Lista IA – 2

ALUNO: Caio Gomes Alcântara Glória

MATRICULA: 763989

PROFESSOR(A): Cristiane Neri

Questão 1)

Registros de teste	Tamanho da Pétala	Largura da Pétala	Tamanho da Sépala	Largura da Sépala
Instância 1	3.46	0.87	2.45	1.78
Instância 2	1.67	1.89	0.78	1.32
Instância 3	2.56	2.34	2.45	1.78
Instância 4	6.67	2.34	2.45	1.78

1 – íris-versicolor

2 – íris-setosa

3 – íris-versicolor

4 – íris-virginica

Conclusão: letra -> C <-

Questão 2)

a) Verdadeiro, isso ocorre pois o número de regras é o mesmo valor do número de nós folhas.

b) Verdadeiro, íris setosa = 39/39 = 1 = 100%, íris versicolor1 = 34/37 = 0,9189 = 91,9%, íris versicolor2 = 1/37 = 0,0270 = 2,7%, íris virginica1 = 3/44 = 0,0681 = 6,81%, íris virginica2 = 39/44 = 0,8863 =88,63%.

c) Falso, cobertura menor é de 2,7% na íris versicolor 2.

Conclusão: letra -> C <-

Questão 3)

A -> TP = 10, TN = 98, FP =7 , FN=7 .

B-> TP = 15, TN = 96, FP =8 , FN=3 .

C-> TP =20 , TN = 86, FP =6 , FN=10 .

D-> TP =50 , TN = 59, FP =6, FN= 7.

XXXXXXX	precisão	recall	F1Score	TVP	TFN	TFP	TVN
A	0,588	0,588	0,588	0,588	0,411	0,06	0,933
B	0,652	0,833	0,731	0,833	0,166	0,076	0,923
C	0,769	0,6	0,713	0,66	0,333	0,065	0,934
D	0,892	0,877	0,884	0,877	0,122	0,092	0,907

Questão 4)

O índice de Gini é uma métrica central no algoritmo CART para avaliar a qualidade de uma divisão, buscando criar nós com maior pureza (ou homogeneidade) ao construir uma árvore de decisão. Quanto menor o índice de Gini, mais homogêneo é o nó, e o CART seleciona a divisão que minimiza o Gini, criando árvores de decisão mais eficazes na classificação.

O índice de Gini mede o grau de probabilidade de um item ser classificado incorretamente se ele fosse rotulado aleatoriamente de acordo com a distribuição das classes no nó. A métrica varia entre 0 e 1, onde:

- 0: O nó é puro, ou seja, contém apenas exemplos de uma única classe.
- 1: O nó é o mais impuro possível, com uma distribuição uniforme entre as classes.

EX de código:

CART.py

Questão4 > CART.py

```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import matplotlib.pyplot as plt

# Carregando o dataset Iris como exemplo
data = load_iris()
X = data.data
y = data.target

# Dividindo os dados em conjunto de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Criando o modelo CART (Árvore de Decisão)
clf = DecisionTreeClassifier(criterion='gini', random_state=42)

# Treinando o modelo
clf.fit(X_train, y_train)

# Avaliando o modelo
accuracy = clf.score(X_test, y_test)
print(f"Acurácia do modelo: {accuracy * 100:.2f}%")

# Visualizando a árvore de decisão
plt.figure(figsize=(12, 8))
tree.plot_tree(clf, filled=True, feature_names=data.feature_names, class_names=data.target_names)
plt.show()

```

Questão 5)
Arquivo 1:

Dados desbalanceados entram na área de classificação de dados, este problema se torna comum em que dados de um subconjunto aparecem com mais frequência que os de conjunto principal. Um exemplo disso é, se 80% de pessoas vao ao hospital HOJE para fazer teste de COVID dão negativo, 20% significa que estão 100% com COVID, o que é um Desbalanceamento de dados. Torando os negativos majoritários e os positivos minoritários.

Em resumo ter dados desbalanceados prejudica no aprendizado de máquina pois o mesmo acaba favorecendo classificação de novos dados na classe majoritária. Técnicas para resolver este problema podem ser: redefinir tamanho do conjunto, utilizar diferentes custos de classificação e induzir o modelo a uma classe.

Arquivo 2:

Bases de Dados podem ter dados ausentes e isso inflige diretamente na qualidade dos dados. Dados ausentes podem ser causados por: erro de coleta, erro de transmissão de dados, problemas no preenchimento ou na entrada dos dados por seres humanos.

Como resolver estes problemas? Primeiramente eliminar as instâncias que possuem dados ausentes (Não recomentado se a quantidade de valores ausentes não for significante), definir e preencher os dados faltantes também é uma opção viável (Não recomendado se a quantidade de instancias faltantes for muito grande), utilizar algoritmos internos que tratam destes valores ausente (como tratamento de exceções), Weka/Filters/unsupervised/atribute.

Arquivo 3:

Dados inconsistentes podem ser definidos como dados com valores conflitantes, essa situação geralmente ocorre no processo de integração de dados. Duas instâncias iguais com classificações diferentes por exemplo, ou então quando um atributo é muito semelhante a outro e a classificação também não está igual, quando instancias da mesma base de dados são idênticas. Um atributo é redundante quando seu valor para todas as instâncias pode ser deduzido a partir do valor de um ou mais atributos.

Arquivo 4:

Transformar dados simbólicos para numéricos na intenção de facilitar uso de redes neurais artificiais, SVM e algoritmos de agrupamento é algo frequente e existem maneiras de realizar este processo.

CASO 1: Quando o atributo é simbólico mas pode ser classificado com digito binário, exemplo disso pode ser: Gênero -> masculino (0) e feminino (1), ou então Tumor-> maligno(0) e benigno(1).
CASO 2: Quando o atributo escapa da possibilidade de utilizar apenas dois valores ele pode ser convertido para numeral ordinal (em ORDEM NUMERICA). Um método dentro deste caso é utilizar a distância de Hamming, onde cada posição da sequência binária corresponde a um possível valor do atributo nominal. Exemplo: azul = 0001, branco = 0010, preto = 0011, assim por diante. Reforçando que este método é recomentado para atributos com poucas opções de respostas, o mesmo em grande escala não tem a mesma eficácia.

CASO 3: O mais simples é quando os atributos já possuem previamente uma ordem, então uma solução simples é usar um número inteiro ou real para sequencia-los.

Arquivo 5:

Ao contrário do último arquivo, o contrário também existe, isso é preciso quando se deseja trabalhar com valores qualitativos. A recomendação é discretizar o atributo, quanto um atributo quantitativo é discretizado, o conjunto de possíveis valores é dividido em intervalos, e cada intervalo quantitativo é convertido em um valor qualitativo. Estes podem ser SUPERVISIONADOS E NÃO SUPERVISIONADOS, as que são supervisionadas geralmente trazem melhores resultados.

Metodos- >

Larguras iguais: Divide o intervalo original de valores em subintervalos em mesma largura.

Frequências iguais: Atribui o mesmo número de objetos a cada subintervalo.

Inspeção visual.

Um exemplo pode ser em temperaturas: <10 = Frio, 10> e <25 = moderado , >25 = quente.

Arquivo 6:

Algumas vezes, o valor numérico de um atributo precisa ser transformado em outro valor numérico o Isso ocorre quando os limites inferior e superior de valores dos atributos são muito diferentes, o que leva a uma grande variação de valores, ou ainda quando vários atributos estão em escalas diferentes.

Assim uma forma de normalizar estes atributos é pela reescala, esta que possui uma formula para assim os limites superior e inferior sejam 1 e 0 respectivamente.

Outro método pode ser a padronização (formula de Zscore), Se as medidas de localização e de escala forem a média (μ) e o desvio padrão (σ), respectivamente, os valores de um atributo são convertidos para um novo conjunto de valores com média 0 e desvio padrão 1. De uma maneira geral, se a distribuição não é Gaussiana ou o desvio padrão é muito pequeno, normalizar os dados é uma escolha a ser tomada.

Arquivo 7:

A maldição de dimensionalidade, se um conjunto de dados em cada instancia possui 1 atributo e o mesmo pode ter 10 valores distintos, esse conjunto pode ter entao 10 instancias diferentes. Agora, se o número de atributos passar para 5, o número de instâncias passa a ser 10^5, MUITO MAIOR. Uma forma de minimizar o impacto do problema da dimensionalidade é combinar ou eliminar parte dos atributos irrelevantes.

Sendo assim, as abordagens de redução de dimensionalidade podem ser em: AGREGAÇÃO (Substituem os atributos originais por novos atributos formados pela combinação de grupos de atributos) E SELEÇÃO DE ATRIBUTOS (Mantem uma parte dos atributos originais e descartam os demais atributos).