

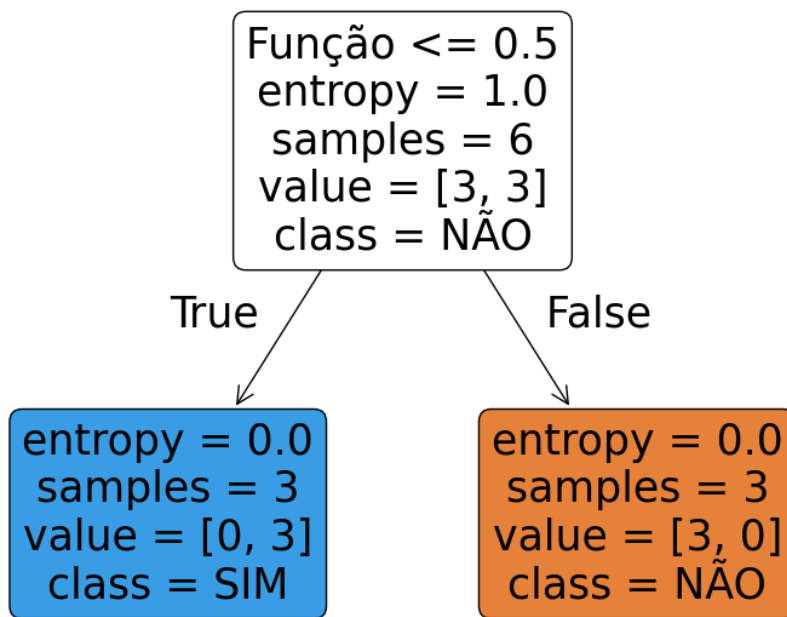
Trabalho Prático de IA - Árvore de Decisão

Aluno: Pedro Henrique Vilaça Valverde

Disciplina: Inteligência Artificial

Exercício 1: Decisão de Aceitar Estágio

a) Árvore de Decisão Gerada:



b) Regras SE-ENTÃO Extraídas:

```
|--- Função <= 0.50
|   |--- class: SIM
|--- Função > 0.50
|   |--- class: NÃO
```

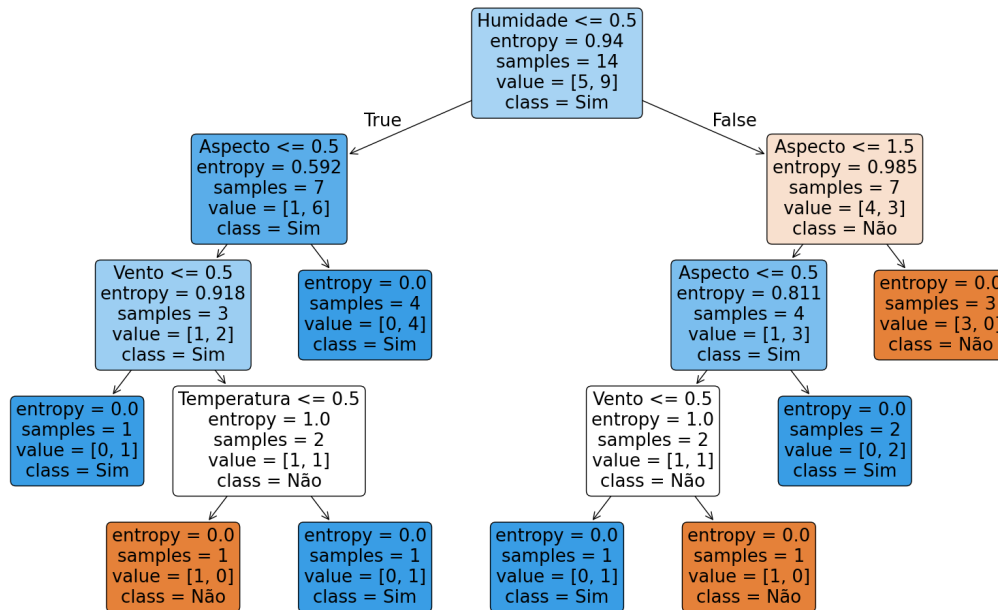
c) e d) Classificações de Novos Casos:

Previsão para (Salário=baixo, Localização=perto, Função=interessante): SIM

Previsão para (Salário=alto, Localização=perto, Função=desinteressante): NÃO

Exercício 2: Decisão de Jogar Tênis

a) Árvore de Decisão Gerada:



b) Regras SE-ENTÃO Extraídas:

```
|--- Humidade <= 0.50
|   |--- Aspecto <= 0.50
|   |   |--- Vento <= 0.50
|   |   |   |--- class: Sim
|   |   |   |--- Vento > 0.50
|   |   |       |--- Temperatura <= 0.50
|   |   |       |   |--- class: Não
|   |   |       |   |--- Temperatura > 0.50
|   |   |       |       |--- class: Sim
|   |   |--- Aspecto > 0.50
|   |       |--- class: Sim
|--- Humidade > 0.50
|   |--- Aspecto <= 1.50
|   |   |--- Aspecto <= 0.50
|   |   |   |--- Vento <= 0.50
|   |   |   |   |--- class: Sim
|   |   |   |   |--- Vento > 0.50
|   |   |   |       |--- class: Não
|   |   |   |--- Aspecto > 0.50
|   |   |       |--- class: Sim
|   |--- Aspecto > 1.50
|       |--- class: Não
```

c) e d) Classificações de Novos Casos:

Previsão para (Aspecto=Sol, Temp.=Ameno, Humidade=Normal, Vento=Forte): Sim

Previsão para (Aspecto=Chuva, Temp.=Quente, Humidade=Normal, Vento=Fraco): Sim

Anexo: Código Fonte (resolver_problema.py)

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier, plot_tree, export_text

# --- CONFIGURAÇÕES ---
RESULT_DIR = './result'

# --- CRIAÇÃO DO DIRETÓRIO DE RESULTADOS ---
if not os.path.exists(RESULT_DIR):
    os.makedirs(RESULT_DIR)
print(f"Diretório '{RESULT_DIR}' pronto para receber os resultados.")

# --- FUNÇÃO PARA RESOLVER O EXERCÍCIO 1: ACEITA ESTÁGIO ---
def resolver_exercicio_1():
    """
    Resolve o primeiro exercício sobre a decisão de aceitar um estágio.
    """
    print("\n--- Iniciando Exercício 1: Classificação de Estágio ---")

    # 1. Entrada de Dados
    X1 = np.array([
        ['alto', 'longe', 'interessante'],
        ['baixo', 'perto', 'desinteressante'],
        ['baixo', 'longe', 'interessante'],
        ['alto', 'longe', 'desinteressante'],
        ['alto', 'perto', 'interessante'],
        ['baixo', 'longe', 'desinteressante']
    ])
    Y = np.array(['SIM', 'NÃO', 'SIM', 'NÃO', 'SIM', 'NÃO'])

    # 2. Pré-processamento: Converter dados categóricos para numéricos
    X = X1.copy()
    mapeamento = {
        'Salário': {'alto': 0, 'baixo': 1},
        'Localização': {'longe': 0, 'perto': 1},
        'Função': {'interessante': 0, 'desinteressante': 1}
    }

    for i in range(len(X1)):
        X[i, 0] = mapeamento['Salário'][X1[i, 0]]
        X[i, 1] = mapeamento['Localização'][X1[i, 1]]
        X[i, 2] = mapeamento['Função'][X1[i, 2]]

    previsores_nomes = ['Salário', 'Localização', 'Função']
    XX = pd.DataFrame(X, dtype=int, columns=previsores_nomes)
    YY = pd.DataFrame(Y, dtype=str, columns=['Decisão'])

    # 3. Criar e Treinar o Classificador de Árvore de Decisão
    arvore = DecisionTreeClassifier(criterion='entropy')
    arvore = arvore.fit(XX, YY)

    acuracia = arvore.score(XX, YY)
    print(f"Acurácia do modelo nos dados de treino: {acuracia*100:.2f}%")

    # 4. Gerar e Salvar os Resultados
```

```

# a) Construir a árvore (Salvar como imagem)
fig, ax = plt.subplots(figsize=(10, 8))
plot_tree(
    arvore,
    feature_names=previsores_nomes,
    class_names=arvore.classes_,
    filled=True,
    rounded=True,
    fontsize=25 # Adicionamos este parâmetro para diminuir a fonte
)
caminho_imagem = os.path.join(RESULT_DIR, 'ex1_arvore.png')
plt.savefig(caminho_imagem)
plt.close(fig)
print(f"Imagem da árvore salva em: {caminho_imagem}")

# b) Indicar a regra SE-ENTÃO (Salvar em .txt)
regras = export_text(arvore, feature_names=previsores_nomes)
caminho_regras = os.path.join(RESULT_DIR, 'ex1_regras.txt')
with open(caminho_regras, 'w', encoding='utf-8') as f:
    f.write(regras)
print(f"Regras SE-ENTÃO salvas em: {caminho_regras}")

# c) e d) Fazer as classificações pedidas
previsoes_texto = []

# c) Classificar Salário baixo, Localização perto, Função interessante
dados_c = [[mapeamento['Salário']]['baixo'], mapeamento['Localização']]['perto'],
mapeamento['Função']]['interessante']]
previsao_c = arvore.predict(dados_c)
resultado_c = f"Previsão para (Salário=baixo, Localização=perto, Função=interessante): {previsao_c[0]}"
previsoes_texto.append(resultado_c)
print(resultado_c)

# d) Classificar Salário alto, Localização perto, Função desinteressante
dados_d = [[mapeamento['Salário']]['alto'], mapeamento['Localização']]['perto'],
mapeamento['Função']]['desinteressante']]
previsao_d = arvore.predict(dados_d)
resultado_d = f"Previsão para (Salário=alto, Localização=perto, Função=desinteressante): {previsao_d[0]}"
previsoes_texto.append(resultado_d)
print(resultado_d)

caminho_previsoes = os.path.join(RESULT_DIR, 'ex1_previsoes.txt')
with open(caminho_previsoes, 'w', encoding='utf-8') as f:
    f.write("\n".join(previsoes_texto))
print(f"Previsões salvas em: {caminho_previsoes}")
print("--- Exercício 1 Finalizado ---")

# --- FUNÇÃO PARA RESOLVER O EXERCÍCIO 2: JOGAR TÊNIS ---
def resolver_exercicio_2():
    """
    Resolve o segundo exercício sobre a decisão de jogar tênis.
    """
    print("\n--- Iniciando Exercício 2: Classificação de Jogar Tênis ---")

    # 1. Entrada de Dados
    X1 = np.array([
        ['Sol', 'Quente', 'Elevada', 'Fraco'], #D1
        ['Sol', 'Quente', 'Elevada', 'Forte'], #D2
        ['Nuvens', 'Quente', 'Elevada', 'Fraco'], #D3
        ['Chuva', 'Ameno', 'Elevada', 'Fraco'], #D4
    ])

```



```

# c) Classificar Aspecto Sol, Temp. Ameno, Humidade Normal, Vento Forte
dados_c = [[mapeamento['Aspecto']['Sol'], mapeamento['Temp']['Ameno'], mapeamento['Humidade']['Normal'],
mapeamento['Vento']['Forte']]]
previsao_c = arvore.predict(dados_c)
resultado_c = f"Previsão para (Aspecto=Sol, Temp.=Ameno, Humidade=Normal, Vento=Forte): {previsao_c[0]}"
previsoes_texto.append(resultado_c)
print(resultado_c)

# d) Classificar Aspecto Chuva, Temp. Quente, Humidade=Normal, Vento=Fraco
dados_d = [[mapeamento['Aspecto']['Chuva'], mapeamento['Temp']['Quente'], mapeamento['Humidade']['Normal'],
mapeamento['Vento']['Fraco']]]
previsao_d = arvore.predict(dados_d)
resultado_d = f"Previsão para (Aspecto=Chuva, Temp.=Quente, Humidade=Normal, Vento=Fraco): {previsao_d[0]}"
previsoes_texto.append(resultado_d)
print(resultado_d)

caminho_previsoes = os.path.join(RESULT_DIR, 'ex2_previsoes.txt')
with open(caminho_previsoes, 'w', encoding='utf-8') as f:
    f.write("\n".join(previsoes_texto))
print(f"Previsões salvas em: {caminho_previsoes}")
print("--- Exercício 2 Finalizado ---")

# --- EXECUÇÃO PRINCIPAL ---
if __name__ == "__main__":
    resolver_exercicio_1()
    resolver_exercicio_2()
    print("\nProcesso finalizado. Todos os arquivos de resultado foram salvos na pasta 'result/'.")

```