tree import DecisionTreeClassifier
```

Lendo os dados

Conhecimento
dos Dados

```
url_dados =
```

```
'https://raw.githubusercontent.com/suzanasvm/SistemasInteligentes/refs/heads/main/datasets/iris/iris.csv'
```

```
df = pd.read_csv(url_dados)
```

```
df.set_index('Id', inplace=True)
```

Conhecendo os dados

Conhecimento
dos Dados

Mostra o Tamanho do Dataframe

```
df.shape
```

Mostra as 5 primeiras linhas

```
df.head()
```

Mostra uma visão estatística dos dados

```
df.describe()
```

Conta os valores nulos em todas as colunas

```
df.isnull().sum()
```

Tratar dados ausentes: aula Dados Ausentes

Separar labels e Features (X e y):

```
X = df.drop('Species', axis=1)
```

```
y = df['Species']
```

Discretizar Dados

```
label_encoder = LabelEncoder()
```

```
y = label_encoder.fit_transform(y)
```

Dividir Dados: Treinamento e Teste

```
X_train, X_test, y_train, y_test = train_test_split(X, y,  
test_size=0.3, random_state=42)
```

Instanciando o modelo:

```
decision_tree = DecisionTreeClassifier(random_state=42)
```

Treinar o modelo:

```
decision_tree_model = decision_tree.fit(X_train, y_train)
```

Qual a taxa de acerto do modelo:

```
acuracia_treinamento = decision_tree.score(X_train, y_train)  
print(f"Acurácia do modelo nos dados de treinamento:  
{acuracia_treinamento * 100:.2f}%")
```


Obtendo predições nos dados de Teste

```
predito_ArvoreDecisao = decision_tree.predict(X_test)
predito_ArvoreDecisao
```

Qual a taxa de acerto do modelo com esses dados novos (dados de teste):

```
acuracia_teste = accuracy_score(y_test,
predito_ArvoreDecisao)
print(f"Acurácia do modelo de Árvores de Decisão:
{acuracia_teste * 100:.2f}%")
```

Quais features foram mais importantes?

```
importances = decision_tree.feature_importances_  
feature_names = X.columns  
importances_df = pd.DataFrame(importances,  
index=feature_names,  
columns=['Importance']).sort_values(by='Importance',  
ascending=False)  
print("\nImportância das características:")  
print(importances_df)
```

Vamos plotar a árvore de Decisão

```
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
import numpy as np

# Definir o tamanho da figura
plt.figure(figsize=(25, 20))

# Plotar a árvore de decisão
plot_tree(decision_tree,
          feature_names=X.columns,
          class_names=[ 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica' ],
          filled=True,
          rounded=True,
          fontsize=12)

plt.title('Árvore de Decisão')

colors = [ '#ff9999', '#66b3ff', '#99ff99' ]
class_names = [ 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica' ]
handles = [plt.Line2D([0], [0], marker='o', color='w', markerfacecolor=color, markersize=10) for color in
           colors]

plt.legend(handles, class_names, title= "Classes", loc="upper left", fontsize=12)

plt.show()

)
```

Interpretabilidade do Modelo

BONUS:
Interpretabilidade do
Modelo

Árvore de Decisão

Classes
● Iris-setosa
● Iris-versicolor
● Iris-virginica

