UNIVERSIDADE DE AVEIRO

ML — Final Exam Model

Respostas Completas

Nome: _____	Nº: _____ Turma: _____

Q1. Which of these is a reasonable definition of machine learning?

✗	Machine Learning (ML) learns from labeled data.
✓	**ML is the field of study that gives computers the ability to learn without being explicitly programmed.**
✗	ML is the field of allowing robots to act intelligently.
✗	ML is the science of programming computers.

Justificação: Esta é a definição clássica de Arthur Samuel. ML não se limita a dados rotulados (exclui a opção 1), não é só sobre robôs (opção 3), nem é simplesmente programar computadores (opção 4).

Q2. Predict how much rain will fall tomorrow — classification or regression?

✓	**Regression**
✗	Classification

Justificação: A quantidade de chuva (em mm ou polegadas) é um valor contínuo. Prever um valor numérico contínuo é sempre um problema de regressão.

Q3. Predict whether a company will win a patent lawsuit — classification or regression?

✗	Regression
✓	**Classification**

Justificação: O output é binário — ganhar ou perder. Prever uma categoria/classe discreta é sempre classificação. Não existe um valor contínuo intermédio entre ganhar e perder.

Q4. To which of the following would you apply supervised learning?

✗	Discover categories/types of patients in terms of drug response → Unsupervised (Clustering)
✗	Find clusters of heart disease patients for separate treatments → Unsupervised (Clustering)
✓	**Classify web page as child-friendly or adult → SUPERVISED — Classification**
✓	**Predict next year's crop yields from 50 years of data → SUPERVISED — Regression**

Justificação: As primeiras duas opções não têm labels/rótulos (descobrir grupos) → não supervisionado. As últimas duas têm outputs conhecidos: classe (child/adult) = classificação; valor de produção = regressão.

Q5. Logistic regression outputs h_θ(x) = 0.2. This means (check all that apply):

✓	**Our estimate for P(y=1|x;θ) is 0.2.**
✓	**Our estimate for P(y=0|x;θ) is 0.8.**
✗	Our estimate for P(y=1|x;θ) is 0.8.
✗	Our estimate for P(y=0|x;θ) is 0.2.

Justificação: A regressão logística produz diretamente P(y=1|x). Se h(x)=0.2, então P(y=1)=0.2. Por complementaridade, P(y=0) = 1 - 0.2 = 0.8.

Q6. Is the dataset in Table 1 linearly or nonlinearly separable?

Attr 1	Attr 2	Class
-1	1	1
1	-1	1
-1	-1	-1
1	1	-1

Justificação: Este é o problema XOR — NONLINEARLY SEPARABLE. A classe +1 ocorre quando os atributos têm sinais DIFERENTES ((-1,1) e (1,-1)), e a classe -1 quando têm o MESMO sinal ((-1,-1) e (1,1)). Não é possível traçar uma linha reta que separe as duas classes. É necessário um kernel não-linear (ex: RBF) ou features polinomiais.

Q7. Which statements about unsupervised learning are true? (Check all that apply)

✓	**In unsupervised learning, the training set is of the form {x(1), x(2), ..., x(m)} without labels y(i).**
✓	**Clustering is an example of unsupervised learning.**
✓	**In unsupervised learning, you are given an unlabeled dataset and are asked to find 'structure' in the data.**
✗	Clustering is the only unsupervised learning algorithm.

Justificação: As três primeiras são verdadeiras — definem corretamente a aprendizagem não supervisionada. A quarta é falsa: existem outros métodos não supervisionados como PCA, Autoencoders e deteção de anomalias.

Q8. Logistic regression on nonlinearly separable data — which statements are true?

✓	**J(θ) will be a convex function, so gradient descent should converge to the global minimum.**
✓	**Adding polynomial features could increase how well we can fit the training data.**
✗	The positive and negative examples cannot be separated using straight line. So, gradient descent will fail to converge.
✗	Linear regression will perform as well as logistic regression on this data.

Justificação: J(θ) na regressão logística é SEMPRE convexa → gradient descent converge ao mínimo global (opção 1 ✓). Features polinomiais criam fronteiras não-lineares → melhor fit (opção 2 ✓). Gradient descent não falha — converge, só não consegue separar perfeitamente (opção 3 ✗). Linear regression é pior para classificação (opção 4 ✗).

Q9. Which are correct gradient descent updates for logistic regression? (Check all that apply)

✓	**θ := θ - α(1/m)Σ(1/(1+e^(-θᵀx)) - y)x [= h_θ(x) - y]**
✗	θ := θ - α(1/m)Σ(θᵀx - y)x [usa θᵀx em vez de h_θ(x) → regressão linear, não logística]
✓	**θ := θ - α(1/m)Σ(h_θ(x(i)) - y(i))x(i)**
✗	θj := θj - α(1/m)Σ(θᵀx - y)xj [usa θᵀx em vez de sigmoid → errado]

Justificação: A regra correta é θ := θ - α × (1/m) × Σ(h_θ(x) - y)x, onde h_θ(x) = sigmoid(θᵀx). As opções 1 e 3 são equivalentes e corretas. As opções 2 e 4 usam θᵀx (sem sigmoid) — isso seria regressão linear, não logística.

Q10. Which statements about logistic regression are true? (Check all that apply)

✗	For logistic regression, gradient descent will sometimes converge to a local minimum (fail to find global minimum).
✓	**The cost function J(θ) for logistic regression trained with m≥1 examples is always ≥ 0.**
✗	Linear regression always works well for classification using a threshold.
✓	**The sigmoid function g(z) = 1/(1+e^-z) is never greater than 1.**

Justificação: Opção 1 ✗ J(θ) na regressão logística é convexa → só tem um mínimo global, gradient descent converge sempre. Opção 2 ✓ J usa log loss que é sempre ≥ 0. Opção 3 ✗ regressão linear não é adequada para classificação. Opção 4 ✓ sigmoid está sempre em (0,1), nunca excede 1.

Q11. θ0=-6, θ1=1, θ2=0. Which figure represents the decision boundary?

Justificação: A fronteira de decisão é onde $\theta^T x = 0$: -6 + x1 + 0 = 0 → x1 = 6. É uma linha vertical em x1=6. Para x1 > 6: $\theta^T x > 0$ → h(x) > 0.5 → prevê y=1. Para x1 < 6: prevê y=0. Portanto, y=0 à ESQUERDA de x1=6 e y=1 à DIREITA. Corresponde ao 2º gráfico (y=0 esquerda, y=1 direita, linha em x1=6).

Q12. Which statements about logistic regression training are true? (Check all that apply)

✓	**Adding a new feature always results in equal or better performance on the training set.**
✓	**Introducing regularization always results in equal or better performance on examples NOT in the training set.**
✗	Adding many new features helps prevent overfitting on the training set.
✗	Introducing regularization always results in equal or better performance on the training set.

Justificação: Opção 1 ✓: mais features = modelo mais expressivo = treino só pode melhorar ou manter. Opção 2 ✓: regularização reduz overfitting e melhora generalização. Opção 3 ✗: mais features aumenta overfitting, não o previne. Opção 4 ✗: regularização penaliza os pesos e pode piorar o erro de treino.

Q13. λ=0 vs λ=1: which θ corresponds to λ=1?

✓	**θ = [1.37, 0.51] ← λ=1 (mais regularização → pesos menores)**
✗	θ = [74.81, 45.05] ← λ=0 (sem regularização → pesos grandes)

Justificação: λ=1 aplica regularização forte que penaliza pesos grandes, forçando-os a valores menores. λ=0 não regulariza, permitindo pesos muito grandes que se ajustam perfeitamente ao treino. Portanto λ=1 → θ pequenos [1.37, 0.51].

Q14. Which statements about regularization are true? (Check all that apply)

✗	Using too large a value of λ can cause your hypothesis to OVERFIT the data.
✓	**Adding regularization may cause the classifier to incorrectly classify some training examples it previously classified correctly (when λ=0).**
✗	Because logistic regression outputs $0 \leq h(x) \leq 1$, regularization is generally not helpful for it.
✗	Using a very large value of λ cannot hurt performance; we avoid it only for numerical reasons.

Justificação: Opção 1 ✗ λ grande causa UNDERFITTING (modelo demasiado simples), não overfitting. Opção 2 ✓: regularização pode piorar o erro de treino (esse é precisamente o objetivo — evitar overfitting). Opção 3 ✗ regularização é útil para regressão logística. Opção 4 ✗ λ muito grande leva a underfitting severo — prejudica claramente o desempenho.

Q15. In which figure does the hypothesis OVERFIT the training set?

✓	**Figura 1 — curva muito complexa (ondulada) que passa por todos os pontos, incluindo ruído ✓OVERFITTING**
✗	Figura 2 — linha reta simples que não se ajusta bem aos dados ✗(seria underfitting)

Justificação: Overfitting = modelo demasiado complexo que memoriza o ruído dos dados de treino. Reconhece-se pela curva que passa exatamente por todos os pontos de treino mas teria péssima performance em dados novos.

Q16. In which figure does the hypothesis UNDERFIT the training set?

✓	**Figura 1 — linha quase plana que não captura o padrão real dos dados ✓ UNDERFITTING**
✗	Figura 2 — curva complexa que passa por todos os pontos ✗(seria overfitting)

Justificação: Underfitting = modelo demasiado simples (high bias) que não consegue capturar os padrões reais nos dados. Reconhece-se pelo erro alto tanto no treino como no teste.

Q17. Gaussian kernel: which plot corresponds to $\sigma^2=0.25$?

✓	**Plot mais estreito e alto (σ^2 menor → curva mais estreita, drop-off mais rápido) ✓**
✗	Plot mais largo e baixo (corresponderia a σ^2 maior) ✗

Justificação: σ^2 controla a largura da distribuição Gaussiana. $\sigma^2=0.25$ é menor que $\sigma^2=1$ → a curva cai mais rapidamente à medida que nos afastamos do centro → plot mais estreito e alto. σ^2 grande → curva mais larga e baixa.

Q18. How should you choose kernel and parameters (C, σ^2) for SVM?

✗	Choose whatever performs best on the training data.
✓	**Choose whatever performs best on the cross-validation data.**
✗	Choose whatever performs best on the test data.

✗	Choose whatever gives the largest SVM margin.

Justificação: Hyperparâmetros (C, σ²) devem ser escolhidos com base no conjunto de VALIDAÇÃO (cross-validation). Usar o treino leva a overfitting da escolha; usar o teste é 'batota' — o teste deve ser intocável até à avaliação final.

Q19. K-means: cost J is much higher for k=5 than k=3. What can you conclude?

✗	This is mathematically impossible. There must be a bug in the code.
✗	The correct number of clusters is k=3.
✓	**In the run with k=5, k-means got stuck in a bad local minimum. Try re-running with multiple random initializations.**
✓	**In the run with k=3, k-means got lucky. Try re-running k=3 with different initializations until it performs no better than k=5.**

Justificação: Matematicamente, k=5 pode ter J maior que k=3 se ficou preso num mínimo local (K-means não garante o mínimo global). A solução é correr K-means várias vezes com inicializações aleatórias diferentes e escolher a melhor. NÃO se pode concluir que k=3 é o número correto de clusters.

Q20. What is the recommended way to initialize K-means?

✗	Set all centroids to the same random point x(i) → BAD (symmetry problem, all clusters identical)
✗	Pick k distinct random integers from {1,...,k} → índices inválidos (k pode ser menor que m)
✓	**Pick k distinct random integers i1,...,ik from {1,...,m}. Set μ1=x(i1), ..., μk=x(ik)** ✓
✗	Set every element of μ to a random value between -ε and ε → NOT recommended

Justificação: O método recomendado é selecionar aleatoriamente k exemplos do conjunto de treino e usá-los como centróides iniciais. Isso garante que os centróides começam em locais realistas e distintos no espaço de dados.

Q21. Which are good/recommended applications of PCA? (Select all that apply)

✓	**To compress the data so it takes up less computer memory / disk space.** ✓
✓	**To reduce the dimension of the input data to speed up a learning algorithm.** ✓
✗	Instead of using regularization, use PCA to reduce features to reduce overfitting. ✗
✓	**To visualize high-dimensional data (by choosing k=2 or k=3).** ✓

Justificação: PCA é útil para compressão, speedup e visualização. NÃO é substituto de regularização — PCA não resolve overfitting da mesma forma e pode até perder informação importante. Para overfitting, a solução correta é regularização (Ridge/Lasso).

Q22. Which vectorized implementation correctly computes a(2)?

✓	**a2 = sigmoid(Theta1 * x) ✓— multiplicação matricial correta**
✗	a2 = sigmoid(x * Theta1) ✗— ordem errada (dimensões incompatíveis)
✗	a2 = sigmoid(Theta2 * x) ✗— matriz errada (Theta2 é da camada 2→3)
✗	z = sigmoid(x); a2 = Theta1 * z ✗— sigmoid aplicado antes da multiplicação

Justificação: O forward pass calcula: z = Theta1 × x (produto matricial), depois a = sigmoid(z). Em forma vetorizada: a2 = sigmoid(Theta1 * x). Theta1 tem dimensão 3×3 (3 nós hidden × 3 inputs com bias), x tem dimensão 3×1.

Q23. $J(\theta) = 2\theta^3 + 2$, $\theta=1$, $\varepsilon=0.01$. Compute $[J(\theta+\varepsilon) - J(\theta-\varepsilon)] / 2\varepsilon$

✗	5.9998
✗	6
✓	**6.0002** ✓
✗	8

Justificação: $J(1.01) = 2\times(1.01)^3 + 2 = 2\times1.030301 + 2 = 4.060602$. $J(0.99) = 2\times(0.99)^3 + 2 = 2\times0.970299 + 2 = 3.940598$. $[4.060602 - 3.940598] / (2\times0.01) = 0.120004 / 0.02 = 6.0002$. (A derivada exata é $dJ/d\theta = 6\theta^2 = 6\times1 = 6$, a aproximação numérica dá 6.0002.)

Q24. Does the KNN classifier need the training set during the test phase? Justify.

Justificação: SIM. O KNN é um algoritmo 'lazy' (sem fase de treino real) — não aprende um modelo explícito. Em vez disso, armazena TODOS os exemplos de treino. Durante o teste, para classificar um novo ponto, o KNN calcula a distância a todos os exemplos de treino, encontra os K mais próximos e prevê a classe maioritária. Sem o conjunto de treino, o KNN não consegue fazer qualquer previsão.

Q25. Gradient descent plots A, B, C correspond to which learning rates (α=0.01, 0.1, 1)?

✗	A com α=0.01, B com α=0.1, C com α=1
✗	A com α=0.1, B com α=0.01, C com α=1

✗	A com α=1, B com α=0.01, C com α=0.1
✓	**A com α=1, B com α=0.1, C com α=0.01** ✓

Justificação: Plot A diverge/oscila → α=1 demasiado grande (ultrapassa o mínimo). Plot B converge rapidamente → α=0.1 valor ideal. Plot C converge muito lentamente → α=0.01 demasiado pequeno. Regra: α grande = convergência rápida mas instável; α pequeno = convergência lenta mas estável.

Q26. Check all that apply regarding backpropagation:

✓	**It can be stacked into poor local minima.** ✓
✗	It does not require labeled data. ✗
✗	It is very slow in networks with multiple hidden layers. ✗
✗	It cannot be applied to multilayer perceptron. ✗

Justificação: Backpropagation pode ficar em mínimos locais porque a função de custo das redes neuronais não é convexa (✓). Requer dados rotulados — calcula o erro entre previsão e label real (✗opção 2). É eficiente mesmo com múltiplas camadas — foi desenhado precisamente para isso (✗opção 3). É o algoritmo padrão para treinar multilayer perceptrons (✗opção 4).

Q27. Which ML architectures are related with deep learning? (Check all that apply)

✓	**Sparse Stacked Autoencoder** ✓**— arquitetura deep learning**
✓	**Multilayer Perceptron** ✓**— rede neuronal profunda com múltiplas camadas**
✓	**Convolution Neural Network (CNN)** ✓**— arquitetura deep learning clássica**
✗	Reinforcement Learning ✗— paradigma de aprendizagem, não uma arquitetura deep learning

Justificação: Sparse Stacked Autoencoder, MLP e CNN são arquiteturas de deep learning. Reinforcement Learning é um paradigma de aprendizagem (pode usar deep learning — ex: Deep Q-Network — mas não é em si uma arquitetura de deep learning).

Q28. Which statements about Softmax Regression are true? (Check all that apply)

✓	**Softmax Regression is a supervised learning algorithm.** ✓
✗	Softmax Regression is a binary classifier. ✗
✓	**Softmax Regression is more suitable than Logistic Regression for mutually exclusive classes.** ✓
✗	The gradient descent cannot be applied for Softmax Regression. ✗

Justificação: Softmax é supervisionado (✓). NÃO é binário — é a generalização da regressão logística para K classes (✗). É mais adequado para multiclasse com classes mutuamente exclusivas (✓). Gradient descent aplica-se normalmente (✗opção 4).

Q29. m=14 examples, n=3 features. Dimensions of X, y and θ?

✓	**A. X=14×4, y=14×1, θ=4×1** ✓
✗	B. X=14×3, y=14×1, θ=3×3 ✗
✗	C. X=14×3, y=14×1, θ=3×1 ✗
✗	D. X=14×4, y=14×4, θ=4×4 ✗

Justificação: X tem dimensão m×(n+1) = 14×4 porque inclui a coluna de bias (x0=1). y tem dimensão 14×1 (um output por exemplo). θ tem dimensão (n+1)×1 = 4×1 (um peso por feature + bias). Multiplicação: X×θ = (14×4)×(4×1) = 14×1 ✓.

Q30. Propose a solution to treat the problem of class unbalanced data.

O desequilíbrio de classes ocorre quando uma classe é muito mais frequente que outra (ex: 95% vs 5%), tornando a accuracy uma métrica enganadora. As soluções são:

A nível dos dados:

- Oversampling: duplicar ou criar amostras sintéticas da classe minoritária
- SMOTE (Synthetic Minority Over-sampling Technique): criar exemplos sintéticos interpolando entre exemplos existentes da classe minoritária — mais eficaz que duplicação simples
- Undersampling: remover amostras da classe maioritária para equilibrar as classes

A nível do algoritmo:

- class_weight='balanced': penalizar mais os erros na classe minoritária durante o treino
- Ajustar o threshold de decisão (ex: baixar de 0.5 para 0.3) para aumentar o Recall da classe minoritária

A nível das métricas:

- Usar F1-Score, AUC-ROC ou Balanced Accuracy em vez de accuracy simples
- Analisar a matriz de confusão e focar no Recall da classe minoritária