Enunciado

A BioFermenta S.A. deseja garantir que o modelo logístico para prever a eficiência das bateladas não esteja sofrendo com sobreajuste. Para isso, a equipe estatística propõe aplicar técnicas de reamostragem que estimem a estabilidade dos resultados, como validação cruzada e bootstrap.

Resposta modelo

Etapa 1 – Entendimento do Problema

Após construir modelos preditivos para estimar rendimento e tempo de fermentação, a equipe da BioFermenta levantou uma nova preocupação: o modelo logístico ajustado pode estar sobreajustado (overfitting) — ou seja, pode estar "decorando" o comportamento da base de treino, sem generalizar bem para novos dados.

Meu objetivo, como analista de dados da equipe de engenharia de processos, foi avaliar a robustez e estabilidade do modelo logístico já ajustado para prever se uma batelada será eficiente (1) ou ineficiente (0), utilizando técnicas de reamostragem estatística (validação cruzada e bootstrap).

Etapa 2 – Preparação e Diagnóstico Inicial

Antes de aplicar os métodos de reamostragem, conferi com apoio da IA a consistência dos dados e a necessidade de padronização das variáveis contínuas (pH, glicose, temperatura e agitação). A IA sugeriu o uso de `StandardScaler` dentro de um pipeline do scikit-learn, garantindo que a escala das variáveis não influenciasse de forma desproporcional os coeficientes do modelo logístico.

Também validei, que a divisão entre treino e teste deveria ser estratificada, dado o desbalanceamento entre bateladas eficientes e ineficientes.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42
)
```

A IA me explicou que o parâmetro `stratify=y` assegura que a proporção entre classes (eficiente/ineficiente) se mantenha igual nas amostras de treino e teste — essencial para modelos de classificação binária.

Etapa 3 – Aplicação da Validação Cruzada

A partir daí, tentei entender como a validação cruzada poderia estimar a variabilidade do desempenho do modelo. Optei por usar esses dois métodos:

- K-Fold (5 dobras): Nesse método, o conjunto de dados foi dividido em cinco partes aproximadamente iguais. Em cada rodada, o modelo foi treinado em quatro partes e testado na quinta, repetindo o processo cinco vezes, de forma que todas as amostras serviram como teste uma vez. A acurácia final é a média das cinco execuções, refletindo o desempenho médio esperado em dados novos. O K-Fold é computacionalmente eficiente e fornece uma estimativa estável do desempenho, equilibrando bem viés e variância. Por isso, é amplamente utilizado em contextos industriais, onde há volume razoável de dados e necessidade de análises rápidas.
- Leave-One-Out Cross-Validation (LOOCV): O Leave-One-Out é uma forma extrema de K-Fold, em que cada amostra individual é usada como teste uma única vez, e o restante do conjunto (n−1 amostras) serve para o treino. Assim, se a base possui 5.000 registros, o modelo é ajustado 5.000 vezes — uma para cada ponto. Essa estratégia tem viés muito baixo, pois quase todos os dados são usados em cada treinamento, mas apresenta alta variância, sendo mais sensível a pequenas alterações na base e computacionalmente mais custosa.

Com base nisso, implementei o seguinte código:

```
from sklearn.model_selection import cross_val_score, StratifiedKFold
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
modelo_log = Pipeline([
    ('scaler', StandardScaler()),
    ('logreg', LogisticRegression(max_iter=1000))
])
scores = cross_val_score(modelo_log, X_train, y_train, cv=cv, scoring='accuracy')
print(f"Acurácia média: {scores.mean():.3f} ± {scores.std():.3f}")
```

O resultado mostrou uma acurácia média de 0,89 ± 0,03, o que indica bom desempenho e baixa variabilidade — sinal de que o modelo generaliza bem e não está sobreajustado.

Já o desvio-padrão (±0,03) representa a oscilação entre as dobras, valores baixos reforçam a estabilidade da performance.

Após avaliar o desempenho do modelo logístico por meio da reamostragem, foram comparados dois métodos distintos de validação cruzada: K-Fold (5) e Leave-One-Out (LOOCV). A tabela abaixo resume os resultados obtidos:

Validação Cruzada - Reamostragem

	Método	Acurácia Média
0	K-Fold (5)	0.879778
1	Leave-One-Out	0.880795

A diferença entre os métodos foi mínima, o que indica consistência e estabilidade do modelo — evidenciando que não há sinais relevantes de sobreajuste.

- O K-Fold tem menor custo computacional e menor variância;
- O Leave-One-Out usa quase todos os dados a cada iteração, mas tende a ter alta variância e custo elevado.

Como o desempenho foi praticamente idêntico, concluiu-se que o modelo logístico é robusto e não sofre de sobreajuste, podendo utilizar o K-Fold (5) como método-padrão de validação na BioFermenta, por ser mais eficiente e suficientemente confiável.

Etapa 4 – Aplicação do Bootstrap

Para aprofundar a análise de estabilidade, apliquei o método bootstrap. Esse método cria múltiplas amostras aleatórias (com reposição) da base de treino e reestima o modelo em cada uma, permitindo medir a variação nos coeficientes.

O código adaptado ficou assim:

```
import numpy as np

n_iter = 1000
coef_list = []

for i in range(n_iter):
    sample_idx = np.random.choice(len(X_train), len(X_train), replace=True)
    X_sample = X_train.iloc[sample_idx]
    y_sample = y_train.iloc[sample_idx]

    modelo_log.fit(X_sample, y_sample)
    coef_list.append(modelo_log.named_steps['logreg'].coef_[0])

coef_mean = np.mean(coef_list, axis=0)
coef_std = np.std(coef_list, axis=0)
for var, mean, std in zip(X.columns, coef_mean, coef_std):
    print(f"{var}: {mean:.4f} ± {std:.4f}")
```

A IA me explicou que:

- coef_mean mostra a tendência média do peso de cada variável no modelo.
- coef_std mostra a instabilidade: quanto menor o desvio, mais estável é o efeito daquela variável sobre a probabilidade de eficiência.

Os resultados indicaram maior estabilidade para as variáveis pH e Glicose, e ligeira variação para Agitação, o que faz sentido técnico — já que a agitação é uma variável mais sensível a ruídos operacionais.

O resultado do notebook gerou a seguinte tabela de **erros-padrão** (variabilidade dos coeficientes):

Erro Padrão via Bootstrap

	Coeficiente	Erro Padrão (Bootstrap)
0	const	0.931468
1	Temperatura (°C)	0.013169
2	Tempo (h)	0.010105
3	pH	0.102571
4	Glicose (g/L)	0.001955
5	Agitação (rpm)	0.000892

Podemos concluir que quanto menor o erro-padrão, mais estável e confiável é o coeficiente. Os resultados mostraram baixa variabilidade em quase todas as variáveis, especialmente Glicose e Agitação, o que indica impactos consistentes no modelo.O pH, por outro lado, apresentou a maior flutuação (≈0,10), coerente com a sensibilidade química do processo fermentativo a variações nessa variável.

Essa etapa demonstrou que o modelo é estatisticamente estável, e que os coeficientes se mantêm próximos mesmo em diferentes subconjuntos de amostras.

Etapa 5 – Reflexão sobre o Uso da IA

Durante todo o processo, a IA atuou como um copiloto estatístico e conceitual. Ela:

- Explicou o propósito e as diferenças entre validação cruzada e bootstrap.
- Me ajudou a ajustar parâmetros (`random_state`, `shuffle`, `max_iter`) para evitar viés e travamentos.
- Propôs formas de interpretar o desvio-padrão dos coeficientes como medida de robustez do modelo.

O uso da IA foi fundamental para traduzir conceitos estatísticos complexos em práticas aplicáveis ao contexto da engenharia de processos, reforçando a integração entre ciência de dados e domínio técnico da fermentação.

Etapa 6 – Conclusão

As técnicas de reamostragem mostraram que o modelo logístico da BioFermenta apresenta **boa estabilidade e generalização**, sem indícios de sobreajuste. A análise confirmou que pequenas flutuações nos dados não comprometem a precisão nem a interpretação das variáveis-chave do processo fermentativo.

Assim, a equipe pode seguir utilizando o modelo para prever a eficiência das bateladas com **confiabilidade estatística e respaldo técnico**, consolidando o uso de inteligência artificial como ferramenta de apoio à decisão industrial.