TÓPICOS DE APRENDIZAGEM AUTOMÁTICA

Exame – Respostas

Licenciatura em Informática | 2024/2025

Nome: _____	Nº: _____ Turma: _____

Questão 1. Explique a diferença entre regressão e classificação. Dê um exemplo de cada.

A principal diferença está no tipo de output que o modelo produz.

A regressão prevê valores contínuos — o output é um número real. Exemplo: prever o preço de uma casa com base na sua área e número de quartos. O modelo retorna um valor como €250.000.

A classificação prevê categorias discretas — o output é uma classe. Exemplo: detetar se um email é spam ou não spam. O modelo retorna uma das classes possíveis (spam/não spam).

Em resumo: a regressão responde à pergunta "Quanto?" enquanto a classificação responde à pergunta "Qual classe?"

Questão 2. Artificial neural network

a) Qual a operação de um "node"?

Cada nó realiza três operações sequenciais:

- Ponderação: multiplica cada input pelo seu peso — $z = \Sigma(w_i \times x_i) + b$
- Soma: soma todos os inputs ponderados mais o bias
- Ativação: aplica uma função de ativação ao resultado — $a = f(z)$. Se a saída exceder um threshold, o nó "dispara" e passa o valor para a camada seguinte

Este processo é inspirado no neurónio biológico que dispara quando o sinal acumulado ultrapassa um limiar.

b) O que é uma activation function?

Uma função de ativação é uma função matemática aplicada à saída ponderada de cada neurónio. É essencial porque introduz não-linearidade na rede — sem ela, a rede seria matematicamente equivalente a um único modelo linear, independentemente do número de camadas, sendo incapaz de aprender padrões complexos.

Exemplos principais:

- ReLU: max(0,z) — padrão nas hidden layers, computacionalmente eficiente
- Sigmoid: 1/(1+e^-z) — output entre 0 e 1, usada para classificação binária
- Softmax: para classificação multiclasse, os outputs somam 1 (probabilidades)

Questão 3. O tratamento de séries temporais carece de uma metodologia especial? Justifique e apresente estratégias.

Sim, definitivamente. Os dados de séries temporais violam o pressuposto de independência (i.i.d.) dos algoritmos tradicionais de ML. Cada observação é influenciada pelas anteriores, existindo autocorrelação — ignorar esta dependência temporal produz modelos que falham em capturar padrões essenciais.

Principais desafios:

- Dependência temporal: os dados não são independentes — o valor de hoje depende do de ontem
- Sazonalidade: padrões que se repetem em intervalos fixos (diário, semanal, anual)
- Tendência: crescimento ou declínio de longo prazo
- Não-estacionaridade: média e variância podem mudar ao longo do tempo

Estratégias a adotar:

- Divisão temporal correta: NUNCA dividir aleatoriamente — treino = passado, teste = futuro
- Decomposição T+S+E: separar Tendência, Sazonalidade e Resíduo para entender a estrutura
- Feature engineering temporal: lag features, dia da semana, mês, indicadores de feriados
- Modelos específicos: ARIMA/SARIMA para séries estacionárias; LSTM/GRU para dependências longas
- Walk-forward validation: cross-validation respeitando a ordem temporal

Questão 4. Classificação binária com dados desequilibrados (90% classe A, 10% classe B)

a) A accuracy é uma métrica enganadora neste contexto? Justifique.

Sim, a accuracy é extremamente enganadora neste contexto. Um classificador que prevê sempre a classe A (maioritária) obtém 90% de accuracy sem nunca detetar a classe B — o Recall da classe B é 0%. O modelo parece bom mas é completamente inútil para o objetivo real, que é precisamente identificar os casos da classe minoritária.

b) Que outras métricas poderiam ser usadas neste contexto?

- Recall (Sensibilidade): mede quantos positivos reais foram detetados — fundamental quando não detetar a classe B é perigoso
- Precision: mede dos que o modelo classifica como B, quantos são realmente B
- F1-Score: média harmónica de Precision e Recall — ideal para datasets desequilibrados
- Balanced Accuracy: média do recall de cada classe, não é afetada pelo desequilíbrio
- AUC-ROC: avalia a capacidade geral do modelo de distinguir as classes independentemente do threshold

Para além das métricas, deve-se também tratar o desequilíbrio nos dados com técnicas como SMOTE (oversampling sintético), undersampling da classe maioritária, ou penalizar o algoritmo com class_weight.

Questão 5. Interprete a seguinte matriz de confusão e comente o desempenho do classificador.

		Previsto +	Previsto -
Real	Positivo	TP = 20	FN = 80
	Negativo	FP = 15	TN = 85

Com base na matriz: TP=20, FN=80, FP=15, TN=85. Total = 200 exemplos.

Cálculo das métricas:

- Accuracy = (20+85)/200 = 52,5% — desempenho global muito fraco
- Recall = 20/(20+80) = 20% — o modelo deteta apenas 20% dos casos positivos reais
- Precision = 20/(20+15) = 57% — dos que classifica como positivo, 57% são corretos

O principal problema é o Recall extremamente baixo — existem 80 Falsos Negativos, ou seja, o modelo falha em detetar 80% dos casos positivos reais. Se esta for uma aplicação crítica como diagnóstico médico ou deteção de fraude, o modelo é claramente insatisfatório.

Para melhorar o Recall, deve-se reduzir o threshold de decisão — o modelo passaria a classificar como positivo com menor certeza, aumentando os TPs à custa de mais FPs. Este trade-off é aceitável em contextos onde os FN têm custo elevado.

Questão 6. Alta precisão mas baixa sensibilidade — a equipa considera o desempenho satisfatório. Concorda?

Não necessariamente — depende totalmente do contexto e do custo relativo dos erros.

Alta precisão significa que quando o modelo prevê positivo, raramente erra — há poucos Falsos Positivos. Contudo, baixa sensibilidade (recall) significa que muitos casos positivos reais não são detetados — há muitos Falsos Negativos.

Se a aplicação for diagnóstico médico ou deteção de fraude, os Falsos Negativos têm custo elevado: o modelo falha em detetar doenças ou fraudes reais, o que pode ser perigoso ou financeiramente prejudicial. Neste contexto, o desempenho NÃO é satisfatório.

Se o objetivo for enviar apenas comunicações de marketing quando há alta certeza de interesse do cliente, os Falsos Positivos são mais custosos (irritam utilizadores), e alta precisão com recall moderado pode ser aceitável.

Conclusão: a avaliação correta depende sempre do problema de negócio e do custo relativo dos FP e FN. A equipa deve justificar a sua posição com base no contexto, e nunca avaliar o desempenho com base numa única métrica isolada.

Questão 7. Erros sistemáticos em determinados períodos (fim de semana, férias). O que causa e como resolver?

Os erros sistemáticos em períodos específicos indicam que o modelo não captura os padrões sazonais associados a esses períodos. As causas mais prováveis são:

- Features insuficientes: o modelo não tem variáveis que indiquem dia da semana, feriados ou época de férias
- Dados de treino desequilibrados: fins de semana e feriados têm poucos exemplos nos dados de treino, logo o modelo não aprendeu esses padrões
- O modelo assume linearidade quando a relação real nesses períodos é não-linear

Ações a tomar para melhorar o desempenho:

- Adicionar features temporais: dia da semana, é feriado (variável binária), mês, semana do ano, época de férias
- Usar modelos de séries temporais como SARIMA que modelam sazonalidade explicitamente
- Considerar decomposição multiplicativa se a amplitude sazonal varia com o nível da série
- Verificar se os dados de treino têm representação suficiente desses períodos — se não, recolher mais dados
- Analisar os resíduos do modelo nesses períodos específicos para perceber o padrão do erro

Questão 8. 100.000 registos, 50+ variáveis — prever cancelamentos de contrato. Descreva os passos principais.

O desenvolvimento deste modelo preditivo seguiria os seguintes passos:

1. Exploração e Tratamento dos Dados

- Análise exploratória: distribuições, correlações, identificar missing values e outliers
- Tratamento de valores em falta: imputação (média, mediana) ou remoção consoante a relevância
- Correção de inconsistências e encoding de variáveis categóricas
- Normalização das variáveis numéricas (Z-score ou Min-Max conforme o algoritmo)

2. Seleção de Variáveis

- Remover features não-informativas (variância muito baixa)
- Remover features redundantes/colineares (correlação alta entre si)
- Usar métodos filter (correlação com o target) ou wrapper (forward/backward selection)
- Com 50+ variáveis, a redução é importante para evitar overfitting e reduzir complexidade

3. Tratamento do Desequilíbrio de Classes

- Cancelamentos são tipicamente raros — verificar se há desequilíbrio de classes
- Se sim: aplicar SMOTE, undersampling, ou penalizar o algoritmo com class_weight

4. Escolha do Modelo

- Começar com Regressão Logística como baseline (simples e interpretável)
- Testar Random Forest e XGBoost — robustos, lidam bem com muitas features
- Comparar modelos usando cross-validation

5. Validação

- Usar K-Fold Cross-Validation para avaliação robusta
- Métricas: F1-Score, AUC-ROC e Recall — nunca só a accuracy dada a possível classe desequilibrada
- Analisar importância das features e erros sistemáticos para possível melhoria