mentoria zeroum

RESUMO COMPLETO CGU · P3 · AFFC TI, CIÊNCIA DE DADOS

CICLO DE VIDA E MLOPS

Operação, Governança e Gestão de Soluções de Dados e IA · Tópico 1

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

Sobre o seu exemplar

Este exemplar é seu. Ele sai com o seu nome e o seu CPF em todas as páginas, e é assim que eu mantenho o material fora dos grupos de rateio. É por isso que ele não tem senha nem bloqueio: imprima, destaque, anote à mão, leve para onde você estuda.

O Resumo Completo CGU é obra protegida pela Lei 9.610 de 1998, e a licença é individual, vinculada ao seu CPF. Ela não cobre revenda, rateio, nem publicação em grupo ou nuvem compartilhada.

Se você encontrar este arquivo circulando com o nome de outra pessoa, me avise em contato@mentoriazeroum.com.br.

Se encontrar um erro, use o botão abaixo. O relato chega direto a mim, com o nome deste resumo já preenchido, e cada correção que proceder entra na próxima versão do arquivo.

Reportar um erro

VERSÃO 1 · PRIMEIRA VERSÃO EM OUTUBRO DE 2026

Esta é a primeira versão do Resumo Completo CGU. Cada relato de erro é conferido na fonte e, quando procede, a correção sai numa versão nova deste arquivo, registrada abaixo.

ATUALIZAÇÕES DESTE RESUMO

- Nenhuma até agora. Quando houver correção, ela entra aqui, com a data e o que mudou.

Bons estudos.

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

RESUMO COMPLETO CGU · P3 · AFFC TI, CIÊNCIA DE DADOS

Ciclo de vida e MLOps

Operação, Governança e Gestão de Soluções de Dados e IA · Tópico 1. 1.1 Versionamento de dados, código e modelos. 1.2 Registro de experimentos. 1.3 Pipelines, testes e integração e entrega contínuas. 1.4 Reprodutibilidade. 1.5 Disponibilização por APIs, implantação e reversão. 1.6 Monitoramento de mudanças nas distribuições, nas relações e no desempenho.

COMO ESTE MATERIAL ESTÁ ORGANIZADO

1. **O trecho do edital**, logo acima, e o **mapa do tópico**, com cada item do edital, a seção deste material que o cobre, o curso para aprofundar e a fonte oficial.
2. **O texto**, no recorte exato do edital, na ordem dos itens.
3. **Os exemplos comentados**, com a consulta, o código ou o cálculo e a leitura do resultado. **As citações literais**, conferidas no texto oficial, com o artigo ou a página de onde saíram.
4. **Os quadros Como o Cebraspe pode cobrar e Pontos importantes**, com as pegadinhas da banca no tema e o essencial do tópico.
5. **As caixas pretas do Estratégia**, com as páginas de cada aula sobre o assunto.
6. **As questões abertas**, com as respostas na página seguinte, e **as questões de certo ou errado**, com o gabarito comentado.
7. **A documentação oficial e as normas** do tópico, com os links.
8. **Os cadernos do TEC** do tópico, com os assuntos que cada um cobra.

SUMÁRIO

Mapa do tópico	4
1. O ciclo de vida do modelo e o MLOps	5
2. Versionamento de dados, código e modelos	7
3. Registro de experimentos	9
4. Pipelines, testes e integração e entrega contínuas	11
5. Reprodutibilidade	14
6. Disponibilização por APIs, implantação e reversão	16
7. Monitoramento de mudanças nas distribuições, nas relações e no desempenho	19
Questões abertas de fixação	23
Respostas das questões abertas	25
Questões de certo ou errado	26
Gabarito comentado	28
Documentação oficial, normas e fontes	29
Atividades sugeridas do curso e do TEC neste tópico	29

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

Para situar este tópico: A separação entre treino, validação e teste e o vazamento de informação estão no tópico 2 da matéria Aprendizado de Máquina e IA; as métricas, no tópico 3. Custo, latência, disponibilidade e observabilidade da solução em produção são o tópico 2 desta matéria, e governança, revisão humana e proteção de dados, o tópico 3. Aqui o foco é o caminho do modelo, do experimento à produção, e o acompanhamento depois dela. Os exemplos usam um modelo fictício de triagem de pagamentos a fornecedores, com dados gerados pelo código da seção 1.

Mapa do tópico

Item do edital	Neste material	Curso para aprofundar	Fonte oficial
1.1 Versionamento de dados, código e modelos	Seções 1 e 2	Ciência de dados e IA (Ciência de Dados), cap. 8, p. 3 a 10 e 15	NC 21/IN01/DSIC/GSIPR (resumo criptográfico); SGD, Guia Unificado de IA, p. 56 (linhagem); MLflow, Model Registry; Enap, curso O papel do DevOps na Transformação Digital dos Serviços Públicos (EVG), módulo 3, p. 17 e 18 (controle de versão com Git)
1.2 Registro de experimentos	Seção 3	Ciência de dados e IA (Ciência de Dados), cap. 8, p. 10 e 24 a 25	SGD, Guia Unificado de IA, p. 45 e 46; MLflow, Tracking
1.3 Pipelines, testes e integração e entrega contínuas	Seção 4	Ciência de dados e IA (Ciência de Dados), cap. 8, p. 8 a 13	Portaria SGD/MGI 1.070/2023, Anexo I, item 4.1.7 (DevSecOps); Google Cloud, MLOps: pipelines de entrega contínua e automação (pt-BR); Python, unittest; Enap, curso O papel do DevOps na Transformação Digital dos Serviços Públicos (EVG), módulo 1, p. 13 e 14, e módulo 3, p. 4 a 13
1.4 Reprodutibilidade	Seção 5	Ciência de dados e IA (Ciência de Dados), cap. 8, p. 13, 19 e 25 a 26	SGD, Guia Unificado de IA, p. 48 e 50; Python, venv; scikit-learn, Common pitfalls (aleatoriedade)
1.5 Disponibilização por APIs, implantação e reversão	Seção 6	Ciência de dados e IA (Ciência de Dados), cap. 8, p. 13 e 20 a 22. Contrato da API, azul-verde e sombra: só neste material	IN GSI/PR 3/2021, arts. 28 a 36 (gestão de mudanças); SGD, Guia Unificado de IA, p. 52, 53 e 56; Google SRE, Canarying Releases
1.6 Monitoramento de mudanças nas distribuições, nas relações e no desempenho	Seção 7	Ciência de dados e IA (Ciência de Dados), cap. 8, p. 13 a 14 e 21 a 22. Testes de Kolmogorov-Smirnov e PSI: só neste material	SGD, Guia Unificado de IA, p. 55; Portaria Normativa CGU 179/2024, art. 1º, § 2º, VII e VIII; Google, Curso de ML: monitoramento de pipelines (pt-BR)

Páginas das fontes e dos cursos: as do PDF, como aparecem no leitor de PDF.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

1 O ciclo de vida do modelo e o MLOps

Um modelo de aprendizado de máquina em produção depende de três coisas que mudam por motivos diferentes: os **dados**, o **código** que prepara os dados, treina e serve o modelo, e o **modelo** em si, isto é, os parâmetros aprendidos. Mudar qualquer uma delas pode mudar as previsões. Por isso o modelo que funcionava no notebook pode falhar ao ser integrado aos sistemas do órgão, e o que funcionava em produção pode se degradar sem que ninguém mexa no código: os dados do mundo real mudaram.

MLOps (machine learning operations) é a cultura e o conjunto de práticas que levam ao aprendizado de máquina o que o DevOps fez com o software: unir o desenvolvimento do modelo (Dev) e a sua operação (Ops), com automação, versionamento, testes, integração e entrega contínuas e monitoramento, ao longo de todo o ciclo de vida. Em relação ao DevOps tradicional, a validação alcança também os **dados** e o **modelo**; a entrega pode ser um **pipeline de treinamento** inteiro, que gera e publica o serviço de predição; existe o **treinamento contínuo** (CT), próprio do ML; e o **monitoramento contínuo** pesa mais, porque o modelo perde desempenho quando a realidade se afasta dos dados de treino. O ciclo de vida vai da definição do problema e dos critérios de sucesso aos dados, à experimentação, à validação, à implantação, à operação com monitoramento e, por fim, ao retreino ou à desativação do modelo, com os registros preservados.

Nível de maturidade (Google Cloud)	Características
Nível 0: processo manual	Etapas feitas à mão, em notebooks; o modelo treinado é entregue a outra equipe para implantar; versões raras; sem CI, sem CD e sem monitoramento ativo
Nível 1: automação do pipeline de ML	O pipeline de treinamento roda sozinho com dados novos (treinamento contínuo), com validação automática de dados e de modelo
Nível 2: CI/CD do pipeline	A construção, o teste e a implantação do próprio pipeline também são automatizados

As práticas contínuas costumam ser resumidas como CI (integração contínua), CD (entrega contínua), CT (treinamento contínuo) e CM (monitoramento contínuo).

Os exemplos usam o código abaixo, que gera lotes fictícios de pagamentos a fornecedores: valor (em milhares de reais), número de aditivos do contrato, indicação de fornecedor novo e o rótulo `irregular`, que a auditoria confirma depois. Os parâmetros `escala_valor` e `peso_aditivos` simulam, na seção 7, mudança na distribuição e mudança na relação.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

CÓDIGO COMUM AOS EXEMPLOS (RODA ANTES DE CADA UM)

```
import hashlib, json
import numpy as np
import pandas as pd

def gera_lote(n, semente, escala_valor=1.0, peso_aditivos=0.8):
    """Pagamentos fictícios: valor (R$ mil), aditivos do contrato, fornecedor
    novo (0 ou 1) e o rótulo irregular, confirmado depois pela auditoria."""
    rng = np.random.default_rng(semente)
    valor = np.round(rng.lognormal(4.0, 0.8, n) * escala_valor, 2)
    aditivos = rng.poisson(1.2, n)
    novo = rng.binomial(1, 0.3, n)
    logito = -3.5 + 0.01 * valor + peso_aditivos * aditivos + 0.9 * novo
    irregular = rng.binomial(1, 1 / (1 + np.exp(-logito)))
    return pd.DataFrame({'valor': valor, 'aditivos': aditivos,
                         'fornecedor_novo': novo, 'irregular': irregular})
```

ESTRATÉGIA · APROFUNDE AQUI · P. 3 A 9

Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · Capítulo 8 (Ciclo de vida e MLOps)

Foco: o conceito de MLOps, as fases de design, experimentação e operação e os níveis de automação.

https://concursos.estrategia.com/ldi/ciencia-de-dados-e-inteligencia-artificial-para-cgu-auditor-federal-de-financas-e-controle-ti-ciencias-de-dados-pos-edital

SE O SEU CURSO FOR OUTRO

Gran (vídeo): disciplina "Operação, Governança e Gestão de Soluções de Dados e IA", vídeo "MLOps"

Direção (livro digital): módulo "Operação, Governança e Gestão de Soluções de Dados e IA", aula 1 (Ciclo de vida e MLOps, parte 1 de 2), capítulo "MLOps"

PONTOS IMPORTANTES

- A mudança nos dados, no código ou no modelo muda o comportamento do sistema de ML.
- O MLOps aplica as práticas de DevOps ao ML e acrescenta a validação dos dados e do modelo, o treinamento contínuo e o monitoramento do desempenho.
- No nível 0, tudo é manual; no nível 1, o pipeline de treinamento é automatizado; no nível 2, o próprio pipeline passa por CI/CD.

COMO O CEBRASPE PODE COBRAR

- Afirmar que a automação do MLOps se limita ao treinamento: ela percorre da ingestão dos dados à implantação e ao monitoramento.
- Dizer que o modelo validado mantém o desempenho enquanto o código não mudar.
- Atribuir ao nível 0 de maturidade o treinamento contínuo automatizado, que é a marca do nível 1.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

2 Versionamento de dados, código e modelos

Versionar é guardar cada estado de um artefato com um identificador estável, para saber o que estava em uso em cada momento e voltar a qualquer estado anterior. Em ML, os três artefatos pedem ferramentas diferentes:

Artefato	Como se versiona	O que identifica a versão
Código	Git: commits, ramos, etiquetas e revisão por pull request	O identificador do commit
Dados	Cópias imutáveis; o DVC guarda no Git só um arquivo pequeno com o hash do conjunto, que fica num armazenamento remoto; tabelas de lakehouse (Delta Lake, Iceberg) guardam o histórico e permitem consultar a tabela como estava numa data	O hash do conteúdo ou o número da versão da tabela
Modelo	Registro de modelos (model registry): versões numeradas, métricas, etiquetas e linhagem (a execução, os dados e o código que geraram cada versão)	O número da versão; um apelido (alias), como `producao`, aponta para a versão em uso

Versionam-se também a configuração, os hiperparâmetros, o esquema dos dados, a definição dos atributos e o ambiente (seção 5). A versão publicada não se altera, e a correção gera uma versão nova. Promover ou reverter um modelo é mudar o apelido, sem retreinar.

O **hash** funciona como impressão digital do conteúdo: dois arquivos com o mesmo hash SHA-256 são, na prática, idênticos, e a alteração de um único caractere muda o hash inteiro. A norma do GSI usa o mesmo recurso para provar que um arquivo não foi alterado:

> **CITAÇÃO LITERAL**
>
> *"Resumo Criptográfico: é um método criptográfico que, quando aplicado sobre uma informação, independente do tamanho desta, gera um resultado único e de tamanho fixo, também chamado de "hash"."*
>
> NC 21/IN01/DSIC/GSIPR, item 4.18 · abrir a fonte

O Guia Unificado de IA da Secretaria de Governo Digital (2026) põe esse registro entre os resultados de cada retreino do modelo, na etapa de sustentação:

> **CITAÇÃO LITERAL**
>
> *"Registro completo da **linhagem do novo modelo**, incluindo **ID único**, dataset utilizado, hiperparâmetros e métricas de validação."*
>
> SGD, Guia Unificado de IA (2026), fase 6.2, resultado Logs de retreinamento e versão, p. 56 · abrir a fonte

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

EXEMPLO COMENTADO · A IMPRESSÃO DIGITAL DOS DADOS E A FICHA DA VERSÃO DO MODELO

```
import pickle
from sklearn.linear_model import LogisticRegression
resumo = lambda b: hashlib.sha256(b).hexdigest()[:16]   # SHA-256, 16 dígitos

dados = gera_lote(1000, semente=7)
alterado = dados.copy()
alterado.loc[0, 'valor'] += 0.01                # dez reais a mais num pagamento
for nome, df in [('versão 1', dados), ('uma alteração', alterado),
                 ('versão 1, de novo', dados)]:
    print(f'dados, {nome:17}', resumo(df.to_csv(index=False).encode()))

X, y = dados.drop(columns='irregular'), dados['irregular']
modelo = LogisticRegression(C=1.0, max_iter=1000).fit(X, y)
ficha = {'versao': 3, 'dados': resumo(dados.to_csv(index=False).encode()),
         'commit': '4f9c2e1', 'C': 1.0, 'modelo': resumo(pickle.dumps(modelo))}
for campo, valor in ficha.items():
    print(f'{campo}: {valor}')
```

SAÍDA DA EXECUÇÃO

```
dados, versão 1          465b0e03f1c48074
dados, uma alteração     9366d52d2ff8d24f
dados, versão 1, de novo 465b0e03f1c48074
versao: 3
dados: 465b0e03f1c48074
commit: 4f9c2e1
C: 1.0
modelo: badb31662c4cf850
```

Dez reais a mais em um único pagamento, entre mil, mudaram o hash inteiro do conjunto; refeito o cálculo sobre os dados originais, o hash volta a ser o mesmo. A ficha amarra a versão 3 do modelo ao conteúdo exato dos dados, ao commit do código e aos hiperparâmetros: é essa linhagem que permite dizer, meses depois, com que dados e com que código foi gerada uma previsão, e treinar de novo o mesmo modelo.

ESTRATÉGIA · APROFUNDE AQUI · P. 9 A 10 E 15

Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · Capítulo 8 (Ciclo de vida e MLOps)

Foco: versionamento (p. 9), DVC (p. 10) e versionamento dos dados desde a coleta (p. 15).

https://concursos.estrategia.com/ldi/ciencia-de-dados-e-inteligencia-artificial-para-cgu-auditor-federal-de-financas-e-controle-ti-ciencias-de-dados-pos-edital

SE O SEU CURSO FOR OUTRO

Gran (vídeo): disciplina "Operação, Governança e Gestão de Soluções de Dados e IA", vídeo "MLOps"

Direção (livro digital): módulo "Operação, Governança e Gestão de Soluções de Dados e IA", aula 1 (Ciclo de vida e MLOps, parte 1 de 2), capítulo "MLOps"; aula 2 (Ciclo de vida e MLOps, parte 2 de 2), capítulo "Pipeline de dados: versionamento, logging e auditoria, retries e checkpoints, Integração com CI/CD"

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

PONTOS IMPORTANTES

- O código fica no Git, os dados ficam em cópias imutáveis identificadas por hash e os modelos ficam num registro com versões e apelidos.
- O mesmo conteúdo dá sempre o mesmo hash, e qualquer alteração muda o hash.
- A linhagem liga cada versão do modelo aos dados, ao código e aos parâmetros; sem ela não há rastreabilidade nem reprodução.

COMO O CEBRASPE PODE COBRAR

- Afirmar que, em MLOps, o controle de versão se restringe ao código-fonte: ele abrange dados, modelos e configurações, para assegurar reprodutibilidade e auditabilidade.
- Dizer que o Git, sozinho, é a forma recomendada de guardar grandes conjuntos de dados, ou que o DVC substitui o Git.
- Dizer que voltar à versão anterior do modelo exige retreiná-lo.

3 Registro de experimentos

O **registro de experimentos** (experiment tracking) guarda cada tentativa de modelagem de forma sistemática, para comparar resultados e refazer qualquer uma delas. No MLflow, um **experimento** agrupa as **execuções** (runs) feitas para um mesmo objetivo, e cada execução registra os **parâmetros** (algoritmo, hiperparâmetros, semente), as **métricas** (com o esquema de validação usado), os **artefatos** (o modelo treinado, gráficos, relatórios) e o **contexto** (commit do código, hash dos dados, ambiente, data e autor). A execução escolhida vira versão no registro de modelos, sem perder a ligação com o experimento. Além do MLflow, que cobre rastreamento, reprodução, empacotamento e implantação, há o Weights and Biases, o DVC e as plataformas gerenciadas.

Três cuidados dão valor ao registro. O primeiro é registrar **todas** as execuções, inclusive as ruins: relatar só a melhor esconde a instabilidade, e a média com o desvio entre dobras ou sementes diz mais do que o melhor número. O segundo é comparar os candidatos **sobre a mesma versão dos dados e o mesmo esquema de validação**. O terceiro é escolher pela validação, usando o conjunto de teste uma única vez, no fim.

No setor público, o rastreamento de experimentos é recomendação expressa do Guia Unificado de IA, por causa da auditoria:

CITAÇÃO LITERAL

*"Desenvolver algoritmos. Aplicar algoritmos estatísticos e de ML e garantir que o desenvolvimento suporte a rastreabilidade (linhagem) das variáveis e o **rastreamento de experimentos** (Experiment Tracking), **crucial para auditorias** em contextos governamentais. (...) Rastreabilidade de experimentos. Model registry. Utilizar catalogação e **versionamento de modelos desde o início da experimentação** para garantir a reprodutibilidade dos testes."*

SGD, Guia Unificado de IA (2026), fase 3.2, recomendações, p. 45 e 46 · abrir a fonte

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

EXEMPLO COMENTADO · TRÊS EXECUÇÕES REGISTRADAS SOBRE A MESMA VERSÃO DOS DADOS

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score

treino = gera_lote(1500, semente=1)
X, y = treino.drop(columns='irregular'), treino['irregular']
dados = hashlib.sha256(treino.to_csv(index=False).encode()).hexdigest()[:8]
candidatos = [
    ('regressão logística', 'C=1.0', LogisticRegression(C=1.0, max_iter=1000)),
    ('árvore de decisão', 'max_depth=3',
     DecisionTreeClassifier(max_depth=3, random_state=0)),
    ('árvore de decisão', 'max_depth=15',
     DecisionTreeClassifier(max_depth=15, random_state=0)),
]
execucoes = []
for i, (algoritmo, params, modelo) in enumerate(candidatos, start=1):
    auc = cross_val_score(modelo, X, y, cv=5, scoring='roc_auc')   # 5 dobras
    execucoes.append({'execução': f'run-{i:02d}', 'algoritmo': algoritmo,
                      'parâmetros': params, 'dados': dados,
                      'AUC média': round(auc.mean(), 3),
                      'desvio': round(auc.std(), 3)})
print(pd.DataFrame(execucoes).to_string(index=False))
```

SAÍDA DA EXECUÇÃO

```
execução           algoritmo   parâmetros    dados  AUC média  desvio
  run-01 regressão logística        C=1.0 ad712efb      0.779   0.033
  run-02   árvore de decisão  max_depth=3 ad712efb      0.693   0.019
  run-03   árvore de decisão max_depth=15 ad712efb      0.591   0.028
```

As três execuções usaram os mesmos dados (o hash `ad712efb` aparece em todas) e a mesma validação, por isso a comparação vale. A regressão logística teve a maior AUC média (0,779); a árvore com profundidade 15 caiu para 0,591, sinal de sobreajuste. Quem precisar refazer a execução vencedora sabe o algoritmo, o parâmetro e a versão dos dados.

ESTRATÉGIA · APROFUNDE AQUI · P. 10 E 24 A 25

Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · Capítulo 8 (Ciclo de vida e MLOps)

Foco: acompanhamento de experimentos (p. 10) e MLflow (p. 24 e 25); as demais ferramentas estão nas p. 22 a 27.

https://concursos.estrategia.com/ldi/ciencia-de-dados-e-inteligencia-artificial-para-cgu-auditor-federal-de-financas-e-controle-ti-ciencias-de-dados-pos-edital

SE O SEU CURSO FOR OUTRO

Gran (vídeo): disciplina "Operação, Governança e Gestão de Soluções de Dados e IA", vídeo "MLOps"

Direção (livro digital): módulo "Operação, Governança e Gestão de Soluções de Dados e IA", aula 1 (Ciclo de vida e MLOps, parte 1 de 2), capítulo "MLOps"

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

PONTOS IMPORTANTES

- Cada execução registra os parâmetros, as métricas, os artefatos e o contexto (a versão do código, a versão dos dados e o ambiente).
- A comparação só vale com os mesmos dados e a mesma validação, e registra-se tudo, não só o melhor resultado.
- O MLflow cobre o ciclo inteiro: rastreamento, reprodução, empacotamento, registro e implantação de modelos.

COMO O CEBRASPE PODE COBRAR

- Afirmar que o MLflow é ferramenta exclusiva para uma única tarefa, como o treinamento ou a implantação.
- Dizer que basta registrar a execução de melhor desempenho.
- Considerar válida a comparação entre execuções feitas sobre versões diferentes dos dados, desde que a métrica seja a mesma.

4 Pipelines, testes e integração e entrega contínuas

O **pipeline de ML** é a sequência automatizada que leva dos dados ao modelo publicado: extração, validação dos dados, preparação, treinamento, avaliação, validação do modelo, registro e implantação. Cada etapa recebe e produz artefatos versionados, e um orquestrador (Airflow, Kubeflow Pipelines, TFX) controla a ordem, as repetições e as falhas. O pipeline é acionado por agenda, por dados novos, por queda de desempenho ou por mudança no código. Os testes vão além do software, porque o comportamento depende dos dados:

Alvo do teste	O que verifica
Código	Testes de unidade das funções que criam atributos e de integração entre as etapas; análise estática na construção
Dados	Esquema, tipos, nulos, faixas e domínios, unicidade, distribuição parecida com a do treino, conformidade com a LGPD
Modelo	Desempenho mínimo; comparação com o modelo em produção e com uma linha de base simples; desempenho por grupos; invariância e direção esperada
Infraestrutura	Reprodutibilidade do treino, carga, consistência entre treino e produção, liberação para poucos usuários (canário)

O teste de unidade não substitui os demais: ele não dispensa os testes de integração e de aceitação nem a revisão de código.

Prática	O que automatiza	Até onde vai
Integração contínua (CI)	Mudanças pequenas e frequentes (várias vezes ao dia) são integradas a um repositório central, e cada integração dispara construção, análise estática e testes; em ML, também a validação de dados e de modelos	Até um artefato testado; não leva nada à produção

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

Entrega contínua (CD)	O artefato aprovado passa pelos ambientes de teste e homologação e fica pronto para produção	A ida à produção depende de uma aprovação
Implantação contínua	Toda mudança aprovada nos testes vai à produção sozinha	Produção, sem intervenção humana

O treinamento contínuo (CT) automatiza o retreino com dados novos, e o conjunto de testes funciona como portão de qualidade: o que não passa não segue.

A infraestrutura também se descreve como código (**IaC**), versionado e testado: na forma **declarativa** (Terraform, manifestos do Kubernetes) descreve-se o estado final desejado e a ferramenta calcula o que mudar; na **imperativa**, escrevem-se os comandos na ordem de execução. Os contêineres (Docker) empacotam o modelo com as dependências, o registro de contêineres guarda as imagens versionadas e o Kubernetes orquestra os contêineres em produção.

No governo federal, o modelo de contratação de operação de infraestrutura da Secretaria de Governo Digital reúne essas práticas sob o nome DevSecOps, com a segurança integrada ao desenvolvimento e à operação:

CITAÇÃO LITERAL

*"4.1.7. Um ambiente **DevSecOps** possui um ou mais recursos que asseguram condições para se alcançar um elevado grau de automação da infraestrutura. Em geral, são empregadas ferramentas de: a) Controle de versão; b) **Integração contínua**; c) **Testes contínuos**; d) Gerenciamento de configuração e deployment; e) Monitoramento contínuo; f) Conteinerização; g) Orquestração; h) Segurança integrada; e i) Gerenciamento integrado de demandas."*

Portaria SGD/MGI 1.070/2023, Anexo I, item 4.1.7 · abrir a fonte

EXEMPLO COMENTADO · O PORTÃO DE QUALIDADE DE UM PIPELINE, COM UNITTEST

```
import io, unittest

class TestaLote(unittest.TestCase):
    lote = None                    # o lote novo, posto antes de rodar
    def test_esquema(self):
        colunas = ['valor', 'aditivos', 'fornecedor_novo', 'irregular']
        self.assertEqual(list(self.lote.columns), colunas)
    def test_sem_nulos(self):
        self.assertFalse(self.lote.isna().any().any())
    def test_valor_positivo(self):
        self.assertTrue((self.lote['valor'] > 0).all())

def portao(lote, nome):
    TestaLote.lote = lote
    r = unittest.TextTestRunner(stream=io.StringIO()).run(
        unittest.TestLoader().loadTestsFromTestCase(TestaLote))
    falhas = [t.id().split('.')[-1] for t, _ in r.failures + r.errors]
    print(f'{nome}: {r.testsRun} testes, falhas: {falhas or "nenhuma"}')
    print('   ->', 'pipeline interrompido' if falhas else 'treinamento liberado')
```


AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

```
bom, ruim = gera_lote(500, semente=5), gera_lote(500, semente=6)
ruim.loc[10, 'valor'] = -350.0              # estorno lançado como pagamento
ruim.loc[20, 'aditivos'] = None             # campo vazio na extração
portao(bom, 'lote de março')
portao(ruim, 'lote de abril')
```

SAÍDA DA EXECUÇÃO

```
lote de março: 3 testes, falhas: nenhuma
   -> treinamento liberado
lote de abril: 3 testes, falhas: ['test_sem_nulos', 'test_valor_positivo']
   -> pipeline interrompido
```

O lote de março passou nos três testes e o treinamento seguiu. No de abril, um valor negativo e um campo vazio reprovaram dois testes, e o pipeline parou antes de treinar: sem o portão, o modelo seria retreinado com um estorno tratado como pagamento. Num pipeline real, o portão inclui também testes do modelo, como desempenho pelo menos igual ao da versão em produção.

ESTRATÉGIA · APROFUNDE AQUI · P. 8 A 13

Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · Capítulo 8 (Ciclo de vida e MLOps)

Foco: níveis de automação e práticas contínuas (p. 8 e 9) e as tabelas de testes de dados, de modelo e de infraestrutura (p. 10 a 13).

https://concursos.estrategia.com/ldi/ciencia-de-dados-e-inteligencia-artificial-para-cgu-auditor-federal-de-financas-e-controle-ti-ciencias-de-dados-pos-edital

SE O SEU CURSO FOR OUTRO

Gran (vídeo): disciplina "Operação, Governança e Gestão de Soluções de Dados e IA", vídeos "Pipeline de Machine Learning" (I e II)

Direção (livro digital): módulo "Operação, Governança e Gestão de Soluções de Dados e IA", aula 1 (Ciclo de vida e MLOps, parte 1 de 2), capítulo "MLOps"; aula 2 (Ciclo de vida e MLOps, parte 2 de 2), capítulo "Pipeline de dados: versionamento, logging e auditoria, retries e checkpoints, Integração com CI/CD"

PONTOS IMPORTANTES

- Em ML, a CI testa o código, os dados, os esquemas e os modelos, e o teste que falha interrompe o pipeline.
- A CI integra e testa as mudanças; na entrega contínua, a ida à produção depende de aprovação, e na implantação contínua, não.
- A IaC declarativa descreve o estado desejado, e a imperativa descreve os passos para chegar a ele.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

COMO O CEBRASPE PODE COBRAR

- Definir a integração contínua como a liberação automática das mudanças do repositório até a produção, o que é implantação contínua.
- Dizer que, numa boa prática de CI, as mudanças são acumuladas e integradas semanalmente; o recomendado são integrações pequenas e frequentes.
- Atribuir à IaC declarativa a descrição dos comandos, na ordem, para chegar ao estado desejado.

5 Reprodutibilidade

Um resultado é **reprodutível** quando, repetidos os mesmos dados, o mesmo código, a mesma configuração e o mesmo ambiente, se obtém o mesmo resultado. Sem isso não é possível auditar um modelo, investigar uma previsão contestada nem saber se uma melhora veio da mudança feita ou do acaso. Distinguem-se a reprodutibilidade do **método** (a descrição completa permite refazer o processo) e a dos **resultados** (o resultado se sustenta quando o experimento se repete).

Cada fonte de variação tem o seu controle. A aleatoriedade na divisão dos dados, na amostragem e na inicialização se controla fixando e registrando a semente; os dados que mudam na origem, como a consulta "até hoje", com o treino sobre uma cópia imutável, identificada por hash; as versões de bibliotecas e ferramentas, com as versões exatas no arquivo de dependências, o ambiente isolado e a imagem de contêiner identificada pelo digest; o paralelismo e a GPU, em que a ordem das somas varia, com as opções de determinismo do framework; e os passos manuais em notebooks executados fora de ordem, com o pipeline automatizado e a configuração como código.

O Guia Unificado de IA liga a reprodutibilidade ao ambiente empacotado e à infraestrutura descrita como código:

CITAÇÃO LITERAL

*"O uso de **imagens de contêineres imutáveis** permite que todas as dependências sejam configuradas, reduzindo riscos de inconsistências. (...) O provisionamento de infraestrutura deve ser realizado via ferramentas de Infraestrutura como Código (IaC), **garantindo reprodutibilidade** e trilha de auditoria das configurações."*

SGD, Guia Unificado de IA (2026), recomendações das fases 3.3 e 4.1, p. 48 e 50 · abrir a fonte

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

EXEMPLO COMENTADO · A MESMA RECEITA COM E SEM A MESMA SEMENTE

```
import sys, sklearn
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

dados = gera_lote(2000, semente=3)
X, y = dados.drop(columns='irregular'), dados['irregular']

def treina(semente):
    Xtr, Xte, ytr, yte = train_test_split(X, y, test_size=0.3,
                                          random_state=semente)
    rf = RandomForestClassifier(n_estimators=50,
                                random_state=semente).fit(Xtr, ytr)
    return (f'acurácia {rf.score(Xte, yte):.4f}, '
            f'importâncias {rf.feature_importances_.round(3).tolist()}')

print('semente 1          ', treina(1))
print('semente 2          ', treina(2))
print('semente 1, de novo ', treina(1))
print('ambiente: Python', sys.version.split()[0], '| numpy', np.__version__,
      '| pandas', pd.__version__, '| scikit-learn', sklearn.__version__)
```

SAÍDA DA EXECUÇÃO

```
semente 1           acurácia 0.7300, importâncias [0.839, 0.132, 0.029]
semente 2           acurácia 0.7100, importâncias [0.821, 0.157, 0.022]
semente 1, de novo  acurácia 0.7300, importâncias [0.839, 0.132, 0.029]
ambiente: Python 3.14.7 | numpy 2.5.4 | pandas 3.0.6 | scikit-learn 1.9.1
```

Sem fixar a semente, cada execução usa uma diferente; para o exemplo sair sempre igual, as duas usam as sementes 1 e 2. A mesma receita, sobre os mesmos dados, deu acurácia de 0,73 com uma e de 0,71 com a outra, e importâncias diferentes. Repetida a semente 1, tudo volta igual. A última linha registra o ambiente: com outra versão de biblioteca, o mesmo código pode dar outro número. Reprodutível não quer dizer correto, e a diferença entre sementes mede a estabilidade do modelo; por isso a avaliação séria repete o treino e relata a média e o desvio.

ESTRATÉGIA · APROFUNDE AQUI · P. 13, 19 E 25 A 26

Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · Capítulo 8 (Ciclo de vida e MLOps)

Foco: teste de reprodutibilidade (p. 13), reprodutibilidade do método e dos resultados (p. 19) e contêineres no MLflow e no TFX (p. 25 e 26).

https://concursos.estrategia.com/ldi/ciencia-de-dados-e-inteligencia-artificial-para-cgu-auditor-federal-de-financas-e-controle-ti-ciencias-de-dados-pos-edital

SE O SEU CURSO FOR OUTRO

Gran (vídeo): disciplina "Operação, Governança e Gestão de Soluções de Dados e IA", vídeo "MLOps"

Direção (livro digital): módulo "Operação, Governança e Gestão de Soluções de Dados e IA", aula 1 (Ciclo de vida e MLOps, parte 1 de 2), capítulo "MLOps"

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

PONTOS IMPORTANTES

- A reprodutibilidade exige os mesmos dados, o mesmo código, a mesma configuração, as mesmas sementes e o mesmo ambiente.
- Fixa-se a semente, registra-se o ambiente e leem-se os dados de uma cópia imutável, e não da origem viva.
- A variação entre sementes mede a estabilidade do modelo, e por isso se relatam a média e o desvio.

COMO O CEBRASPE PODE COBRAR

- Afirmar que fixar a semente aleatória basta para reproduzir o treino, mesmo com os dados relidos da origem a cada execução.
- Dizer que uma ferramenta de infraestrutura como código trata sozinha a compatibilidade de versões, dispensando fixá-las.
- Tratar a reprodutibilidade como garantia de que o resultado está correto.

6 Disponibilização por APIs, implantação e reversão

O modelo validado chega a quem usa a previsão de quatro formas: **em lote** (inferência estática: um processo agendado calcula os escores e grava numa tabela, sem exigência de latência), **online por API** (inferência dinâmica: cada pedido é respondido na hora, via REST ou gRPC, com baixa latência), **em fluxo** (o modelo consome eventos de uma fila) e **embarcada** (o modelo roda no próprio dispositivo).

Na API, o ponto central é o **contrato** (endereço, método, esquema da entrada e da saída, códigos de resposta, versão), descrito de preferência em OpenAPI. O serviço recusa o pedido fora do contrato (400 ou 422), exige autenticação e devolve, com a previsão, a **versão do modelo** e um identificador da predição. A versão da API (`/v1/`) independe da versão do modelo, e o mesmo artefato, em contêiner, é promovido de ambiente em ambiente sem ser reconstruído.

EXEMPLO COMENTADO · O CONTRATO DA API E A RESPOSTA COM A VERSÃO DO MODELO

```
MODELO = {'nome': 'triagem-pagamentos', 'versao': 7, 'b0': -3.5,
          'pesos': {'valor': 0.01, 'aditivos': 0.8, 'fornecedor_novo': 0.9}}
CONTRATO = {'valor': (int, float), 'aditivos': int, 'fornecedor_novo': int}

def prever(corpo):                  # simula o endpoint POST /v1/triagem
    pedido = json.loads(corpo)
    erros = [c for c, tipo in CONTRATO.items()
             if not isinstance(pedido.get(c), tipo)]
    if erros:
        return 422, {'erro': 'pedido fora do contrato', 'campos': erros}
    z = MODELO['b0'] + sum(MODELO['pesos'][c] * pedido[c] for c in CONTRATO)
    rastro = f"{corpo}|{MODELO['versao']}".encode()
    rastro = hashlib.sha256(rastro).hexdigest()[:12]
    return 200, {'escore': round(1 / (1 + np.exp(-z)), 3),
                 'modelo': MODELO['nome'], 'versao': MODELO['versao'],
                 'id_predicao': rastro}
```


AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

```
for corpo in ['{"valor": 180.0, "aditivos": 3, "fornecedor_novo": 1}',
              '{"valor": "180,00", "aditivos": 3}']:
    status, resposta = prever(corpo)
    print(status, json.dumps(resposta, ensure_ascii=False, indent=1))
```

SAÍDA DA EXECUÇÃO

```
200 {
 "escore": 0.832,
 "modelo": "triagem-pagamentos",
 "versao": 7,
 "id_predicao": "afb5906f5707"
}
422 {
 "erro": "pedido fora do contrato",
 "campos": [
  "valor",
  "fornecedor_novo"
 ]
}
```

O primeiro pedido respeita o contrato e recebe o escore 0,832, com o nome e a versão do modelo e um identificador da predição. O segundo traz o valor como texto e não traz o indicador de fornecedor novo: o serviço responde 422 e diz o que está errado, em vez de devolver um escore calculado sobre entrada inválida.

Estratégia de implantação	Como funciona	Como se reverte
Recriação	Desliga a versão antiga e sobe a nova; há interrupção	Reimplantando a antiga
Atualização gradual (rolling)	Troca as instâncias aos poucos; por um tempo as duas versões atendem	Interrompendo a troca e voltando as já trocadas
Azul-verde (blue-green)	Dois ambientes completos; o novo é testado e depois recebe todo o tráfego	Devolvendo o tráfego ao ambiente antigo, que continua pronto
Canário	Uma fração pequena do tráfego vai para a versão nova, comparada à atual antes de ampliar	Retirando a fração; poucos usuários foram expostos
Sombra (shadow)	A versão nova recebe cópia do tráfego real, mas a resposta dela não é usada	Nada a reverter: a nova não atendeu ninguém
Teste A/B	Grupos de usuários recebem versões diferentes para medir o efeito no resultado	Encerrando o teste e mantendo a vencedora

O canário protege a operação (a versão nova quebra algo?); o A/B mede o efeito (qual versão produz melhor resultado?).

A **reversão** (rollback) é a volta rápida à última versão estável e precisa ser preparada antes, com a versão anterior preservada no registro, o apelido de produção (reverter é mudar o apelido), o contrato e o esquema de atributos compatíveis entre as versões e os critérios de reversão definidos antes da liberação, de preferência acionados automaticamente. Ela não se confunde com a **contingência**

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

(fallback), a alternativa usada quando o modelo falha ou fica indisponível, como uma regra simples, a operação manual ou travas que limitam as saídas. O Guia Unificado de IA da Secretaria de Governo Digital pede as duas coisas: o plano de reversão testado antes da entrada em produção, com implantação segura, e o plano de contingência na sustentação:

CITAÇÃO LITERAL

*"**Plano de rollback construído e testado**. (...) Estratégias de Safe Deployment como **Shadow Mode (execução em paralelo) ou Canary Release (liberação gradual)**. (...) Documentação de regras de negócio determinísticas ou sistemas legados que **assumem o serviço caso o modelo de IA falhe** ou se degrade criticamente."*

SGD, Guia Unificado de IA (2026), fase 4.3 (plano de rollback e implantação segura), p. 52 e 53, e fase 6.2 (plano de contingência, failover), p. 56 · abrir a fonte

A implantação é uma mudança, e a norma do GSI pede processo para isso:

CITAÇÃO LITERAL

"O processo mencionado no caput, além de promover o controle das mudanças planejadas, deve considerar a análise crítica das consequências de mudanças não previstas, atuando em ações para amenizar os efeitos adversos."

IN GSI/PR 3/2021, art. 28, § 2º · abrir a fonte

SÓ NESTE MATERIAL

O capítulo do Estratégia trata do canário, da implantação gradual, do teste A/B e da contingência, mas não do contrato da API nem das estratégias azul-verde e sombra. Para esses pontos, este material é a fonte de estudo.

ESTRATÉGIA · APROFUNDE AQUI · P. 13 E 20 A 22

Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · Capítulo 8 (Ciclo de vida e MLOps)

Foco: canary testing (p. 13), implantação no CRISP-ML(Q), com contingência e implantação gradual (p. 20 e 21), e A/B e volta à versão anterior (p. 22).

https://concursos.estrategia.com/ldi/ciencia-de-dados-e-inteligencia-artificial-para-cgu-auditor-federal-de-financas-e-controle-ti-ciencias-de-dados-pos-edital

SE O SEU CURSO FOR OUTRO

Gran (vídeo): disciplina "Operação, Governança e Gestão de Soluções de Dados e IA", vídeos "Pipeline de Machine Learning" (I e II)

Direção (livro digital): módulo "Operação, Governança e Gestão de Soluções de Dados e IA", aula 1 (Ciclo de vida e MLOps, parte 1 de 2), capítulo "Deploy de modelos"

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

PONTOS IMPORTANTES

- O lote serve quando o escore pode ser calculado antes, e a API, quando a previsão é pedida na hora.
- A API valida a entrada pelo contrato e devolve a versão do modelo com a previsão.
- O azul-verde reverte devolvendo o tráfego ao ambiente antigo, o canário expõe poucos usuários e a sombra não expõe ninguém.

COMO O CEBRASPE PODE COBRAR

- Dizer que, no lançamento canário, a nova versão vai a todos os usuários de uma vez.
- Afirmar que, na implantação em sombra, as respostas da versão nova são entregues aos usuários.
- Confundir o teste A/B, que mede o efeito de versões concorrentes, com o canário, que protege a operação.

7 Monitoramento de mudanças nas distribuições, nas relações e no desempenho

O modelo aprende a relação entre atributos e alvo que existia nos dados de treino. Em produção, três coisas podem mudar, e cada uma pede vigilância diferente:

O que muda	Nome usual	Como se percebe
A distribuição das entradas, P(x)	Mudança de distribuição (data drift, covariate shift)	Comparando cada atributo em produção com o treino (KS, qui-quadrado, PSI); não precisa de rótulo
A proporção das classes, P(y)	Mudança de prevalência (label shift)	Pela taxa de positivos e pela distribuição dos escores
A relação entre entradas e alvo, P(y|x)	Mudança de conceito (concept drift)	Pela queda do desempenho e pela mudança das correlações; exige rótulos

Monitoram-se também as saídas (distribuição dos escores), a qualidade dos dados, a distorção entre treino e serviço (o mesmo atributo calculado de forma diferente em produção) e a operação (latência e erros, no tópico 2).

O Guia Unificado de IA da Secretaria de Governo Digital pede, na sustentação, alertas automáticos para a mudança nas entradas e para a mudança na relação:

CITAÇÃO LITERAL

*"Notificações automatizadas disparadas quando há **mudança na distribuição dos dados de entrada** (Data Drift) ou **na relação entre variáveis** (Concept Drift)."*

SGD, Guia Unificado de IA (2026), fase 6.1, resultado Alertas de drift e anomalias, p. 55 · abrir a fonte

O teste de **Kolmogorov-Smirnov** (KS) de duas amostras mede a maior distância entre as funções de distribuição acumulada e testa se as amostras vêm da mesma distribuição; serve a atributos contínuos. O **qui-quadrado** faz o mesmo para atributos categóricos. O **índice de estabilidade populacional** (PSI)

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

divide a referência em faixas e soma, em cada faixa, a diferença entre a proporção atual e a de referência multiplicada pelo logaritmo da razão entre elas. Pela convenção de mercado, herdada da modelagem de crédito, o PSI abaixo de 0,1 indica população estável; entre 0,1 e 0,25, mudança moderada; e acima de 0,25, mudança significativa. O teste diz se a diferença é detectável; o PSI diz se é grande.

EXEMPLO COMENTADO · O VALOR DOS PAGAMENTOS EM TRÊS MESES DE PRODUÇÃO: KS E PSI

```
from scipy.stats import ks_2samp

def psi(ref, atual, faixas=10):         # faixas pelos decis da referência
    cortes = np.quantile(ref, np.linspace(0, 1, faixas + 1))
    cortes[0], cortes[-1] = -np.inf, np.inf
    p, q = (np.clip(np.histogram(x, cortes)[0] / len(x), 1e-6, None)
            for x in (ref, atual))
    return float(np.sum((q - p) * np.log(q / p)))

ref = gera_lote(5000, semente=31)['valor']           # dados de treino
print(f'treinamento: mediana {ref.median():.1f}')
for mes, semente, escala in [('maio', 131, 1.0), ('junho', 132, 1.1),
                             ('julho', 133, 1.6)]:
    atual = gera_lote(5000, semente=semente, escala_valor=escala)['valor']
    ks = ks_2samp(ref, atual)
    print(f'{mes:6} mediana {atual.median():6.1f} | KS {ks.statistic:.3f}, '
          f'p-valor {ks.pvalue:.2g} | PSI {psi(ref, atual):.3f}')
```

SAÍDA DA EXECUÇÃO

```
treinamento: mediana 55.1
maio   mediana   54.7 | KS 0.010, p-valor 0.98 | PSI 0.003
junho  mediana   61.1 | KS 0.062, p-valor 1e-08 | PSI 0.021
julho  mediana   87.3 | KS 0.228, p-valor 2.7e-114 | PSI 0.312
```

Maio saiu da mesma distribuição do treino: p-valor de 0,98 e PSI quase nulo. Em junho os valores subiram 10%: com 5 mil pagamentos de cada lado, o KS já rejeita a igualdade (p-valor da ordem de 10^{-8}), mas o PSI de 0,021 está bem abaixo de 0,1: a diferença existe e é pequena. Em julho, a alta de 60% levou o PSI a 0,312, acima de 0,25, mudança significativa que justifica investigar a causa e avaliar retreino. Com amostras grandes, quase toda diferença fica significativa, por isso o alerta costuma usar a magnitude.

A mudança na **relação** é a mais traiçoeira, porque as entradas podem continuar idênticas: se uma nova regra de contratação faz os aditivos deixarem de indicar problema, ou se os fornecedores aprendem a evitar o sinal, o modelo continua aplicando a relação antiga. Só o **desempenho**, medido quando os rótulos chegam, e a mudança das correlações revelam o problema. Em auditoria os rótulos chegam com meses de atraso, e por isso se combinam indicadores imediatos (entradas e escores) com indicadores atrasados (desempenho).

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

EXEMPLO COMENTADO · MUDANÇA NA RELAÇÃO: AS ENTRADAS IGUAIS E O DESEMPENHO EM QUEDA

```
from scipy.stats import ks_2samp
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score

treino = gera_lote(5000, semente=20)
X, y = treino.drop(columns='irregular'), treino['irregular']
modelo = LogisticRegression(max_iter=1000).fit(X, y)
linhas = []
meses = [('ago', 0.8), ('set', 0.8), ('out', 0.4), ('nov', 0.1), ('dez', -0.3)]
for i, (mes, peso) in enumerate(meses):
    lote = gera_lote(5000, semente=200 + i, peso_aditivos=peso)
    Xm, ym = lote.drop(columns='irregular'), lote['irregular']
    linhas.append([mes, ks_2samp(X['aditivos'], Xm['aditivos']).pvalue,
                   np.corrcoef(Xm['aditivos'], ym)[0, 1], ym.mean(),
                   roc_auc_score(ym, modelo.predict_proba(Xm)[:, 1])])
colunas = ['mês', 'KS aditivos (p)', 'corr(aditivos, irregular)',
           'taxa irregular', 'AUC']
print(pd.DataFrame(linhas, columns=colunas).round(3).to_string(index=False))
```

SAÍDA DA EXECUÇÃO

```
mês  KS aditivos (p)  corr(aditivos, irregular)  taxa irregular   AUC
ago            1.000                      0.323           0.227 0.784
set            0.980                      0.333           0.233 0.790
out            1.000                      0.156           0.155 0.728
nov            1.000                      0.024           0.107 0.659
dez            0.744                     -0.070           0.073 0.633
```

A distribuição dos aditivos não mudou (p-valor do KS sempre alto), mas a correlação entre aditivos e irregularidade caiu de 0,323 em agosto para 0,024 em novembro e ficou negativa em dezembro. O modelo, que continua tratando aditivos como sinal de risco, viu a AUC sair de 0,784 em agosto para 0,633 em dezembro. Um painel que olhasse só as entradas não teria visto nada.

Com o diagnóstico, decide-se investigar a causa, retreinar (por agenda, por queda abaixo de um limiar ou por alerta), ajustar o limiar ou redesenhar o modelo. Há duas armadilhas: o **ciclo de retroalimentação** (o modelo escolhe quem é auditado, só os escolhidos ganham rótulo e o retreino aprende com amostra enviesada por ele mesmo) e a **média que esconde grupos** (o desempenho geral se mantém enquanto cai num segmento).

Na CGU, a Portaria Normativa CGU 179/2024 criou a Comissão de Governança de Ciência de Dados (CGCD), que aprova as soluções antes de entrarem em produção e acompanha as revisões das que já estão em uso (art. 4º, VII e VIII). A norma define os dois conceitos:

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

CITAÇÃO LITERAL

*"VII - **revisão** - ajuste da solução de Ciência de Dados devido à **mudança do contexto** em que ela é utilizada em relação ao momento em que ela foi concebida; e VIII - **supervisão técnica** - avaliação realizada pela CGCD sobre as soluções de Análise de Dados e de Inteligência Artificial que estejam ou devam entrar em produção na Controladoria-Geral da União, de modo a garantir que essas soluções atendam às diretrizes para projetos de Ciência de Dados do órgão."*

Portaria Normativa CGU 179/2024, art. 1º, § 2º, VII e VIII, p. 1 do PDF (DOU) · abrir a fonte

SÓ NESTE MATERIAL

O capítulo do Estratégia trata do que monitorar e da degradação do modelo, mas não dos testes de Kolmogorov-Smirnov e qui-quadrado nem do PSI. Para esses instrumentos, este material é a fonte de estudo.

ESTRATÉGIA · APROFUNDE AQUI · P. 13 A 14 E 21 A 22

Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · Capítulo 8 (Ciclo de vida e MLOps)

Foco: o que monitorar em produção (p. 13 e 14) e monitoramento e manutenção no CRISP-ML(Q), com ciclos de retroalimentação (p. 21 e 22).

https://concursos.estrategia.com/ldi/ciencia-de-dados-e-inteligencia-artificial-para-cgu-auditor-federal-de-financas-e-controle-ti-ciencias-de-dados-pos-edital

SE O SEU CURSO FOR OUTRO

Gran (vídeo): disciplina "Operação, Governança e Gestão de Soluções de Dados e IA", vídeo "Monitoramento de Modelos"

Direção (livro digital): módulo "Operação, Governança e Gestão de Soluções de Dados e IA", aula 1 (Ciclo de vida e MLOps, parte 1 de 2), capítulo "MLOps"

PONTOS IMPORTANTES

- A mudança de distribuição altera P(x) e se detecta sem rótulos; a mudança de conceito altera P(y|x) e aparece no desempenho.
- O KS serve aos atributos contínuos e o qui-quadrado, aos categóricos; o PSI mede a magnitude da mudança (abaixo de 0,1, estável; acima de 0,25, significativa).
- Os rótulos atrasados e o ciclo de retroalimentação exigem indicadores imediatos e atrasados e o monitoramento por segmento.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

COMO O CEBRASPE PODE COBRAR

- Afirmar que a mudança de conceito é detectada comparando apenas a distribuição dos atributos de entrada.
- Dizer que o PSI acima de 0,25 indica população estável.
- Tratar um p-valor muito pequeno no teste de KS, com amostras grandes, como prova de mudança relevante.

? Questões abertas de fixação

Responda sem consultar. As respostas estão na página seguinte.

1. De que três elementos depende um sistema de ML em produção, e por que ele pode perder desempenho sem mudança de código?

2. Como se versionam código, dados e modelos, e o que é linhagem?

3. O que se registra em cada execução de um experimento?

4. Diferencie integração contínua, entrega contínua e implantação contínua.

5. Que testes, além dos de código, um pipeline de ML deve ter?

6. Cite três fontes de falta de reprodutibilidade e o controle de cada uma.

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

7. Compare azul-verde, canário e sombra quanto à exposição dos usuários e à reversão.

8. Diferencie mudança de distribuição e mudança de conceito e diga como se detecta cada uma.

9. O que o PSI mede e como se interpreta?

10. O que é o ciclo de retroalimentação num modelo de seleção para auditoria?

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

✓ Respostas das questões abertas

1.

De que três elementos depende um sistema de ML em produção, e por que ele pode perder desempenho sem mudança de código?

Depende dos dados, do código e do modelo (os parâmetros aprendidos). Ele perde desempenho porque a realidade se afasta dos dados de treino: mudam a distribuição das entradas, a proporção das classes ou a própria relação entre os atributos e o alvo.

2.

Como se versionam código, dados e modelos, e o que é linhagem?

O código fica no Git; os dados, em cópias imutáveis identificadas por hash (DVC, versões de tabela de lakehouse); os modelos, num registro com versões numeradas e apelidos. A linhagem é a ligação de cada versão do modelo à execução, aos dados, ao código e aos parâmetros que a geraram.

3.

O que se registra em cada execução de um experimento?

Registram-se os parâmetros (algoritmo, hiperparâmetros, semente), as métricas com o esquema de validação, os artefatos (modelo, gráficos, relatórios) e o contexto (commit do código, hash dos dados, ambiente, data e autor).

4.

Diferencie integração contínua, entrega contínua e implantação contínua.

Na CI, as mudanças pequenas e frequentes são integradas a um repositório central, e cada integração dispara a construção e os testes, sem ir à produção. Na entrega contínua, o artefato aprovado fica pronto para produção, e a ida depende de aprovação. Na implantação contínua, toda mudança aprovada vai à produção sem intervenção humana.

5.

Que testes, além dos de código, um pipeline de ML deve ter?

Ele deve ter testes de dados (esquema, tipos, nulos, faixas, distribuição), de modelo (desempenho mínimo, comparação com o modelo em produção e com a linha de base, desempenho por grupos, invariância) e de infraestrutura (reprodutibilidade, carga, consistência entre treino e produção).

6.

Cite três fontes de falta de reprodutibilidade e o controle de cada uma.

A aleatoriedade se controla fixando e registrando a semente; os dados lidos da origem viva, com uma cópia imutável identificada por hash; as versões de bibliotecas, com as dependências em versão exata, o ambiente isolado e o contêiner. Também contam o paralelismo em GPU e os passos manuais em notebooks.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

7.

Compare azul-verde, canário e sombra quanto à exposição dos usuários e à reversão.

No azul-verde, o ambiente novo recebe todo o tráfego e a reversão devolve o tráfego ao antigo; no canário, só uma fração pequena recebe a nova versão e a reversão retira essa fração; na sombra, a nova versão processa cópia do tráfego sem responder a ninguém.

8.

Diferencie mudança de distribuição e mudança de conceito e diga como se detecta cada uma.

A mudança de distribuição altera as entradas (P(x)) e se detecta comparando distribuições, sem rótulos (KS, qui-quadrado, PSI). A mudança de conceito altera a relação entre as entradas e o alvo (P(y|x)) e só aparece no desempenho e nas correlações, o que exige rótulos.

9.

O que o PSI mede e como se interpreta?

O PSI mede a magnitude da diferença entre a distribuição atual e a de referência, somando por faixa a diferença das proporções multiplicada pelo logaritmo da razão. Por convenção, abaixo de 0,1 a população é estável; de 0,1 a 0,25, a mudança é moderada; e acima de 0,25, significativa.

10.

O que é o ciclo de retroalimentação num modelo de seleção para auditoria?

O modelo escolhe quem é auditado, e só os escolhidos ganham rótulo; o retreino aprende com uma amostra enviesada pelo próprio modelo e tende a reforçar as escolhas anteriores.

§ Questões de certo ou errado

Cada item cobra um ponto do tópico, do jeito que o Cebraspe cobra. Marque C ou E. O gabarito comentado está na página seguinte.

1. Em MLOps, o controle de versão alcança dados, modelos e código, o que contribui para a reprodutibilidade e para a auditabilidade dos treinamentos. ( )

2. Em sistemas de aprendizado de máquina, a integração contínua pode incluir a validação de dados, de esquemas e de modelos, além dos testes de código. ( )

3. A integração contínua corresponde à liberação automática das alterações feitas pelos desenvolvedores, do repositório até o ambiente de produção. ( )

4. Na infraestrutura como código declarativa, o desenvolvedor descreve a sequência de comandos que leva a infraestrutura ao estado desejado. ( )

5. Dois arquivos de dados com o mesmo resumo SHA-256 podem ser considerados idênticos, e qualquer alteração no conteúdo gera resumo diferente. ( )

6. O registro de experimentos deve guardar apenas a execução de melhor desempenho, para não confundir a escolha do modelo. ( )

7. Fixar a semente aleatória garante a reprodutibilidade do treinamento, ainda que os dados sejam consultados na origem a cada execução. ( )

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

8. Na implantação em sombra, a nova versão do modelo recebe cópia do tráfego real, mas suas respostas não são entregues aos usuários. ( )

9. No lançamento canário, a nova versão é entregue simultaneamente a todos os usuários, e a anterior fica só como cópia de segurança. ( )

10. A mudança de conceito pode ocorrer sem que a distribuição dos atributos de entrada se altere. ( )

11. No monitoramento de modelos, um índice de estabilidade populacional superior a 0,25 indica, por convenção, população estável. ( )

12. Com amostras muito grandes, o teste de Kolmogorov-Smirnov tende a apontar diferenças estatisticamente significativas mesmo quando a mudança na distribuição é pequena. ( )

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

✓ Gabarito comentado

1 C

Em MLOps, o controle de versão alcança dados, modelos e código, o que contribui para a reprodutibilidade e para a auditabilidade dos treinamentos.

Só o código versionado não permite refazer o treino, porque os dados e o modelo também mudam.

2 C

Em sistemas de aprendizado de máquina, a integração contínua pode incluir a validação de dados, de esquemas e de modelos, além dos testes de código.

É uma das diferenças em relação ao DevOps de software tradicional.

3 E

A integração contínua corresponde à liberação automática das alterações feitas pelos desenvolvedores, do repositório até o ambiente de produção.

Essa é a implantação contínua; a CI integra e testa, sem levar à produção.

4 E

Na infraestrutura como código declarativa, o desenvolvedor descreve a sequência de comandos que leva a infraestrutura ao estado desejado.

Na declarativa, descreve-se o estado final; os comandos em ordem são a abordagem imperativa.

5 C

Dois arquivos de dados com o mesmo resumo SHA-256 podem ser considerados idênticos, e qualquer alteração no conteúdo gera resumo diferente.

É o que permite usar o hash como identificador da versão dos dados.

6 E

O registro de experimentos deve guardar apenas a execução de melhor desempenho, para não confundir a escolha do modelo.

Registram-se todas as execuções; relatar só a melhor esconde a instabilidade do modelo.

7 E

Fixar a semente aleatória garante a reprodutibilidade do treinamento, ainda que os dados sejam consultados na origem a cada execução.

Se os dados mudam na origem, o resultado muda; é preciso treinar sobre cópia imutável.

8 C

Na implantação em sombra, a nova versão do modelo recebe cópia do tráfego real, mas suas respostas não são entregues aos usuários.

Por isso a sombra não expõe nenhum usuário à versão nova.

9 E

No lançamento canário, a nova versão é entregue simultaneamente a todos os usuários, e a anterior fica só como cópia de segurança.

O canário expõe apenas uma fração pequena do tráfego antes de ampliar.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

10 C

A mudança de conceito pode ocorrer sem que a distribuição dos atributos de entrada se altere.

Muda a relação entre as entradas e o alvo, e só o desempenho e as correlações revelam essa mudança.

11 E

No monitoramento de modelos, um índice de estabilidade populacional superior a 0,25 indica, por convenção, população estável.

O PSI acima de 0,25 indica mudança significativa; abaixo de 0,1, estabilidade.

12 C

Com amostras muito grandes, o teste de Kolmogorov-Smirnov tende a apontar diferenças estatisticamente significativas mesmo quando a mudança na distribuição é pequena.

Por isso se combina o teste com uma medida de magnitude, como o PSI.

≡ Documentação oficial, normas e fontes

Documentação oficial e normas

- Google Cloud, MLOps: pipelines de entrega contínua e automação no aprendizado de máquina (pt-BR): os níveis 0, 1 e 2 de maturidade, a validação de dados e de modelos e o treinamento contínuo.
- Google, Regras do machine learning (pt-BR) e Curso intensivo de ML, Sistemas de ML de produção (pt-BR): inferência estática e dinâmica, teste de implantação e monitoramento de pipelines.
- MLflow, Tracking e Model Registry; Python 3.15, unittest e venv (pt-BR); scikit-learn, Common pitfalls; SciPy, ks_2samp.
- Google SRE, Canarying Releases (Workbook, cap. 16) e Release Engineering (livro SRE, cap. 8).
- NC 21/IN01/DSIC/GSIPR (item 4.18, resumo criptográfico) e IN GSI/PR 3/2021 (arts. 28 a 36, gestão de mudanças).
- SGD, Guia Unificado de IA (2026): metodologia, implementação e acompanhamento de projetos de IA no setor público; rastreamento de experimentos e versionamento de modelos (p. 45 e 46), contêineres e infraestrutura como código (p. 48 e 50), plano de rollback e implantação segura (p. 52 e 53), alertas de drift, linhagem e plano de contingência (p. 55 e 56).
- Portaria SGD/MGI 1.070/2023, modelo de contratação de operação de infraestrutura e atendimento: Anexo I, item 4.1.7 (ferramentas de DevSecOps).
- Portaria Normativa CGU 179/2024: Comissão de Governança de Ciência de Dados da CGU, com a revisão e a supervisão técnica das soluções antes da produção.

Atividades sugeridas do curso e do TEC neste tópico

Curso e aula	Páginas	Foco
Ciência de dados e inteligência artificial para CGU (TI, Ciência de Dados) · **Capítulo 8 (Ciclo de vida e MLOps)**	p. 3 a 27	MLOps, práticas contínuas, testes, CRISP-ML(Q), monitoramento e ferramentas

Links do Estratégia: o LDI (Livro Digital Interativo) de cada curso, no objetivo da área. As páginas são as do PDF do capítulo baixado no botão Baixar curso do LDI, versão Completo.

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

Cadernos do TEC

Caderno	Assuntos cobrados (questões do Cebraspe em cada um)
Caderno do tópico, 74 questões tecconcursos.com.br/s/Q7BaxG	Git (42); Github e Gitlab (22); Gerência de Configuração de Software (10).
Caderno extra 1, 123 questões tecconcursos.com.br/s/Q7BcgZ	DevOps, IaC, Integração Contínua e Entrega Contínua (123).

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

Quem escreveu isso

Meu nome é Pietro Viecili. Auditor Federal de Finanças e Controle. aprovado em 2º lugar na CGU. Sou pai de 4 crianças e não abro mão da convivência com elas.

Em 1º de abril de 2022 eu recebi o salário às 10h da manhã. Às 15h44 eu tinha pouco mais de 1 real na conta. A luz estava atrasada há dois meses. a água no gato. a internet já cortada. Eu tinha ganhado uma passagem para fazer a prova da CGE-MS e não tinha dinheiro para chegar no aeroporto. Já era pai. e tinha acabado de fazer a prova da CGU sem saber o resultado.

Antes daquele dia eu estudei numa tábua de passar roupa. porque não tinha mesa. e dormia num colchonete no chão. Estudava de madrugada. no Natal. com bebê no colo. Foram 7 reprovações. Uma por 0.05 ponto. Outra porque eu troquei a folha de resposta da discursiva.

A virada não foi esforço. Eu já estava no limite. Foi parar de tratar concurso como prova de resistência e começar a estudar como atleta: estratégia. análise. planejamento e periodização. Veio o 2º lugar na CGU. o 1º no TJ-RO e na DPE-RO. a PF e a FAB. São 7 aprovações em 14 concursos. e hoje mais de 150 mentorados acompanhados com esse método.



Escrevi estes 143 resumos com o edital aberto item por item e as fontes oficiais do lado: o Manual de Orientações Técnicas. as instruções normativas da Secretaria Federal de Controle. a Lei 14.133. a Lei de Acesso à Informação. as normas de integridade pública e privada. os materiais da Escola Virtual de Governo e. nos resumos de TI. a documentação oficial das tecnologias cobradas. São 2.113 citações literais. cada uma conferida no texto oficial e com o artigo ou a página de onde saiu, porque é dali que o Cebraspe tira a frase e troca uma palavra.

Em 121 dos 143 tópicos existe pelo menos uma seção que você não vai achar em aula nenhuma. Essas eu precisei escrever do zero.

Você não nasceu pra ser mais um.
Nasceu pra ser o ZeroUm.

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

Você já tem o conteúdo certo

Os resumos te dão o recorte do edital e o conteúdo do jeito que a banca cobra. O que eles não decidem é a terça-feira. Quando o trabalho atropela a semana, quando o simulado volta com 54% e você não sabe qual das 13 disciplinas atacar primeiro, quando falta um mês e meio e você precisa escolher o que vai abandonar. Material nenhum decide isso por você, porque a decisão depende da sua rotina, do seu histórico de acerto e da sua data.

É isso que eu faço na Mentoria Pós-Edital CGU, com você, até 13 de dezembro:

- plano montado sobre este edital e sobre a sua rotina real, com meta diária de teoria, questão e revisão
- periodização até o pico na data da prova, do jeito que atleta treina para competição
- plantão de dúvidas e acesso direto a mim no WhatsApp
- módulo de discursiva, que vale metade da nota da prova
- Assinatura Premium do Estratégia Concursos inclusa

Dois meses, R$ 797 com a assinatura do Estratégia inclusa, ou R$ 627 sem ela.

O que você pagou por este material é abatido integralmente. Me chame no WhatsApp, confiro seu pedido e aplico o crédito.

Chamar o Pietro no WhatsApp

Silêncio e trabalho. VAMO.

AMOSTRA

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui

Esta foi uma amostra completa

Você acabou de ler um dos 143 resumos do Resumo Completo CGU, inteiro, do jeito que o aluno recebe: o recorte exato do edital, as citações literais conferidas na norma, os quadros do Cebraspe, as questões com gabarito e os cadernos do TEC.

O material completo cobre a P1 e a P2, iguais para todos, e as quatro P3: Auditoria, Ciência de Dados, Infraestrutura e Desenvolvimento. São mais de 2.100 citações literais e 3.053 questões com gabarito.

Só o Resumo Completo CGU

Os 143 resumos por R$ 197 à vista ou 12x de R$ 20,37.

Quero o Resumo Completo CGU

Missão CGU, com o Resumo incluso

Dois meses de mentoria individual até a prova, com plano de estudos, reuniões semanais, acesso a mim pelo WhatsApp, 3 simulados no formato Cebraspe e os 143 resumos inclusos. A partir de R$ 627 à vista.

Quero a Missão CGU

Ficou com dúvida? Me chame no WhatsApp.

Silêncio e trabalho. VAMO.

AMOSTRA

Resumo Completo CGU · Edição 2026

Mentoria ZeroUm

AMOSTRA · Resumo Completo CGU · material completo do tópico, sem cortes · garanta o seu aqui