

1. [2 questões] Conhecimento contextual

1.1. A área de “Inteligência Artificial” tem crescido grandemente na última década, num relacionamento de autorreforço com outras duas áreas de Informática.

→ Quais essas duas outras áreas? Justifique e comente o “autorreforço”.

As áreas são IoT (Internet of Things) e Computação na Nuvem (Cloud Computing/CC).

IoT descentralizou e aumentou a disponibilidade de dados; CC tornou possível o arquivo e a computação sobre esses dados a qualquer utilizador (pagador), contribuindo para economias de escala nos recursos e serviços relacionados. Tem-se um autorreforço destas três áreas (IoT, CC e AI) porque colocam pressão crescente umas sobre as outras, e à medida que soluções vão ficando disponíveis, criam-se condições para um patamar crescente de novas exigências.

1.2. Imagine um dataset de cogumelos, todos classificados por peritos quanto à sua “adequação culinária”. A classificação está feita numa escala discreta de 10 níveis, por exemplo de 0/totalmente NÃO adequado a uso culinário, até 9/perfeito para uso culinário.

Cada cogumelo é uma amostra estruturada em 22 medidas, não importa exatamente quais, mas pode imaginar atributos como “área do chapéu”, “intensidade do odor”, etc.

Destas medidas, algumas não são de observação direta, mas resultam de computações feitas utilizando toda a coleção de dados, como médias e/ou normalizações. Por exemplo, uma das medidas pode indicar o fator pelo qual a amostra excede a média em “intensidade do odor”.

→ Quando um modelo for treinado com parte deste dataset, vê [sim ou não?] algum problema/vantagem [qual?] decorrente da estrutura das amostras? Justifique.

Sim, identifica-se um problema. O problema é um contributo para overfitting, uma vez que todas as amostras do dataset, incluindo as elegidas para treino, terão informação das outras (incluindo sobre as reservadas para teste), via computações indiretas em alguns atributos.

Nota adicional: também se chama a este fenómeno de dados que se desejavam reservados para teste estarem [residualmente] nos dados de treino, “data leakage”.

2. [2 questões] Algoritmos de distância em Python

Nesta questão vai trabalhar com amostras hipotéticas de um dataset, cujos valores dependem do seu número de estudante.

Do seu número obtêm-se 4 amostras (o, r, d2, s), na forma de lista:
o = lista dos dígitos do seu número de aluno;
r = lista dos dígitos do seu número de aluno, pela ordem inversa
d2 = lista, paralela às listas o e r, em que cada componente é a soma dos componentes correspondentes em o e r, dividida (divisão inteira) por 2;
s = lista sintética, paralela às listas o e d2, em que cada componente é a soma dos componentes correspondentes em o e d2, nas posições pares (começando a indexação em zero), caso contrário é a sua diferença (subtração).

Exemplo para o aluno com número "125":

o = [1, 2, 5]

r = [5, 2, 1]

d2 = [3, 2, 3]

s = [4, 0, 8]

Considere que se pretende usar um algoritmo de distância "minkowski_distance_no_root", em tudo idêntico à distância de Minkowski de ordem 2, exceto que não se calcula a operação de raiz de quadrada.

→ 2.1. Para o seu número de aluno, quanto vale `minkowski_distance_no_root(o, s)` ?

Para o aluno "125" vale 22.

o = [1, 2, 5]

r = [5, 2, 1]

d2 = [3, 2, 3]

s = [4, 0, 8]

`d(o, s) = 22`

→ 2.2. Escreva a função Python com assinatura:

```
def student_number_to_tuples(p_student_number: str) -> tuple
```

que possa utilizada para obter as listas anteriores:

- deve receber uma frase correspondente a um número de aluno;
- deve formar as listas o, r, d2, s;
- deve retornar um tuplo composto por essas listas.

TODO

3. [2 questões] *Abordagens Bayesianas*

Está a testar-se um sistema "novo", para substituir outro sistema, considerado "perfeito", mas inaceitavelmente caro.

Há 4 variantes do sistema "novo", com custos muito diferentes: V1, V2, V3, V4.

O sistema "perfeito" produz, por dia, um grande conjunto de medidas. Ao longo de vários dias, registaram-se quantas vezes as variantes V1, V2, V3, V4, funcionaram exatamente conforme o sistema "perfeito".

Quando uma variante coincide perfeitamente com o sistema "perfeito", tem-se uma situação "P"; caso contrário, tem-se uma situação "NOT P".

Eis uma descrição textual dos resultados obtidos:

Aconteceram, ao todo, 60 utilizações do sistema "novo", 12 do tipo V1, 14 V2, 16 V3, 18 V4.

11 V1 atingiram P, 1 não. 10 V2 atingiram P, 4 não.

10 V3 atingiram P, 6 não. 6 V4 atingiram P, 12 não.

Ao todo, 37 unidades atingiram "P", 23 unidades não.

→ 3.1 Complete a tabela que sintetiza a realidade observada, de forma a corresponder à descrição textual:

Variante	Quantidade de instalações da variante com resultado "NOT P"	Quantidade de instalações da variante, com resultado "P"	Relação
V1	1	11	12/60
V2	4	10	14/60
V3	6	10	16/60
V4	12	6	18/60
Totais	23	37	
Relação	23/60	37/60	

→ 3.2 Qual a probabilidade de o sistema "novo" não atingir P, na condição de usar-se a variante V3?

Não tem que fazer as contas, basta expressar os valores que entrariam no cálculo.

$$P(\text{NOT P} | V3) = 6/16 \text{ [poderia ser lido diretamente da tabela]}$$

Demonstração:

$$P(\text{NOT P} | V3) = P(V3 | \text{NOT P}) * P(\text{NOT P}) / P(V3) =$$

$$((6 \text{ V3} \wedge \text{NOT P}) / 23 \text{ NOT P no total}) * 23 \text{ NOT P em 60} / 16 \text{ V3 em 60} =$$

$$6/23 * (23/60 / 16/60) = 6/23 * 23/16 = 6/16$$

4. [3 questões] Lógica

4.1 Responda V (Verdade) ou F (Falso), consoante, respetivamente o modelo M seguinte satisfaça a expressão booleana que aparece à direita do operador de satisfação |=

→ 4.1.1

$M = \{a/\text{False}, b/\text{True}, c/\text{True}\}$

$M \models (a \Leftrightarrow b) \text{ or } (a \Leftrightarrow c)$

Truth table for 4.1.1:

a	b	c	$(a \Leftrightarrow b) \text{ or } (a \Leftrightarrow c)$
True	True	True	True
True	True	False	True
True	False	True	True
True	False	False	False
False	True	True	False
False	True	False	True
False	False	True	True
False	False	False	True

Function " $(a \Leftrightarrow b) \text{ or } (a \Leftrightarrow c)$ ", for model $\{ 'a': \text{False}, 'b': \text{True}, 'c': \text{True} \}$, is False

sat result for 4.1.1

False

→ 4.1.2

$M = \{a/\text{False}, b/\text{False}, c/\text{False}\}$

$M \models (a \Leftrightarrow b) \text{ or } \text{not}(a \Rightarrow b)$

Truth table for 4.1.2:

a	b	c	$(a \Leftrightarrow b) \text{ or } \text{not}(a \Rightarrow b)$
True	True	True	True
True	True	False	True
True	False	True	False
True	False	False	True
False	True	True	False
False	True	False	False
False	False	True	True
False	False	False	True

Function "(a<=>b) or not(a=>c)", for model {'a': False, 'b': False, 'c': False}, is True

sat result for 4.1.2

True

→ 4.2 Sejam a, b, c quaisquer proposições booleanas.

Responda V (Verdade) ou F (Falso), consoante, respetivamente, se verifique, ou não, implicação lógica. Se a resposta for F, indique TODOS os modelos possíveis que invalidam a implicação lógica:

$(a \Leftrightarrow b) \text{ or } (a \Rightarrow c) \models a$

Left "(a<=>b) or (a=>c)" logically implies Right "a" is False

F por causa dos modelos: FTF, FFT, FFF

5. [3 questões] Agentes para procuras

Recorde os algoritmos de procura que estudou.

→ 5.1 0 que significa, para o um algoritmo de procura não informada, a estrutura de dados "fronteira"/"frontier" ter ficado vazia?

Se o último estado selecionado para análise não constituir solução, tendo a fronteira ficado vazia, significa que não foi encontrada solução.

→ 5.2 Verdadeiro (V) ou Falso (F)?

O algoritmo BFS (Breadth-First Search) vai, sempre e garantidamente, encontrar o caminho/solução óptima [assumindo o cenário estudado de labirintos de custo unitário e a inexistência de situações que conduzam a ciclos infinitos]. V

→ 5.3 Verdadeiro (V) ou Falso (F)?

O algoritmo DFS (Depth-First Search), nunca encontra o caminho/solução óptima [assumindo o cenário estudado de labirintos de custo unitário e a inexistência de situações que conduzam a ciclos infinitos]. F

6. [4 questões] Interpretação de soluções relacionadas com NN

6.1 Em problemas de classificação multi-classe, é muito comum encontrar-se código Python como o seguinte:

```
from sklearn import model_selection
X_train, X_test, y_train, y_test = \
model_selection.train_test_split (samples, labels)
```

No contexto deste código, responda V (Verdade) ou F (Falso)

→ 6.1.1 $\text{len}(y_{\text{train}}) > \text{len}(X_{\text{train}})$ **F**

→ 6.1.2 Tipicamente, $\text{len}(y_{\text{test}}) > \text{len}(y_{\text{train}})$ **F**

→ 6.1.3 X_{test} representa uma função que opera testes sobre amostras. **F**

6.2 Um modelo programático/rede neuronal baseado na framework Keras, para a classificação de imagens, foi criado da seguinte forma:

```
from keras.models import Sequential
from keras.layers import Dense

iUnits = 16
model = Sequential ( [
    Dense(iUnits, activation="relu", input_shape=(64*64,)),
    Dense(iUnits, activation="relu"),
    Dense(16, activation="softmax")] )
```

→ Qual o número de parâmetros da primeira camada da rede?

Basta apresentar a expressão matemática, não é preciso calcular o valor.

$(64*64)*16 + (1*16)$