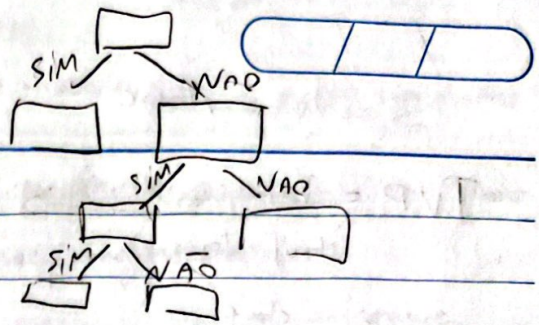

Lista 2 - IA

Q2 - Instância 1 = 1 petallenght >, 2.35 (NÃO) (tamanho)
2 - petalwidth < 1,75 (SIM)
3 - petallenght < 4.95 (SIM)
↳ Iris - Versicolor

Instância 2 = 1 - SIM → Iris - Setosa
Instância 3 = 1 - NÃO 2 - NÃO 3 - SIM → Iris - Versicolor
Instância 4 = 1 - NÃO 2 - NÃO 3 - NÃO → Iris - Virginica

letra c) //

Q2 - I - Sim há 5 folhas na árvore (5 regras de classificação)
II - Iris setosa → 39/120 = 32,5%
Iris versicolor → 35+2/120 = 30,8% (35/120 = 29,1% | 2/120 = 1,6%)
Iris virginica → 39+5/120 = 36,6% (39/120 = 32,5% | 5/120 = 4,16%)
(nenhuma com 100%)
III - a menor cobertura por clone é de 1,6% da versicolor
↳ letra a) // (1 apenas)

Q3 - Precisão (P) = $\frac{Vp}{Vp + Fp}$ A = 10/17 = 0,58 C = 20/26 = 0,76
B = 15/23 = 0,65 D = 50/56 = 0,89

Recall (R) = $\frac{Vp}{Vp + Fn}$ A = 10/17 = 0,58 C = 20/30 = 0,66
B = 15/18 = 0,83 D = 50/57 = 0,87

F1 score = $\frac{2 \cdot (P \cdot R)}{P + R}$ A = $\frac{0,34}{1,17}$ = 0,29 C = $\frac{1,0032}{1,42}$ = 0,70

B = $\frac{1,08}{1,48}$ = 0,72 D = $\frac{1,56}{1,76}$ = 0,88

A = 17
B = 23
C = 26
D = 57

acerto na classe

TVP = Verdadeiros X / classificados como X → A = 10/17 = 0,58 C = 20/26 = 0,76
B = 15/23 = 0,65 D = 50/57 = 0,89

erro na classe

TFN = Falsos X / classificados como X → A = 7/17 = 0,41 C = 10/26 = 0,38
B = 3/23 = 0,13 D = 7/57 = 0,76

TFP = (quantidade de classificados como A que não são da classe A) / (B + C + D) → A = 8/106 C = 6/97
B = 8/100 D = 6/66

(B + C + D: soma de todos os demais classificados)

TVN = (quantidade que não é da classe e foi classificado como não A) / (B + C + D) → A = 96/106 (106 − 10) C = 77/97 (97 − 20)
B = 85/100 (100 − 15) D = 16/66 (66 − 50)

04 - CART → Classification and Regression Trees

↳ A métrica gini é usada no CART para medir a impureza de um nó, relacionado à entropia dele.

↳ Quanto menor o valor de Gini, mais concentrada está uma classe (+ puro é o nó)

↳ Cálculo:

$$Gini = 1 - \sum_i (p_i^2)$$

p_i → probabilidade de critério ser V (proporção de elementos de uma classe)

↳ Assim, ao definir a métrica gini de todos os nós, o algoritmo verifica quais são os de menor valor gini, os quais serão os classificadores da árvore (splitting rule).

05 - PROCESSAMENTO

(etapas de pré-processamento)

PARTE 1 - balanceamento da base de dados

↳ em alguns subconjuntos, seus dados aparecem com frequência maior que os dados dos demais classes.

oversampling causa:
- overfitting → quando o modelo é superajustado aos dados de treinamento

undersampling causa:
- underfitting → quando o modelo não se ajusta aos dados de treinamento.

↳ quando alimentados c/ dados desbalanceados, os algoritmos tendem a favorecer a classificação de novos dados na classe majoritária.

↳ principais técnicas p/ resolver:

↳ redefinir o tamanho do conjunto de dados (adiciona instâncias à classe minoritária - oversampling, quanto remover - undersampling)

↳ utilizar diferentes custos de classificação para os diferentes classes

↳ induzir um novo modelo para uma classe (a classe minoritária ou majoritária são aprendidos separadamente.

PARTE 2 - tratamento de dados ausentes

↳ dificuldade relacionada à qualidade dos dados. Pode ser causada por problemas nos equipamentos de coleta, transmissão e armazenamento dos dados ou no preenchimento dos dados (por humanos) → falta de atenção, distração, inexistência, etc

Δ como resolver:

↳ eliminar as instâncias c/ dados ausentes

↳ preencher manualmente os valores faltantes → como moda, média, etc

↳ método / heurística para definir valores automáticos nos campos faltantes

↳ algoritmos de AM que lidam internamente c/ valores ausentes.

PARTE 3 - dados inconsistentes e redundantes

↳ são os dados que possuem valores conflitantes em seus atributos. Diferentes conjuntos de dados podem usar escalas diferentes p/ uma mesma medida, etc.

↳ podem ser tanto instâncias quanto atributos

ex: dados idênticos com resultados opostos → ex: idade e data de nascimento

Δ como resolver:

↳ filtros que ajudam na eliminação de redundantes ou inconsistentes.

PARTE 4- conversão simbólico-numérico

↳ técnicas como Redes Neurais artificiais, SVM e outros algoritmos de agrupamento lidam apenas com dados numéricos, e muitas vezes precisamos transformar os nomes nominais/categóricos em numéricos.

Δ como resolver:

↳ caso 1: o atributo assume apenas dois valores de presença/ausência → usamos um dígito binário.

↳ caso 2: tipo simbólico c/ + de dois valores

nominal ←

- a diferença entre todos os valores numéricos deve ser a mesma (cada valor é uma sequência de c bits → codificação 1-de-c)
- para a distância → distância de Hamming
- são 6 bits! Não um nº inteiro! outra alternativa: pseudoatributos

→ ordinal

- a ordem dos valores deve estar clara
- ordena os valores categóricos ordinais e codifica cada valor de acordo c/ sua posição na ordem.
- código cinza ou termômetro se for necessário

PARTE 5- conversão numérico-simbólica

↳ algumas técnicas de AM usam valores qualitativos.

Δ como resolver:

↳ discretizar o atributo

↳ o conjunto de possíveis valores é dividido em intervalos, e cada intervalo de valores quantitativos é convertido em um valor qualitativo.

↳ podem ser supervisionados ou não-supervisionados

info sobre classe dos exemplos

PARTE 6 - Transformação de atributos numéricos

↳ um valor numérico pode precisar ser convertido em outro valor numérico, quando os limites inferior e superior de valores dos atributos são muito diferentes ou estão em escalas diferentes.

Δ como resolver:

↳ Transformação de normalização de dados:

↳ por amplitude (por reescala ou padronização)

define uma nova escala

$$V_{novo} = min + \frac{V_{atual} - menor}{maior - menor}(max - min)$$

↳ cada valor é somado + ou - uma medida e depois × ou ÷ outra.

$$V_{novo} = \frac{V_{atual} - \mu}{\sigma}$$

PARTE 7 - redução de dimensionalidade

↳ muitos problemas possuem um nº elevado de atributos, e poucas técnicas de AM podem lidar c/ um número tão grande, chamado de maldição de dimensionalidade

Δ como resolver:

↳ combinar ou eliminar parte dos atributos irrelevantes com:

↳ agregação (novos atributos formados pela combinação de grupos de atributos)

↳ seleção de atributos (descartam os demais)

↳ 3 abordagens são usadas p/ avaliar a qualidade/desempenho de um subconjunto:

↳ embutida (seleção embutida no algoritmo de aprendizado. Ex: árvore de decisão)

↳ baseada em filtro (filtra os subconjuntos. Ex: correlação)

↳ baseada em wrapper (usa o próprio algoritmo como uma caixa-preta p/ a seleção)

↳ eficientes e simples