Lista 3

Inteligência Artificial

Nome: Vitor de Meira Gomes
Matrícula: 800643

Questão 1:

1.1) Retirei a coluna de arquivo de ***cols_label_encode*** e fiz separadamente

(Link do Colab)

```
#para codificar todos os atributos para laberEncoder de uma única vez
#base_encoded = base.apply(LabelEncoder().fit_transform)
cols_label_encode = ['Alternativo', 'Bar', 'SexSab','fome', 'Preco', 'Chuva', 'Res','Tempo']
base[cols_label_encode] = base[cols_label_encode].apply(LabelEncoder().fit_transform)
```

✓ 0.0s

```
base
```

✓ 0.0s

	Alternativo	Bar	SexSab	fome	Cliente	Preco	Chuva	Res	Tipo	Tempo	Conclusao
0	1	0	0	1	Alguns	2	0	1	Frances	0	Sim
1	1	0	0	1	Cheio	0	0	0	Tailandes	2	Nao
2	0	1	0	0	Alguns	0	0	0	Hamburger	0	Sim
3	1	0	1	1	Cheio	0	1	0	Tailandes	1	Sim
4	1	0	1	0	Cheio	2	0	1	Frances	3	Nao
5	0	1	0	1	Alguns	1	1	1	Italiano	0	Sim
6	0	1	0	0	Nenhum	0	1	0	Hamburger	0	Nao
7	0	0	0	1	Alguns	1	1	1	Tailandes	0	Sim
8	0	1	1	0	Cheio	0	1	0	Hamburger	3	Nao
9	1	1	1	1	Cheio	2	0	1	Italiano	1	Nao
10	0	0	0	0	Nenhum	0	0	0	Tailandes	0	Nao
11	1	1	1	1	Cheio	0	0	0	Hamburger	2	Sim

```
mapa = {"Nenhum": 0, "Alguns": 1, "Cheio": 2}

base["Cliente"] = base["Cliente"].map(mapa)

base
```

✓ 0.0s

	Alternativo	Bar	SexSab	fome	Cliente	Preco	Chuva	Res	Tipo	Tempo	Conclusao
0	1	0	0	1	1	2	0	1	Frances	0	Sim
1	1	0	0	1	2	0	0	0	Tailandes	2	Nao
2	0	1	0	0	1	0	0	0	Hamburger	0	Sim
3	1	0	1	1	2	0	1	0	Tailandes	1	Sim
4	1	0	1	0	2	2	0	1	Frances	3	Nao
5	0	1	0	1	1	1	1	1	Italiano	0	Sim
6	0	1	0	0	0	0	1	0	Hamburger	0	Nao
7	0	0	0	1	1	1	1	1	Tailandes	0	Sim
8	0	1	1	0	2	0	1	0	Hamburger	3	Nao
9	1	1	1	1	2	2	0	1	Italiano	1	Nao
10	0	0	0	0	0	0	0	0	Tailandes	0	Nao
11	1	1	1	1	2	0	0	0	Hamburger	2	Sim

Questão 2:

Código utilizado para 2.1, 2.2 e 2.3 (Link)

2.1)

Primeiramente, eu carreguei a base de dados do Titanic e analisei suas dimensões, atributos e valores ausentes. Entre os principais atributos, considerei Sex, Pclass, Age, SibSp, Parch, Fare e Embarked, além da variável alvo Survived.

Para entender a relação entre esses atributos e a sobrevivência, eu utilizei agrupamentos e cálculos de proporção. Ao cruzar `Survived` com `Sex`, percebi uma diferença muito clara entre homens e mulheres. Também analisei `Pclass` e constatei que passageiros da 1ª classe tinham mais chance de sobreviver do que os da 3ª.

No caso da idade, como era uma variável contínua, eu agrupei os valores em faixas etárias (crianças, adolescentes, adultos jovens, adultos e idosos). Isso me permitiu perceber que crianças tiveram mais chance de sobrevivência, enquanto idosos estavam entre os mais vulneráveis. Além disso, eu considerei o porto de embarque (`Embarked`), que também mostrou padrões diferentes conforme a origem dos passageiros.

Com esses passos, eu consegui visualizar os principais padrões da base e entender como diferentes características se relacionam com a sobrevivência.

2.2)

Depois da exploração, eu precisei preparar os dados para a árvore de decisão. Para isso, eu converti as variáveis categóricas em valores numéricos: Sex passou a ser 0 para masculino e 1 para feminino, e Embarked passou a ser 0 para Southampton, 1 para Cherbourg e 2 para Queenstown.

Eu também tratei os valores ausentes. Para Age e Fare, usei a mediana; para Embarked, preenchi com a moda (porto mais frequente). Dessa forma, eliminei os dados faltantes que poderiam prejudicar o modelo.

Além disso, eu removi colunas sem relevância direta para a predição, como Name, Ticket, Cabin e PassengerId, já que não contribuem para a classificação ou tinham informação excessivamente específica.

Com essas transformações, eu deixei o dataset apenas com atributos numéricos e sem valores nulos, pronto para treinar a árvore.

2.3)

Com o dataset tratado, eu treinei uma árvore de decisão para identificar os fatores mais importantes na determinação da sobrevivência. O primeiro atributo usado pela árvore foi o sexo, mostrando que as mulheres tiveram muito mais chance de sobreviver do que os homens.

Entre as mulheres, o fator mais relevante foi a classe da passagem (Pclass): passageiras da 1ª e 2ª classe apresentaram as maiores chances, enquanto as da 3ª tiveram mais dificuldade.

Entre os homens, a idade aparece como decisiva: crianças pequenas tiveram alguma chance de sobrevivência, mas homens adultos e idosos, principalmente da 3ª classe, tiveram chances muito baixas. Em alguns ramos, também apareceu o atributo Fare, indicando que tarifas mais altas estavam associadas a maiores chances de sobreviver.

Com isso, eu pude concluir que os padrões de mortalidade extraídos pela árvore confirmam o relato histórico do desastre: prevaleceu a regra de “mulheres e crianças primeiro”, seguida pela vantagem dos passageiros de classes mais altas, enquanto homens adultos, em especial da 3ª classe, foram os mais afetados.

Questão 3:

3.1)

As principais diferenças entre os algoritmos ID3 e C4.5 são:

- Tratamento de Dados Contínuos: O algoritmo ID3 não consegue lidar eficientemente com atributos contínuos, enquanto o C4.5 pode dividir atributos contínuos em intervalos, permitindo o uso de dados numéricos.
- Tratamento de Valores Ausentes: O C4.5 pode lidar com valores ausentes de forma mais eficaz, o que não é uma capacidade do ID3.
- Podas e Previsão Pessimista: C4.5 implementa técnicas de poda após a criação da árvore, tornando-a menos propensa ao overfitting. ID3, por outro lado, não possui uma estratégia de poda tão robusta.
- Cálculo do Ganho de Informação: C4.5 melhora o cálculo do ganho de informação do ID3 introduzindo a razão de ganho, que ajuda a evitar o viés em atributos com muitos valores, um ponto fraco do ID3.

3.2)

O algoritmo C4.5 lida com atributos de entrada que são numéricos dividindo esses atributos em intervalos. Durante o processo de construção da árvore, ele analisa os dados contínuos e determina os pontos de divisão que melhor segregam as classes, permitindo que a árvore utilize informações de atributos numéricos de maneira eficaz.

4)

Letra C

5)

Letra A

6)

	Precisão	Recall	F1Score	TVP	TFN	TFP	TVN
A	0.588235	0.588235	0.588235	0.588235	0.411765	0.066667	0.933333
B	0.652174	0.833333	0.731707	0.833333	0.166667	0.076923	0.923077
C	0.769231	0.666667	0.714286	0.666667	0.333333	0.065217	0.934783
D	0.892857	0.877193	0.884956	0.877193	0.122807	0.092308	0.907692

Feito com o seguinte código

```python
import numpy as np
import pandas as pd

cm = np.array([
    [10, 4, 2, 1],  #A
    [1, 15, 2, 0],  #B
    [2, 3, 20, 5],  #C
    [4, 1, 2, 50]   #D
])

classes = ["A","B","C","D"]
metrics = {}

total = cm.sum()

for i, label in enumerate(classes):
    TP = cm[i, i]
    FN = cm[i, :].sum() - TP
    FP = cm[:, i].sum() - TP
    TN = total - (TP + FN + FP)


    precisao = TP / (TP + FP) if (TP+FP) > 0 else 0
    recall   = TP / (TP + FN) if (TP+FN) > 0 else 0
    f1       = 2 * precisao * recall / (precisao + recall) if (precisao+recall) > 0 else 0
    tvp      = recall
    tfn      = FN / (TP + FN) if (TP+FN) > 0 else 0
    tfp      = FP / (FP + TN) if (FP+TN) > 0 else 0
    tvn      = TN / (FP + TN) if (FP+TN) > 0 else 0

    metrics[label] = [precisao, recall, f1, tvp, tfn, tfp, tvn]

df_metrics = pd.DataFrame(metrics,
                          index=["Precisão","Recall","F1Score","TVP","TFN","TFP","TVN"]).T

#? df_percent = (df_metrics * 100).round(2)

print(df_metrics)
```