LIsta de IA #3

806454 - Yago Almeida Melo

Questão 1)

O índice de Gini mede a impureza de um nó em uma árvore de decisão. Ele indica quão misturadas estão as classes dentro de um conjunto de dados.

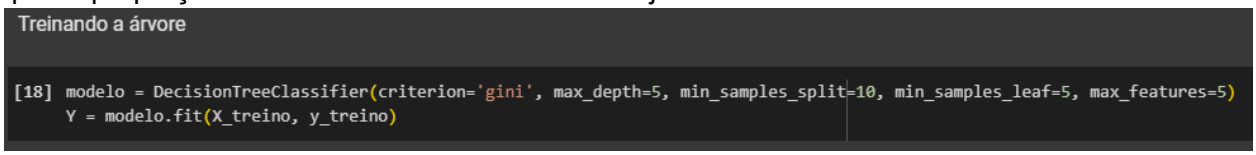
Se Gini = 0, o nó é puro (ou seja, todos os exemplos pertencem a uma única classe).

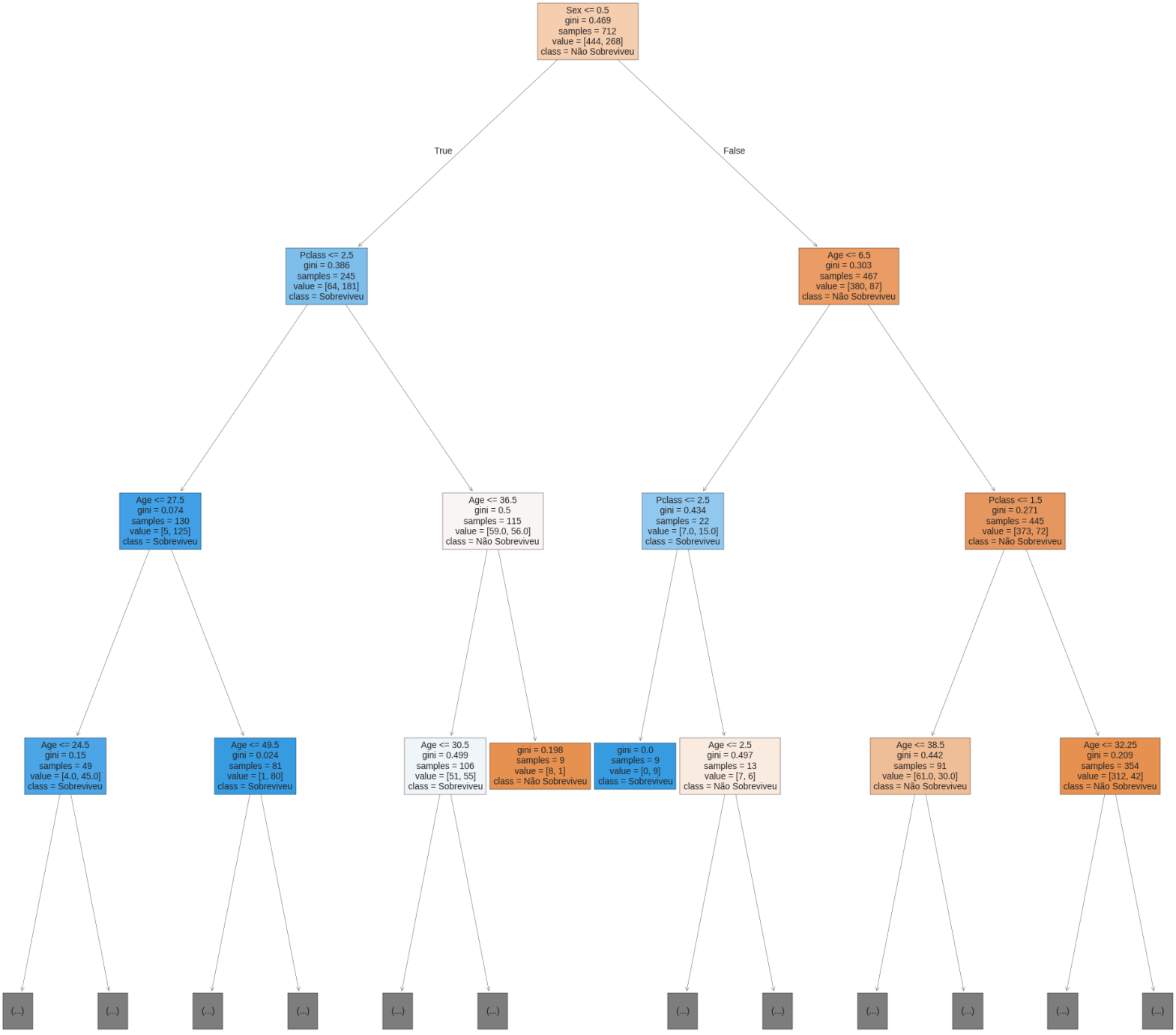
Se Gini = 0.5, temos a máxima impureza (as classes estão distribuídas igualmente).

A fórmula do índice de Gini é:

Gini = $1 - \sum_{i=1}^{c} pi^2$ *c é o número de classes

*pi é a proporção de elementos da classe i no conjunto




Sex <= 0.5
gini = 0.469
samples = 712
value = [444, 268]
class = Não Sobreviveu
True
False
Pclass <= 2.5
gini = 0.386
samples = 245
value = [64, 181]
class = Sobreviveu
Age <= 6.5
gini = 0.303
samples = 467
value = [380, 87]
class = Não Sobreviveu
Age <= 27.5
gini = 0.074
samples = 130
value = [5, 125]
class = Sobreviveu
Age <= 36.5
gini = 0.5
samples = 115
value = [59.0, 56.0]
class = Não Sobreviveu
Pclass <= 2.5
gini = 0.434
samples = 22
value = [7.0, 15.0]
class = Sobreviveu
Pclass <= 1.5
gini = 0.271
samples = 445
value = [373, 72]
class = Não Sobreviveu
Age <= 24.5
gini = 0.15
samples = 49
value = [4.0, 45.0]
class = Sobreviveu
Age <= 49.5
gini = 0.024
samples = 81
value = [1, 80]
class = Sobreviveu
Age <= 30.5
gini = 0.499
samples = 106
value = [51, 55]
class = Sobreviveu
gini = 0.198
samples = 9
value = [8, 1]
class = Não Sobreviveu
gini = 0.0
samples = 9
value = [0, 9]
class = Sobreviveu
Age <= 2.5
gini = 0.497
samples = 13
value = [7, 6]
class = Não Sobreviveu
Age <= 38.5
gini = 0.442
samples = 91
value = [61.0, 30.0]
class = Não Sobreviveu
Age <= 32.25
gini = 0.209
samples = 354
value = [312, 42]
class = Não Sobreviveu
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)

Questão 2)

1. Profundidade Máxima (max_depth)
Este hiperparâmetro define a profundidade máxima que a árvore pode atingir. Se o valor for muito baixo, o modelo pode ser incapaz de capturar padrões importantes nos dados, levando a um underfitting. Por outro lado, se a árvore crescer excessivamente, ela pode memorizar os dados de treinamento, resultando em overfitting. Um valor ideal deve equilibrar a complexidade e a capacidade de generalização do modelo.

2. Número Máximo de Features Consideradas (max_features)
Esse parâmetro determina quantos atributos são avaliados ao decidir a melhor divisão em um nó. Ele pode ser definido como um número absoluto, um percentual do total de atributos ou como 'auto', onde todas as features são usadas. Reduzir o número de features pode aumentar a variabilidade entre diferentes execuções do modelo e ajudar a evitar overfitting, especialmente em bases de dados grandes.

3. Número Mínimo de Amostras em uma Folha (min_samples_leaf)
Define o número mínimo de exemplos que um nó folha pode conter. Se esse valor for muito baixo, a árvore pode se tornar muito complexa, com folhas contendo apenas um ou poucos exemplos, o que pode levar a overfitting. Ao aumentar esse valor, a árvore se torna mais restrita, exigindo que cada folha tenha um número mínimo de amostras, tornando o modelo mais robusto.

4. Número Mínimo de Amostras para Divisão (min_samples_split)
Este hiperparâmetro define o número mínimo de amostras necessárias para dividir um nó. Se esse valor for muito pequeno, a árvore pode crescer excessivamente, criando muitas ramificações e aumentando o risco de overfitting. Se for muito grande, a árvore pode não capturar relações importantes nos dados, causando underfitting.

5. Critério de Impureza (criterion)
A árvore pode utilizar dois critérios principais para medir a impureza dos nós e decidir como dividi-los:

Índice de Gini: mede a impureza dos nós com base na probabilidade de uma amostra ser classificada incorretamente. Esse critério tende a ser mais rápido de calcular.
Entropia: Baseia-se na teoria da informação para medir a incerteza dos nós. Pode fornecer divisões ligeiramente diferentes do Gini e ser mais interpretável em alguns casos.

6. Estratégia de Divisão (splitter)
Esse hiperparâmetro define como a árvore escolhe os pontos de divisão em cada nó:
"best": Escolhe a melhor divisão possível para aquele nó, maximizando o critério escolhido (Gini ou Entropia).
"random": Escolhe aleatoriamente entre as melhores divisões possíveis, o que pode ajudar a aumentar a diversidade quando se utilizam múltiplas árvores (como em florestas aleatórias).

Questão 3)

Os otimizadores de hiperparâmetros GridSearchCV, RandomizedSearchCV e BayesSearchCV são técnicas para aprimorar o desempenho de modelos de machine learning, cada uma com estratégias distintas. O GridSearchCV realiza uma busca exaustiva em todas as combinações pré-definidas de hiperparâmetros. Por exemplo, ao treinar um modelo de Random Forest no dataset do Titanic, ele testa todas as combinações de valores como n_estimators, max_depth e min_samples_split definidos pelo usuário, avaliando cada uma via validação cruzada. Apesar de garantir a melhor combinação dentro do espaço especificado, seu custo computacional é alto, especialmente em grades grandes.

Já o RandomizedSearchCV substitui a abordagem exaustiva por amostragem aleatória. Em vez de testar todas as combinações, ele seleciona um número pré-determinado (n_iter) de valores aleatórios a partir de distribuições (como inteiros uniformes ou listas). No exemplo do Titanic, essa técnica permite explorar intervalos mais amplos (ex: n_estimators entre 50 e 300) com menor tempo de execução, embora não garanta a combinação ótima.

Por fim, o BayesSearchCV utiliza otimização Bayesiana, construindo um modelo probabilístico (como um Processo Gaussiano) para prever quais hiperparâmetros têm maior potencial de melhorar o desempenho. Ele direciona as buscas para regiões promissoras do espaço de hiperparâmetros, reduzindo o número de iterações necessárias. No Titanic, isso se traduz em encontrar combinações eficientes (ex: max_depth=None com min_samples_split=3) com menos recursos computacionais, ideal para espaços complexos ou contínuos.

Questão 4)

a) Apenas I, II

Questão 5)

a) Apenas I, II

Questão 6)

O C4.5 é uma evolução direta do ID3, introduzindo melhorias significativas que tornam o modelo mais eficiente e flexível. As principais diferenças entre eles são:

O C4.5 lida com atributos contínuos, discretos e dados de treinamento com atributos incompletos, poda de árvores após criação e usa a Razão de Ganho.

Questão 7)

→Ganho de Informação (ID3):

Definição: Mede quanto um atributo reduz a incerteza do conjunto de dados.

Problema: Tende a favorecer atributos com muitos valores únicos, como identificadores, que não são úteis para

→Razão de Ganho (C4.5):

Definição: Ajusta o Ganho de Informação para levar em conta a dispersão dos valores do atributo.

Solução: Penaliza atributos com muitos valores distintos, evitando o viés do Ganho de Informação.