



**UNIVERSIDADE FEDERAL DO MARANHÃO**  
**DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO**

**Prova \_\_\_\_**

<b>Curso:</b> Engenharia da Computação	<b>Ano / Semestre:</b>	<b>Data:</b>
<b>Disciplina:</b> Inteligência Artificial	<b>Professor:</b> Thales Levi Azevedo Valente	
<b>Aluno:</b>	<b>Código:</b>	

**OBS: RESPOSTAS DEVEM SER ENTREGUES À CANETA. CASO O ALUNO USE LÁPIS, O MESMO NÃO TERÁ DIREITO A REVISÃO DAS QUESTÕES**

**Descrição do Cenário**

Você foi contratado como engenheiro de machine learning por uma empresa líder no setor de telecomunicações, especializada na operação de redes de fibra óptica transmitidas por dutos subaquáticos. Esses dutos possuem uma certa flexibilidade e são vitais para a infraestrutura de comunicação global. No entanto, estão sujeitos a desafios únicos, como deformações oriundas das forças do ambiente marinho, que podem afetar negativamente sua integridade e eficácia operacional, pois redes de fibra são muito sensíveis a deformações. Sensores de alta precisão, estrategicamente instalados ao longo dos dutos para coletar dados sobre a condição física dos mesmos, incluindo a integridade dos arames internos de tração dos dutos, são essenciais para prevenir deformações excessivas. A deterioração de um único arame pode precipitar falhas exponenciais, potencializando riscos críticos ao sistema. Portanto, o monitoramento em tempo real e a análise preditiva se fazem cruciais.

Para mitigar riscos de falhas dispendiosas e reduzir a frequência de manutenções preventivas e corretivas, a empresa busca implementar um sistema avançado de monitoramento usando IA. Os dados coletados pelos sensores são armazenados em pastas e dados .csv. Não há como transmitir os dados via rede, pois vários sensores estão instalados em plataformas em alto-mar. Essas plataformas de monitoramento conectam os dutos e são gerenciadas por pessoas, pois não há como instalar dutos que atravessam continentes sem pontos de conexão. Em outras palavras, não há conectividade com a internet nas plataformas que coletam dados. Cada plataforma possui seu padrão de organização de dados e possui seu contexto ambiental.

A plataforma deve incorporar algoritmos de machine learning para analisar padrões nos dados coletados e identificar sinais precoces de possíveis falhas, permitindo intervenções proativas para manter a operacionalidade do sistema. A análise preditiva ajudará a empresa a planejar melhor suas operações de manutenção e inspeção, reduzindo custos e maximizando a eficiência. A Empresa acredita que os anos de dados coletados são suficientes para desenvolver um ótimo modelo.

**Com base na descrição do cenário baseado em uma entrevista com o cliente, responda:**

## **1. (Pipeline de Machine Learning)**

**(a)** Explique como você estruturaria o processo de classificação de falhas, em um OU mais sensores, dentre os passos de implantação do modelo (A), engenharia de características (B), pré-processamento (C), treinamento do modelo(D), aquisição dos dados(E), avaliação do modelo(F) e seleção de características(G).

**Resposta:**

**(E) Aquisição dos dados:** os dados são coletados por sensores instalados ao longo dos dutos e salvos como arquivos .csv em dispositivos físicos, já que não há conexão de rede. O primeiro passo seria adquirir esses dados brutos.

**(C) Pré-processamento:** se cada plataforma de monitoramento tem seu próprio padrão de organização, o primeiro passo seria colocar essas dados em um único padrão. Em seguida, aplicar outras técnicas como remoção de sensores defeituosos (dados nulos), Interpolação de pequenas falhas de leitura de sensor em tempos específicos, normalização, etc.

**(B) Engenharia de características:** A partir dos dados crus dos sensores, extraem-se variáveis derivadas como média, mediana, desvio padrão, entropia, entre outros.

**(G) Seleção de características:** Após gerar dezenas de variáveis, é necessário reduzir a dimensionalidade. Esse passo é opcional caso o modelo esteja overfitando.

**(D) Treinamento do modelo:** A escolha depende do volume de dados e da disponibilidade de rótulos de falha. O treinamento pode ser refeito por plataforma baseado nas localizações das mesmas ou por época do ano (devido a mudança de contexto ambiental).

**(F) Avaliação do modelo:** Como o problema é desbalanceado (poucas falhas), **acurácia não é suficiente**. Métricas importantes incluem precisão, recall e f1-score:

**(A) Implantação do modelo:** O modelo precisa ser exportado em formato leve (como .pkl, .onnx ou .tflite). A predição em tempo real deve gerar sinais claros de alerta local (ex.: arquivos de alarme ou dashboards embarcados) e logs que possam ser auditados posteriormente).

## **(b) Comente sobre a importância de garantir a qualidade dos dados**

**Resposta:**

**Modelos aprendem padrões a partir dos dados.** Se os dados forem inconsistentes, incompletos ou errados, o modelo aprenderá padrões incorretos, pois **a qualidade dos dados afeta diretamente todas as etapas do pipeline de IA**, desde o pré-processamento dos dados até a avaliação do modelo.

**(c)** Discuta quais estratégias de divisão (hold-out ou cross-validation) poderiam ser usadas no contexto de ter poucos dados e no contexto de ter muitos dados, justificando sua escolha

**Resposta:**

- Poucos**

**dados:**

A melhor abordagem é a **validação cruzada (cross-validation)**, especialmente em séries temporais:

- Se os dados forem temporais: **TimeSeriesSplit** (mantém ordem temporal, sem embaralhar);  
Vantagem: usa quase todos os dados para treino e ainda fornece estimativas de variância do desempenho.

- Muitos**

**dados:**

Um **hold-out estratificado** com 70% treino / 15% validação / 15% teste é suficiente. Deve-se lembrar de respeitar a ordem cronológica dos dados. Com bastante amostras, a variância da avaliação é pequena, e a cross-validation se torna mais custosa e desnecessária. Pode-se ainda usar uma base externa (ex.: outra plataforma) como **conjunto de teste não visto**, garantindo uma avaliação realista da capacidade de generalização do modelo.

**Resumo:**

Situação	Técnica recomendada	Justificativa
Poucos dados	Cross-validation (TimeSeriesSplit)	Maximiza uso de dados; reduz viés de partição
Muitos dados	Hold-out estratificado temporal	Mais simples e rápido; boa estimativa de generalização

### (B) (Overfitting, Underfitting e Generalização)

Em um sistema que monitora deformações em dutos subaquáticos, quais seriam as consequências de um modelo com overfitting? E de um modelo com underfitting? Explique ambos os conceitos e quais as suposições que especialistas na área de IA poderiam fazer ao treinar e testar modelos.

#### Resposta:

- **Overfitting:** Um modelo com overfitting aprende tão bem os dados de treino que passa a memorizar ruídos e padrões específicos daquela amostra.  
**Consequência no sistema de dutos:**

- No contexto de dutos subaquáticos, pode levar o modelo a **falhar ao identificar deformações reais em plataformas com condições ambientais diferentes** daquelas usadas no treinamento — justamente por ter memorizado padrões específicos ao invés de aprender padrões generalizáveis de falha.
- Isso pode resultar em **ruptura inesperada do cabo** e perda total de capacidade de transmissão.

- **Underfitting:** Ocorre quando o modelo é **simples demais** ou mal treinado e não consegue capturar nem os padrões gerais dos dados.  
**Consequência no sistema:**

- O modelo **não detecta sinais reais de falha**, ignorando tendências sutis nos sensores que indicariam deterioração progressiva.
- Isso pode resultar em **ruptura inesperada do cabo** e perda total de capacidade de transmissão.

#### **- Suposições de especialistas**

1. Analisar curvas de aprendizado para verificar se o modelo está evoluindo (treino vs validação).
2. Avaliar generalização do modelo em contextos diversos (testes usando dados de outras plataformas).
3. Utilizar métricas robustas como F1-score, recall, especialmente em casos de falha rara ao invés de usar a acurácia
4. **Modelo mais complexo ≠ melhor.** A complexidade deve estar alinhada à quantidade e à variabilidade dos dados. É bom experimentar primeiros modelos mais simples e só depois tomar decisões baseado nas observações das curvas de aprendizado.
5. **Pré-processamento e engenharia de atributos são parte crítica.** Não se espera que o modelo “descubra tudo sozinho”.

#### **- Erros comuns de estagiários em IA**

1. **Avaliar o modelo com métricas inadequadas**, como acurácia em problemas desbalanceados.
2. **Assumir que métricas no treino é sinal de sucesso**, sem avaliar generalização no conjunto de teste.

3. Escolher modelos complexos demais logo no início (ex: redes neurais profundas) mesmo com poucos dados.

4. Ignorar o pré-processamento, achando que o modelo “vai aprender tudo sozinho”

**(C) Métricas e Classes Desbalanceadas**

No caso de detecção de falhas (classificação binária: “falha” vs. “não falha”), suponha que a maioria das amostras coletadas indicam funcionamento normal, e apenas uma minoria contenha falhas. Por que a acurácia sozinha pode ser enganosa? Quais métricas adicionais seriam mais adequadas para avaliar o modelo e por quê?

**Resposta:** Se 99 % das instâncias são Não falha, um classificador “sempre não falha” atinge 99 % de acurácia e zero utilidade.

Métricas mais adequadas:

- **Recall (Sensibilidade)** – chance de captar uma falha verdadeira.
- **F1-score** – após conseguir capturar as falhas, o próximo passo seria reduzir falsos positivos. A média harmônica do recall com a precisão ajudaria na busca por um equilíbrio, mas sempre priorizando não comprometer a métrica recall. Falsos negativos é mais crítico que falsos positivos.

**(D) Softmax e Entropia Cruzada**

Em um cenário onde o sistema de monitoramento classifica múltiplos tipos de falha (por exemplo, falha de arame interno, ruptura parcial, falha por vibração excessiva, etc.), qual função de ativação é a mais adequada na camada de saída?. Como a entropia cruzada (categorical cross-entropy) lida com erros de previsão que sejam feitos com “alta confiança” em classes incorretas? Dica: lembrar como o log penaliza e do one-hot encoding.

**(E) Inspiração Biológica e Neurônios Artificiais**

Relacione o funcionamento básico de um neurônio biológico (dendritos, soma, axônio, sinapses) com o neurônio artificial em uma rede. Destaque como a soma ponderada das entradas e a função de ativação permitem modelar a ideia de excitação ou inibição do neurônio.

- Dendritos recebem sinais  $x_i$ .
- As sinapses modulam a força — em RNAs os pesos  $w_i$ .
- A soma no corpo celular (soma) corresponde à soma ponderada  $z = \sum w_i x_i + b$
- Quando z ultrapassa o limiar, o axônio dispara; na RNA usa-se função de ativação  $\sigma(z)$  (ReLU, sigmoid).
- Pesos positivos  $\leftrightarrow$  sinapses excitatórias, pesos negativos  $\leftrightarrow$  inibitórias.

**(F) Projeto de Redes Neurais Artificiais**

Explique como seria uma rede recorrente para realizar a previsão de como os sinais de sensores se comportariam no dia d+1, considerando a quantidade fixa de 5 sensores. Você pode combinar textos e diagramas para explicar. Dê ênfase em informações como tipo de arquitetura, camadas da rede, quantidades de neurônios em cada camada, funções de ativação utilizadas, forma de inicialização dos pesos e função de perda. Explique o funcionamento e as decisões tomadas.

**Resposta:**

**Entrada da rede:** sequência de leituras anteriores dos 5 sensores (5 neurônios, um para cada sensor)

**Arquitetura:**

- 1 ou 2 camadas intermediárias com alguma potência de 2 de neurônios (64, 128, 256, etc.)
- 1 camada densa (Dense) de 5 neurônios para gerar a previsão dos 5 sensores no dia seguinte (1 neurônio para cada sensor)

**Ativações:**

- intermediárias: tanh
- Saída: linear (valores reais)

**Função de perda:** MAE (erro absoluto médio) é mais habitual para esse tipo de problema. Se quisermos dar maior penalidade a erros grandes, usar MSE como função de perda combinado com RMSE para avaliação das previsões.

**Inicialização dos pesos:** pode ser de forma aleatória para experimentos iniciais ou alguma mais esperada

**Saída da rede:** matriz com as previsões dos 5 sensores ao longo do próximo dia

**Uso prático:** prever comportamentos anormais → gerar alertas antes da falha

**(G) Perceptron**

Um perceptron de duas entradas define a sua fronteira de decisão por meio da equação  $w_1x_1 + w_2x_2 + b_{perc} = 0$ , onde  $x_1$  e  $x_2$  são as entradas,  $w_1$  e  $w_2$  os pesos sinápticos e  $b_{perc}$  o bias do perceptron. A equação da reta é definida por meio da equação  $y = ax + b$ . Tome  $y = x_2$  e  $x = x_1$ . Você quer plotar a animação do custo da rede a cada iteração sobre a superfície e sabe que não pode usar a equação do perceptron, mas pode usar a equação da reta. Desenvolva uma função def w2ab(w1, w2, b) que retorne os valores de  $a$  e  $b$  da reta. Você pode usar python, pseudocódigo ou uma explicação matemática para responder essa questão. Dica: Se não souber a fórmula de cabeça, você pode desenvolvê-la primeiramente colocando tudo em função de  $x_2$ .

**Resposta:**

$$w_1x_1 + w_2x_2 + b_{perc} = 0$$

$$w_2x_2 = w_1x_1 - b_{perc}$$

$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{b_{perc}}{w_2}, \text{ comparando com } y = ax + b \text{ temos } a = -\frac{w_1}{w_2} \text{ e } b = -\frac{b_{perc}}{w_2}$$

def w2ab(w1, w2, b):

    a = -w1 / w2

    intercept = -b / w2

    return a, intercept