UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
ELE0606 - TÓPICOS ESPECIAIS EM IA

Docente:
José Alfredo Ferreira Costa

Autor:
Vinícius Venceslau Venancio da Penha

Árvore de Decisão

Natal - RN
Setembro de 2023

Sumário

1. Introdução

Árvores de Decisão, um algoritmo proeminente em aprendizado de máquina supervisionado, são amplamente empregadas em tarefas de classificação e regressão. A sua representação gráfica é um valioso instrumento em análises de dados e decisões em diversos setores, como medicina, finanças e engenharia. Este relatório apresenta os princípios fundamentais, o funcionamento, as vantagens e desvantagens das Árvores de Decisão, bem como a relação desse algoritmo com o desenvolvimento aplicado.

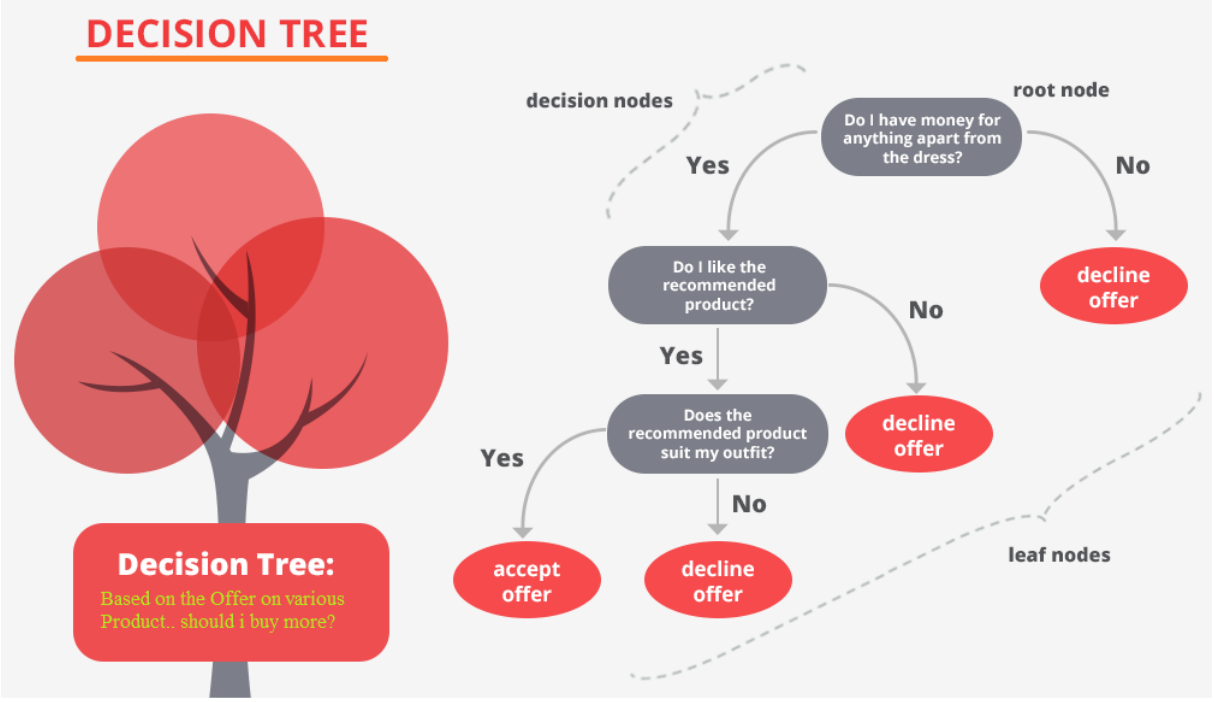


Figura 1: Imagem ilustrativa do funcionamento do método Árvore de Decisão.

2. Princípios Fundamentais das Árvores de Decisão

Uma Árvore de Decisão é uma estrutura hierárquica que desempenha um papel crucial na classificação de informações, assemelhando-se a um roteiro de seleções. Essa estrutura orienta a máquina a tomar decisões com base em características específicas dos dados em análise. Cada ponto de interseção na árvore, denominado "nó", representa um ponto de decisão ou teste, muitas vezes relacionado a uma característica particular. Por outro lado, os "nós folha" constituem os desfechos finais da árvore, indicando as classes ou valores de saída associados às decisões tomadas ao longo do percurso da árvore.

Dessa maneira, a construção de uma Árvore de Decisão implica um processo de segmentação dos dados de treinamento em subconjuntos, com o objetivo de otimizar a pureza das classes atribuídas nos "nós folha". Isso significa que, à medida que se avança na árvore, busca-se criar grupos mais homogêneos de dados, o que resulta em um aprimoramento da capacidade da árvore em fazer previsões ou classificações mais precisas.

Esse procedimento é fundamental para garantir a eficácia do modelo de Árvore de Decisão na tomada de decisões com base em características complexas dos dados.

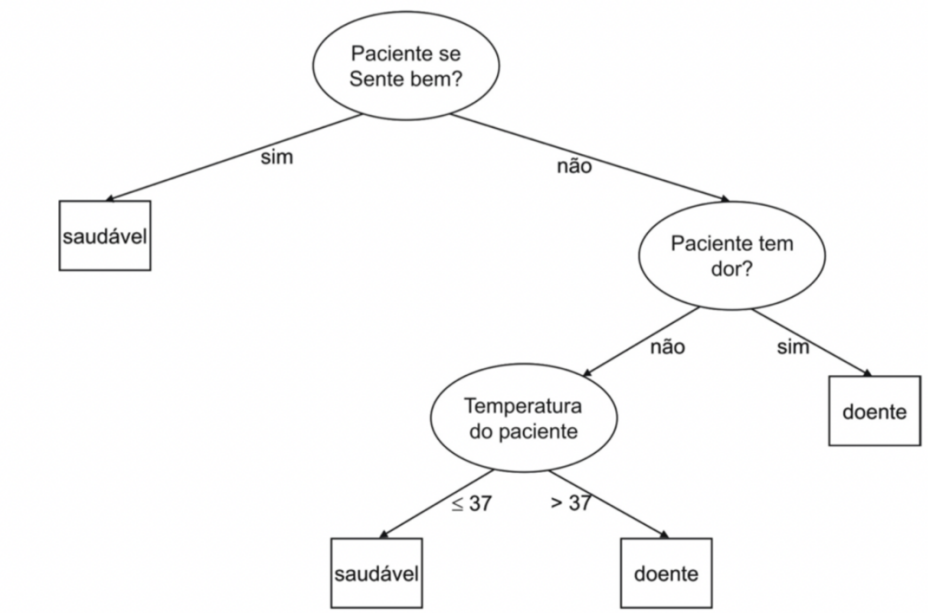


Figura 2: Exemplo de Árvore de Decisão de Classificação.

2.1 Funcionamento das Árvores de Decisão

O funcionamento das Árvores de Decisão compreende as seguintes etapas:

- **Seleção do Atributo:** Inicia-se com a escolha do atributo que melhor divide os dados de treinamento, baseando-se em medidas de impureza, tais como entropia ou índice Gini, calculadas para cada atributo.

- **Segmentação dos Dados:** Os dados são segmentados de acordo com o valor do atributo selecionado, originando subconjuntos.

- **Criação de Ramos:** A Árvore cresce à medida que ramos são criados para cada subconjunto, seguindo o processo recursivo.

- **Critérios de Parada:** A construção da árvore prossegue até que critérios de parada sejam satisfeitos, tais como profundidade máxima ou número mínimo de amostras em um nó.

- **Classificação e Regressão:** Para classificação, o valor da classe mais frequente nos nós folha é atribuído como previsão. Em regressão, a média dos valores alvo nos nós folha é empregada como previsão.

2.2 Vantagens das Árvores de Decisão

- **Interpretabilidade:** Árvores de Decisão são notáveis pela alta interpretabilidade, permitindo compreender o processo decisório de forma clara.

- **Manuseio de Dados Diversificados:** Elas lidam eficazmente com variáveis numéricas e categóricas, sem necessidade de pré-processamento complexo.

- **Identificação de Características Relevantes:** Árvores podem discernir automaticamente quais características são determinantes para classificação ou regressão.

2.3 Desvantagens das Árvores de Decisão

- **Suscetibilidade ao Overfitting:** Existe a possibilidade de super ajustamento (overfitting), em que a Árvore se ajusta excessivamente aos dados de treinamento, prejudicando o desempenho em dados não observados.

- **Sensibilidade a Flutuações nos Dados**: Pequenas variações nos dados de treinamento podem resultar em Árvores distintas, tornando-as menos robustas.

- **Limitações na Representação de Relações Complexas:** Árvores podem não ser ideais para capturar relações complexas entre características.

2.4 Relação com o Algoritmo Desenvolvido

O algoritmo desenvolvido segue os princípios das Árvores de Decisão, construindo uma hierarquia de regras de decisão com base nas características dos dados de entrada. Este modelo de Árvore de Decisão é empregado para classificação e decisões, criando uma representação visual das regras e possibilitando previsões com base nessa estrutura.

Em síntese, as Árvores de Decisão constituem um elemento central no algoritmo desenvolvido, possibilitando decisões com base nas características dos dados de forma transparente e eficaz. Contudo, é crucial compreender as suas vantagens e desvantagens para uma utilização adequada e para mitigar desafios como o overfitting ou a representação limitada de relações complexas.

2.5 Comparação entre o Método KNN e o Método Árvore de decisão

Pontos de Convergência:

- **Acurácia como Métrica de Avaliação:** Ambos os métodos usam a acurácia como métrica para avaliar o desempenho do modelo. A acurácia mede a proporção de previsões corretas em relação ao total de previsões.

- **Supervisionados:** Tanto Árvore de Decisão quanto KNN são algoritmos de aprendizado supervisionado, o que significa que eles exigem rótulos de classe conhecidos para treinamento.

- **Flexibilidade:** Ambos os métodos são versáteis e podem ser usados para problemas de classificação em diversas áreas, desde reconhecimento de padrões até análise de dados.

Pontos de Divergência:

- **Modelo de Decisão vs. Modelo Baseado em Vizinhos:** A maior divergência está na abordagem fundamental. A Árvore de Decisão cria um modelo hierárquico de regras de decisão com base nos atributos, enquanto o KNN é um modelo de aprendizado baseado em instâncias, que faz previsões com base nas instâncias mais próximas.

- **Interpretabilidade:** Árvores de Decisão são altamente interpretáveis, pois as regras de decisão podem ser visualizadas facilmente. KNN, por outro lado, é menos interpretável, pois as previsões são feitas com base em vizinhos mais próximos, sem regras explícitas.

- **Sensibilidade à Escala:** KNN é sensível à escala dos atributos, enquanto Árvores de Decisão não são. Portanto, é comum normalizar ou escalar os atributos ao usar KNN.

- **Complexidade do Modelo:** Árvores de Decisão podem criar modelos complexos com muitas regras, o que pode levar ao overfitting (ajuste excessivo) se não forem adequadamente podadas. KNN é geralmente mais simples em termos de modelo.

- **Hiperparâmetros Diferentes:** Os dois algoritmos têm hiperparâmetros diferentes que precisam ser ajustados. Por exemplo, em Árvores de Decisão, você ajusta a profundidade da árvore e o critério de divisão, enquanto em KNN, você ajusta o número de vizinhos (k).

Características Distintas:

- ➢ **Árvore de Decisão:** É uma representação de decisão hierárquica que pode ser facilmente visualizada. Pode lidar com atributos categóricos e numéricos. É menos sensível a valores ausentes e outliers. Pode sofrer de overfitting se não for podada.

- ➢ **K-Nearest Neighbors (KNN):** Faz previsões com base na proximidade dos vizinhos mais próximos. Sensível à escala dos atributos, requer normalização. Pode ser robusto para outliers, mas é computacionalmente intensivo para grandes conjuntos de dados. Pode ser afetado pelo "problema da maldição da dimensionalidade" em espaços de alta dimensão.

3. Conclusões

Em suma, as Árvores de Decisão representam uma ferramenta poderosa e versátil no domínio do aprendizado de máquina, sendo amplamente empregadas em diversas aplicações devido à sua interpretabilidade e capacidade de identificar características relevantes. No entanto, é essencial considerar suas desvantagens, como a suscetibilidade ao overfitting e a limitação na representação de relações complexas, e adotar estratégias de pré-processamento de dados e ajuste de hiperparâmetros para otimizar seu desempenho.

A relação entre o modelo de Árvore de Decisão e o algoritmo desenvolvido é evidente, uma vez que ambos compartilham os princípios fundamentais deste algoritmo, proporcionando um framework robusto para a tomada de decisões com base nas características dos dados.

Portanto, compreender e aplicar eficazmente as Árvores de Decisão é um passo valioso na construção de soluções de aprendizado de máquina capazes de realizar classificações e previsões com precisão e transparência.

4. Bibliografia

A Árvore de Decisão - Algoritmos de Aprendizado de Máquinas. Disponível em: https://www.youtube.com/watch?v=aNrdgC0lIZ8&t=2033s.

PYTHON MACHINE LEARNING (03): Criar e treinar modelo – Árvore de Decisão. Disponível em: https://www.youtube.com/watch?v=ba3_UMjhAQc.

LAPP, D. **Heart Disease Dataset**. Disponível em: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset.

arvorededecisao-15-09-2023

September 22, 2023

##**Atividade 3** - Rodar os algoritmos de árvore de decisão para as bases de dados **Wine** e **Heart Disease Dataset**

Aluno: Vinícius Venceslau Venancio da Penha

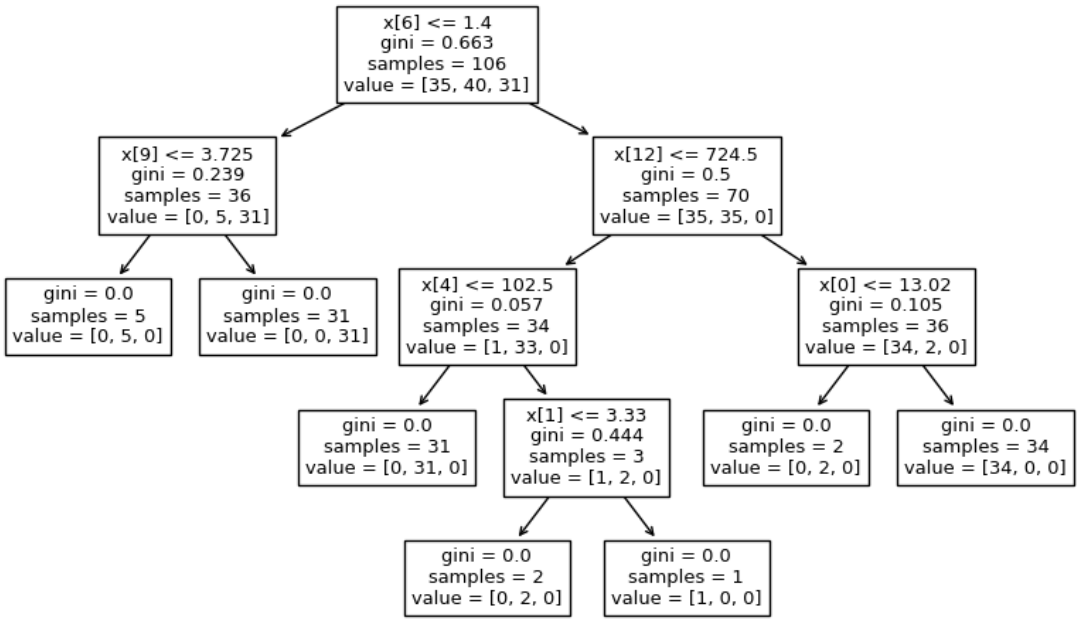
Turma: ELE0606 - Tópicos Especiais em IA

[95]:
```
#Importações de bibliotecas:
import numpy as np
import pandas as pd
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix

#Carregamento da base de dados WINE:
banco_de_dados = load_wine()
wine_df = pd.DataFrame(banco_de_dados.data, columns=banco_de_dados.
 ↪feature_names)

#Adição da coluna 'classe' ao DataFrame:
wine_df['classe'] = banco_de_dados['target']

#Armazenamento dos rótulos de classe em uma variável:
wine_classe = wine_df['classe']

#Remoção da coluna 'classe' do DataFrame, porque ela representa a variável de␣
 ↪resposta ou saída do sistema e não deve ser usada como atributo de entrada␣
 ↪para o modelo.
wine_df.drop(['classe'], axis=1, inplace=True)

#Divisão dos dados em conjuntos de treinamento e teste:
X_train, X_test, y_train, y_test = train_test_split(wine_df, wine_classe,␣
 ↪test_size=0.4, random_state=13)

#Criação do modelo de árvore de decisão:
arvore_de_decisao = DecisionTreeClassifier(random_state=13)
```

```
#Treinamento do modelo:
arvore_de_decisao.fit(X_train, y_train)

#Previsões com o modelo treinado:
y_previsao = arvore_de_decisao.predict(X_test)

#Avaliação do desempenho do modelo:
accuracy = accuracy_score(y_test, y_previsao)
conf_matrix = confusion_matrix(y_test, y_previsao)
class_report = classification_report(y_test, y_previsao,␣
 ↪target_names=banco_de_dados.target_names)

#Impressão dos resultados:
print(f'Acurácia: {accuracy:.4f}\n')
print('Matriz de Confusão:\n')
print(conf_matrix)
print('\n')
print('Relatório de Classificação:\n')
print(class_report)
print('\n')
fig, ax = plt.subplots(figsize=(10,6))
tree.plot_tree(arvore_de_decisao)
print('Visualização Esquemática:\n')
plt.show()
```

```
Acurácia: 0.9444

Matriz de Confusão:

[[22  2  0]
 [ 1 30  0]
 [ 0  1 16]]


Relatório de Classificação:

              precision    recall  f1-score   support

     class_0       0.96      0.92      0.94        24
     class_1       0.91      0.97      0.94        31
     class_2       1.00      0.94      0.97        17

    accuracy                           0.94        72
   macro avg       0.96      0.94      0.95        72
weighted avg       0.95      0.94      0.94        72
```

Visualização Esquemática:



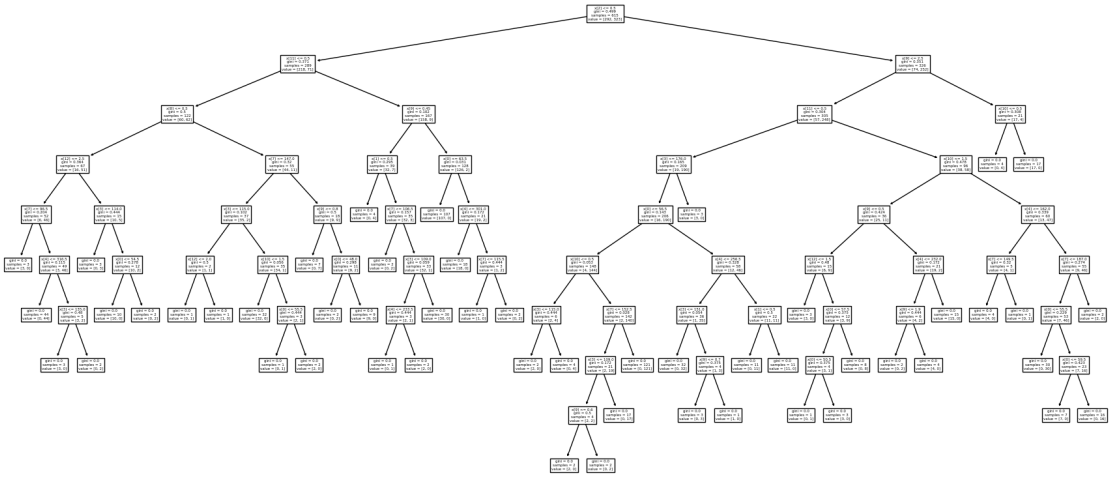
Desenvolver a árvore de decisão para a base de dados **Heart Disease Dataset**:

[96]:
```
#Permitir o google colab acessar os arquivos do Drive:
from google.colab import drive
drive.mount('/content/drive')

#Importar bibliotecas:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix

#Carregamento da base de dados de doença cardíaca a partir do Google Drive:
caminho_arquivo = "/content/drive/My Drive/heart.csv"  #Caminho correto para o␣
 ↪arquivo no Google Drive.
banco_de_dados = pd.read_csv(caminho_arquivo)
```

```
#Criar um Dataframe:
heart_df = banco_de_dados

#Armazenamento dos rótulos de classe em uma variável:
heart_classe = heart_df['target']

#Remoção da coluna 'target' do DataFrame, porque ela representa a variável de␣
 ↪resposta ou saída do sistema e não deve ser usada como atributo de entrada␣
 ↪para o modelo.
heart_df.drop(['target'], axis=1, inplace=True)

#Divisão dos dados em conjuntos de treinamento e teste:
X_train, X_test, y_train, y_test = train_test_split(heart_df, heart_classe,␣
 ↪test_size=0.4, random_state=13)

#Criação do modelo de árvore de decisão:
arvore_de_decisao = DecisionTreeClassifier(random_state=13)

#Treinamento do modelo:
arvore_de_decisao.fit(X_train, y_train)

#Previsões com o modelo treinado:
y_previsao = arvore_de_decisao.predict(X_test)

#Avaliação do desempenho do modelo:
accuracy = accuracy_score(y_test, y_previsao)
conf_matrix = confusion_matrix(y_test, y_previsao)
class_report = classification_report(y_test, y_previsao, target_names=['Classe␣
 ↪0', 'Classe 1'])

#Impressão dos resultados:
print('\n')
print(f'Acurácia: {accuracy:.4f}\n')
print('Matriz de Confusão:\n')
print(conf_matrix)
print('\n')
print('Relatório de Classificação:\n')
print(class_report)

#Visualização gráfica:
print('\n')
fig, ax = plt.subplots(figsize=(22,10))
tree.plot_tree(arvore_de_decisao)
print('Visualização Esquemática:\n')
plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

```
drive.mount("/content/drive", force_remount=True).


Acurácia: 0.9707

Matriz de Confusão:

[[198   9]
 [  3 200]]


Relatório de Classificação:

              precision    recall  f1-score   support

    Classe 0       0.99      0.96      0.97       207
    Classe 1       0.96      0.99      0.97       203

    accuracy                           0.97       410
   macro avg       0.97      0.97      0.97       410
weighted avg       0.97      0.97      0.97       410


Visualização Esquemática:
```



Referências:

A Árvore de Decisão - Algoritmos de Aprendizado de Máquinas. Disponível em: https://www.youtube.com/watch?v=aNrdgC0lIZ8&t=2033s.

PYTHON MACHINE LEARNING (03): Criar e treinar modelo – Árvore de Decisão. Disponível em: https://www.youtube.com/watch?v=ba3_UMjhAQc.

LAPP, D. **Heart Disease Dataset**. Disponível em: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset.