Lista 2 - Inteligência Artificial

Nome: **Victor Ferraz de Moraes**
Matrícula: **802371**

Questão 1)

Instância 1 - **Iris-versicolor**
Instância 2 - **Iris-setosa**
Instância 3 - **Iris-versicolor**
Instância 4 - **Iris-virginica**

Resposta: **Letra C**

Questão 2)

I. Esta árvore possui 5 regras de classificação

Está correto, pois há 5 caminhos distintos até as folhas.

II. Das regras geradas, há apenas uma com cobertura por classe de 100%

Está correto, pois a única regra que classifica totalmente uma classe é a regra “petallenght <= 2.35”. As outras regras não conseguem realizar uma classificação completa.

III. A menor cobertura por classe é de 6.8% e corresponde à classe Iris_Virgínica

Não está correto, pois a folha terminal que possui a menor cobertura por classe é a de Iris-versicolor, possuindo 1/37 (2,7%) de cobertura por classe. A cobertura por classe da classe Iris-Virginica de fato corresponde a 6,8% sendo, portanto, a segunda menor.

Resposta: **Letra C**

Questão 3)

	Precisão	**Recall**	**F1Score**	**TVP**	**TFN**	**TFP**	**TVN**
A	58,8%	58,8%	58,8%	58,8%	41,17%	6,66%	93,33%
B	65,21%	83,3%	73,14%	83,3%	16,6%	7,69%	93,20%
C	76,92%	66,6%	71,38%	66,6%	33,3%	6,52%	93,47%
D	89,28%	87,71%	88,48%	87,7%	12,28%	9,23%	90,76%

Precisão = VP / VP + FP
Recall = TVP = VP / VP + FN
F1 Score = 2 * (Precisão * Recall / Precisão + Recall)
TFN = FN / FN + VP
TFP = FP / FP + VN
TVN = VN / VN + FP

A)

Precisão = 10 / 10 + 7 = 0,588
Recall = TVP = 10 / 10 + 7 = 0,588
F1 Score = 2 * (0,588 * 0,588 / 0,588 + 0,588) = 0,588
TFN = 7 / 7 + 10 = 0,4117
TFP = 7 / 7 + (B + C + D) = 0,0666
TVN = (B+C+D) / (B+C+D) + 7 = 0,9333

B)

Precisão = 15 / 15 + 8 = 0,6521
Recall = TVP = 15 / 15 + 3 = 0,833
F1 Score = 2* (0,6521 * 0,833 / 0,6521 + 0,833) = 0,7314
TFN = 3 / 3 + 15 = 0,166
TFP = 7 / 7 + (A+C+D) = 0,0769
TVN = (A+C+D) / (A+C+D) + 7 = 0,9320

C)

Precisão = 20 / 20 + 6 = 0,7692
Recall = TVP = 20 / 20 + 10 = 0,666
F1 Score = 2* (0,7692 * 0,666) / (0,7692 + 0,666) = 0,7138
TFN = 10 / 10 + 20 = 0,3333
TFP = 6 / 6 + (A+B+D) = 0,0652
TVN = (A+B+D) / (A+B+D) + 6 = 0,9347

D)

Precisão = 50 / 50 + 6 = 0,8928
Recall = TVP = 50 / 50 + 7 = 0,8771
F1 Score = 2 * (0,8928 * 0,8771 / 0,8928 + 0,8771) = 0,8848
TFN = 7 / 7 + 50 = 0,1228
TFP = 6 / 6 + (A+B+C) = 0,0923
TVN = (A+B+C) / (A+B+C) + 6 = 0,9076

Questão 4)

Primeiramente vou explicar do que se trata a métrica GINI. A métrica GINI foi inventada pelo matemático italiano Conrado Gini e é um instrumento de medida que foi utilizado para medir o grau de concentração de renda em determinado grupo. Ele demonstra a diferença entre os rendimentos mais pobres e dos mais ricos. Seu valor varia de 0 a 1, em que 0 representa uma situação de igualdade e 1 representa que uma pessoa detém toda a riqueza.

Dito isto, esta métrica pode ser utilizada em outras áreas, como o Machine Learning. Ela é utilizada no CART para gerar os nós da árvore de decisão de maneira mais pura possível, em que o algoritmo escolhe qual divisão irá realizar ao calcular o índice GINI como 1 - (Ai^2 + Aj^2 + An^2), em que A representa o número de amostras de cada classe no nó, ou seja 1 menos a soma das proporções quadradas de cada classe naquele nó. Após realizar o cálculo para cada classe da base de dados, a divisão será feita naquele que minimiza a média ponderada dos índices de GINI dos nós filhos, assim, reduzindo a incerteza nos dados.

Questão 5)

Parte 1 - Processamento - Balanceamento

O desbalanceamento de uma base de dados ocorre quando diferentes classes em um conjunto de dados têm quantidades desiguais de instâncias. Isto é comum em aplicações do mundo real e por isso, pode levar a um desempenho insatisfatório dos modelos de classificação, pois eles tendem a favorecer a classe majoritária.

Portanto, é necessário desenvolver técnicas para lidar com este desbalanceamento. Existem 3 técnicas principais para resolver o problema, que incluem:

1) **Redefinir o tamanho do conjunto de dados.**
    - Oversampling: Esta técnica envolve aumentar o número de instâncias da classe minoritária. Isso pode ser feito replicando exemplos existentes ou gerando novos exemplos.
    - Undersampling: Ao contrário do oversampling, o undersampling reduz o número de instâncias da classe majoritária. Isso pode ser feito removendo aleatoriamente exemplos da classe majoritária até que as classes estejam mais equilibradas. Embora essa técnica possa ser eficaz, ela pode resultar na perda de informações valiosas.
2) **Utilizar diferentes custos de classificação para as classes.**
    - Essa abordagem atribui custos diferentes para erros de classificação em classes desbalanceadas. Por exemplo, um erro na classificação de um exemplo da classe minoritária pode ser considerado mais grave do que um erro na classe majoritária. Isso força o modelo a ser mais cauteloso ao classificar a classe minoritária, ajudando a melhorar a precisão dessa classe.

3) **Induzir um modelo focado em uma classe específica.**
    - Em vez de tentar classificar todas as classes simultaneamente, essa técnica envolve a construção de modelos que se concentram em prever cada classe.

Parte 2 - Processamento - Dados Ausentes

Dados ausentes referem-se à falta de informações dentro de um conjunto de dados, o que pode comprometer a qualidade e a análise. As causas para dados ausentes podem incluir problemas nos equipamentos de coleta, falhas na transmissão e armazenamento, ou erros humanos durante o preenchimento dos dados. E por este motivo que a presença de dados ausentes pode levar a dificuldades significativas, afetando a precisão dos resultados. É essencial tratar esses dados para garantir a eficácia dos algoritmos de AM.

Dentre as principais técnicas para lidar com dados ausentes, podemos contar com as seguintes:

1) **Utilizar algum método ou heurística para automaticamente definir os valores**
    - Preencher os dados ausentes com valores estimados, como a média, mediana ou moda das instâncias disponíveis. Esta estratégia é uma das mais recomendadas.
2) **Remoção**
    - Eliminar registros ou atributos que contêm dados ausentes, embora isso possa causar a perda de informações valiosas. Esta estratégia não é recomendada quando poucos atributos da instância possuem valores ausentes, quando o número de instâncias que sobrarem for pequeno ou quando o número de atributos com valores ausentes varia muito entre as instâncias com esse problema.
3) **Algoritmos**
    - Utilizar algoritmos que podem lidar diretamente com dados ausentes, evitando a necessidade de imputação ou remoção. Como é o caso, por exemplo, de alguns algoritmos indutores de árvore de decisão.
4) **Inserção manual**
    - Esta estratégia não é recomendada se o número de instâncias da base de dados for muito grande.

Parte 3 - Processamento - Dados inconsistentes e redundantes

Dados inconsistentes: Dados inconsistentes são aqueles que apresentam conflitos ou contradições dentro do conjunto de dados. Isso pode ocorrer devido a erros de entrada, falhas na coleta de dados ou falta de padronização.

Dados Redundantes: Dados redundantes referem-se à duplicação de informações dentro de um conjunto de dados, podendo aumentar o volume de instâncias desnecessariamente.

Tanto a inconsistência quanto a redundância podem comprometer a qualidade dos dados, dificultando a análise e a tomada de decisões. Dados de baixa qualidade podem levar a conclusões erradas e afetar a confiabilidade dos resultados..

Parte 4 - Processamento - Conversão simbólica-numérica

A conversão de dados simbólicos para numéricos é necessária para que os modelos possam processar e analisar os dados de forma eficaz. Dados simbólicos, como categorias ou rótulos, não podem ser diretamente utilizados em cálculos matemáticos, que são fundamentais para a maioria dos algoritmos de AM. Por isso, foi-se desenvolvido alguns métodos para lidar com estes tipos de dados:

1) **Codificação One-Hot**
    - Este método transforma cada categoria em uma nova coluna binária, onde 1 indica a presença da categoria e 0 indica sua ausência.
2) **Label Encoding**
    - Neste método, cada categoria é convertida em um número inteiro único. Embora seja simples, pode acabar inferindo uma ordem que não existe entre as categorias, o que pode ser problemático para alguns algoritmos de AM.
3) **Codificação Ordinal**
    - Utilizada quando as categorias têm uma ordem natural (por exemplo, "baixo", "médio", "alto"). Cada categoria é atribuída a um número que reflete essa ordem.

A escolha do método de conversão deve ser feita com cuidado, pois a forma como os dados são representados pode impactar significativamente o desempenho do modelo. Métodos inadequados podem levar a interpretações errôneas dos dados e, consequentemente, a resultados insatisfatórios.

Parte 5 - Processamento - Conversão numérico-simbólica

A conversão de dados numéricos para simbólicos é necessária para traduzir os resultados de forma compreensível e interpretável. Deve ser feita com cuidado para garantir que a interpretação dos dados seja precisa e que não haja perda de informação.
Dentre os métodos para esta conversão, temos:

1) **Mapeamento Direto**
    - Cada valor numérico é associado a uma categoria específica. Por exemplo, se um modelo retorna valores como 0, 1 e 2, esses podem ser mapeados para categorias como "Baixo", "Médio" e "Alto".
2) **Decisão Baseada em Limites**
    - Para dados contínuos, pode-se definir limites que determinam a qual categoria um valor numérico pertence. Por exemplo, valores abaixo de 50 podem ser categorizados como "Baixo", entre 50 e 75 como "Médio", e acima de 75 como "Alto".

Parte 6 - Processamento - transformação de atributos numéricos

A transformação visa padronizar ou normalizar os valores numéricos para que todos os atributos contribuam de maneira equilibrada para o modelo.
Dentre os principais métodos para realizar estas transformações, temos:

1) **Normalização**
    - Também conhecida como normalização min-max, é o processo de reescalar os valores de um atributo para que fiquem em uma faixa específica, geralmente entre 0 e 1.
    - Sua fórmula pode ser dada por: x’ = min + (x - min(x) / max(x) - min(x)) * (max(x)-min(x))
2) **Padronização**
    - Transforma os dados para que tenham média zero e desvio padrão um. Isso é feito subtraindo a média e dividindo pelo desvio padrão.
    - Sua fórmula pode ser dada por x’ = x - u / rho, onde x é o valor atual, x’ é o novo valor padronizado, u é a média e rho é o desvio padrão.

Parte 7 - Processamento - Redução de dimensionalidade

A redução de dimensionalidade refere-se ao processo de reduzir o número de variáveis (ou dimensões) em um conjunto de dados, enquanto se preserva a informação relevante.
Existem três abordagens principais para avaliar a qualidade ou desempenho de um subconjunto de atributos:

1) **Abordagem Embutida**
    - Nesta abordagem, a seleção de atributos é incorporada diretamente no processo de treinamento do modelo.
    - Uma vantagem é que a seleção é otimizada em relação ao modelo, o que pode levar a melhores resultados.
    - Por outro lado, pode ser dependente do modelo específico utilizado.
2) **Abordagem Baseada em Filtro**
    - A abordagem baseada em filtro avalia a relevância dos atributos de forma independente do modelo. Isso é feito utilizando métricas estatísticas
    - É rápida e não depende de um modelo específico, permitindo que seja aplicada a diferentes algoritmos.
3) **Wrapper**
    - Utiliza um modelo preditivo para avaliar a qualidade de um subconjunto de atributos. Ela envolve a seleção de um conjunto de atributos, treinamento do modelo e avaliação do desempenho, repetindo o processo até encontrar o subconjunto ideal.
    - É computacionalmente intensiva e pode ser propensa ao overfitting, especialmente em conjuntos de dados pequenos.

Reduzir a dimensionalidade pode melhorar a performance dos algoritmos de AM, tornando-os mais rápidos e eficientes. Além disso, com menos dimensões, os algoritmos podem generalizar melhor, evitando problemas associados a maldição da dimensionalidade.