Lista #2
Disciplina: Inteligência Artificial
Professora: Cristiane Neri Nobre
Aluno: Arthur Henrique Tristão Pinto

Questão 01

1) A árvore de decisão é gerada a partir da análise de uma base de dados rotulada. Os atributos da base são avaliados de acordo com sua relevância para a classificação, e, a partir disso, vão sendo construídos os níveis da árvore, com caminhos organizados em ordem de importância. As folhas representam as regras finais. O atributo que ocupa a raiz da árvore é o mais significativo, responsável pelo maior ganho de informação e, consequentemente, pela melhor separação dos dados.

2) Ao gerar uma árvore de decisão a partir de uma base de dados, é possível classificar novos dados com base nos atributos e regras da árvore, bem como prever valores numéricos no caso de regressão. Além disso, a árvore pode ser analisada para extrair regras de decisão e identificar a importância relativa de cada atributo no processo.

3) As principais vantagens de um algoritmo de árvore de decisão são: facilidade de interpretação, eficiência com complexidade de tempo linear, suporte tanto a dados numéricos quanto categóricos e robustez frente a atributos redundantes ou irrelevantes, já que os atributos são selecionados de acordo com seu ganho de informação.Por outro lado, as desvantagens das árvores de decisão incluem: sensibilidade a pequenas mudanças nos dados, o que pode gerar grandes variações na árvore; dificuldade em lidar com atributos contínuos, já que a ordenação desses valores pode consumir até 70% do tempo de indução em grandes conjuntos de dados; e a necessidade de mecanismos especiais para tratar valores ausentes.

4) A qualidade de uma árvore de decisão é avaliada pela capacidade de classificar corretamente novos dados, para isso se avaliam as métricas como precisão, recall,acurácia e F1 Score.

5) Tendo a árvore de decisão, para obter as regras, basta olhar os nós folhas que foram gerados, pois cada um representa uma regra de decisão final.

Questão 02)

Código em python utilizado para o cálculo de entropia e ganho com base nas fórmulas de Shanon e Quinlan:

- Eq 1. Ganho(Atributo) = Entropia(Classe) - Entropia(Atributo)
- Eq 2. Entropia(S) = $\sum_{i=1}^{c} = \quad - pi\ log_2\ pi$

```
import pandas as pd
import numpy as np

# Carregar a tabela de dados do restaurante
tabela_dados = pd.read_csv('restaurante.csv', sep=';')

# Função para calcular a entropia de uma série de dados
# Essa função utiliza a equação 2 para calculo de entropia(Shannon)
def calcular_entropia(serie_dados):
    proporcoes = serie_dados.value_counts(normalize=True)
    # O 1e-9 é para evitar log de zero
    return -sum(proporcoes * np.log2(proporcoes + 1e-9))

# Função para calcular a entropia condicional de um atributo em relação
à classe
def          calcular_entropia_atributo(tabela,          nome_atributo,
nome_classe='Conclusao'):
    num_total_registros = len(tabela)
    entropia_ponderada = 0

    for valor in tabela[nome_atributo].unique():
        sub_tabela = tabela[tabela[nome_atributo] == valor]
        peso = len(sub_tabela) / num_total_registros
                                    entropia_ponderada    +=    peso    *
calcular_entropia(sub_tabela[nome_classe])

    return entropia_ponderada


# 1° Nível: Calcular a entropia e o ganho de informação para a raiz
entropia_total = calcular_entropia(tabela_dados['Conclusao'])
print(f"Entropia Total: {entropia_total:.4f}\n")

atributos = [col for col in tabela_dados.columns if col != 'Conclusao']
resultados = {}
```

```
for nome_atributo in atributos:
        entropia_atributo = calcular_entropia_atributo(tabela_dados, nome_atributo, 'Conclusao')
    ganho = entropia_total - entropia_atributo
        resultados[nome_atributo] = {'entropia': entropia_atributo, 'ganho': ganho}

resultados_df = pd.DataFrame(resultados).T
resultados_df = resultados_df.sort_values(by='ganho', ascending=False)

print("Entropia e Ganho de Informação para cada atributo (RAIZ):")
print(resultados_df.to_string(float_format="%.4f"))

# Identificar e exibir a raiz
raiz = resultados_df.index[0]
print(f"\n>>>> A raiz da árvore de decisão é '{raiz}', pois tem o maior ganho de informação. <<<<\n")

# 2° Nível: Calcular a entropia e o ganho para os nós filhos
valores_raiz = tabela_dados[raiz].unique()
print(f"Calculando o 2° nível a partir da raiz '{raiz}':\n")

for valor_raiz in valores_raiz:
    print(f"---- Subconjunto para '{raiz}' = '{valor_raiz}' ----")
    sub_tabela = tabela_dados[tabela_dados[raiz] == valor_raiz].copy()

    # Calcular a entropia do subconjunto
    entropia_subconjunto = calcular_entropia(sub_tabela['Conclusao'])
    print(f"Entropia do subconjunto: {entropia_subconjunto:.4f}\n")

     # Calcular entropia e ganho para os atributos restantes
     # Utiliza da equação de ganho Eq 1.
     atributos_restantes = [col for col in sub_tabela.columns if col not in [raiz, 'Conclusao']]
    resultados_nivel_2 = {}
    for nome_atributo in atributos_restantes:
            entropia_nivel_2 = calcular_entropia_atributo(sub_tabela, nome_atributo, 'Conclusao')
        ganho_nivel_2 = entropia_subconjunto - entropia_nivel_2
                    resultados_nivel_2[nome_atributo] = {'entropia': entropia_nivel_2, 'ganho': ganho_nivel_2}

    resultados_nivel_2_df = pd.DataFrame(resultados_nivel_2).T
```

```
                                        resultados_nivel_2_df          =
resultados_nivel_2_df.sort_values(by='ganho', ascending=False)
                                        resultados_nivel_2_df          =
resultados_nivel_2_df[~resultados_nivel_2_df.index.duplicated(keep='fir
st')]

        print("Entropia  e  Ganho  de  Informação  para  os  atributos
restantes:")
    print(resultados_nivel_2_df.to_string(float_format="%.4f"))
    print("-" * 40)
```

Resultado do código

1) - Raiz da Árvore

Entropia Total(CLASSE): 1.0000

ATRIBUTO	ENTROPIA	GANHO
Cliente	0.4591	0.5409
Tempo	0.7925	0.2075
Fome	0.8043	0.1957
Preço	0.8043	0.1957
Sex/Sab	0.9793	0.0207
Chuva	0.9793	0.0207
Res	0.9793	0.0207
Tipo	1.0000	0.0000
Bar	1.0000	0.0000
Alternativo	1.0000	0.0000

A raiz da árvore de decisão é 'Cliente', pois tem o maior ganho de informação.

2) - Segundo nível da árvore

Caso o atributo cliente seja “Alguns” ou “Nenhum” o dado é classificado,respectivamente, como Sim e Não, mas para o conjunto de Cliente “Cheio” temos o seguinte resultado:
Entropia Total(Cliente Cheio): 0.9183

ATRIBUTO	ENTROPIA	GANHO
Res	0.6667	0.2516
Tipo	0.6667	0.2516
Preço	0.6667	0.2516
Tempo	0.6667	0.2516
Fome	0.6667	0.2516
Alternativo	0.8091	0.1092
SexSab	0.8091	0.1092
Chuva	0.8742	0.0441
Bar	0.9183	0.0000

Com isso, no segundo nível temos um empate de ganho entre 5 atributos, e devemos escolher de preferência os atributos binários. Sendo assim, o atributo do segundo nível é Fome

Árvore gerada até o segundo nível