databricks

Guia de Exame Databricks

Databricks Certified Machine Learning Associate

Objetivo deste Guia de Exame

Este guia fornece uma visão geral do exame e do que ele abrange para ajudar você a determinar se está preparado para realizá-lo. Este documento será atualizado sempre que houver alterações em um exame (e quando essas alterações entrarem em vigor), para que você possa se preparar. **Esta versão abrange a versão ativa em 28 de outubro de 2024. Verifique novamente duas semanas antes de fazer seu exame para ter certeza de que você tem a versão mais atual.**

Descrição do público

O exame de certificação Databricks Certified Machine Learning Associate avalia a capacidade de um indivíduo de usar o Databricks para executar tarefas básicas de machine learning. Isso inclui a capacidade de entender e usar o IT Databricks e seus recursos de machine learning, como AutoML, Unity Catalog e recursos selecionados do MLflow. Ele também avalia a capacidade de explorar dados e executar engenharia de características. Além disso, o exame avalia a construção de modelos por meio de treinamento, ajuste, avaliação e seleção. Por fim, é avaliada a capacidade de implantar modelos de machine learning. Espera-se que as pessoas aprovadas neste exame de certificação executem tarefas básicas de machine learning usando Databricks e suas ferramentas associadas.

Sobre o exame

- Número de itens: 48 questões pontuadas de múltipla escolha ou múltipla seleção
- Limite de tempo: 90 minutos
- Taxa de inscrição: $200
- Método: supervisionado on-line
- Materiais de apoio para o teste: nenhum é permitido
- Pré-requisito: nenhum; participação no curso e seis meses de experiência prática realizando as tarefas mencionadas na Descrição do Exame a seguir são altamente recomendados. Além disso, consulte o item Preparação Recomendada neste documento.
- Validade: 2 anos.
- Recertificação: a recertificação é necessária a cada dois anos para manter seu status de certificado. Para se recertificar, é preciso fazer o exame completo que está atualmente disponível. Consulte a seção "Preparando-se para o exame" na página do exame para se preparar para fazer o exame novamente.

- Conteúdo sem pontuação: os exames podem incluir itens não pontuados para coletar informações estatísticas para uso futuro. Esses itens não são identificados no formulário e não impactam a sua pontuação. Tempo adicional é levado em consideração para esse conteúdo.

Preparação recomendada

- Conduzido por instrutor: machine learning com Databricks
- Individual (disponível na Databricks Academy): machine learning com Databricks
- Conhecimento prático de Python e das principais bibliotecas que oferecem suporte ao machine learning, como scikit-learn e SparkML
- Conhecimento prático do Unity Catalog e outros recursos de geranciamento de dados do Databricks, como Delta Live Tables
- Familiaridade com os principais tópicos em aprendizado de máquina na documentação do Databricks

Detalhes do exame

Seção 1: Machine Learning com Databricks

- Identificar as melhores práticas de uma estratégia de MLOps
- Identificar as vantagens de usar runtimes de ML
- Identificar como o AutoML facilita a seleção de modelos/recursos.
- Identificar as vantagens que o AutoML traz ao processo de desenvolvimento de modelos
- Identificar os benefícios de criar tabelas do feature store no nível da conta no Unity Catalog no Databricks em comparação com o nível de workspace
- Criar uma tabela do feature store no Unity Catalog
- Gravar dados em uma tabela de repositório de recursos
- Treinar um modelo com recursos de uma tabela do Feature Store.
- Aplicar um modelo usando recursos de uma tabela do Feature Store.
- Descrever as diferenças entre tabelas de recursos online e offline
- Identificar a melhor execução usando a API de cliente do MLflow.
- Registrar manualmente as métricas de log, os artefatos e os modelos em uma execução do MLflow.
- Identificar informações disponíveis na interface de usuário do MLFlow
- Registrar um modelo usando a API do cliente MLflow no catálogo do Unity
- Identificar os benefícios de registrar modelos no registro do Unity Catalog em vez do registro do workspace
- Identificar cenários em que promover código é preferível a promover modelos e vice-versa
- Definir ou remover uma tag para um modelo
- Promover um modelo desafiante para um modelo campeão usando aliases

Seção 2: Processamento de Dados

- Calcular estatísticas de resumo em um Spark DataFrame usando `.summary()` ou resumos de dados dbutils
- Remover outliers de um Spark DataFrame com base no desvio padrão ou IQR
- Criar visualizações para características categóricas ou contínuas
- Comparar dois recursos categóricos ou contínuos usando o método apropriado
- Comparar e contrastar a imputação de valores ausentes com a média, mediana ou valor de moda
- Imputar valores ausentes com a média, mediana ou valor de moda
- Usar codificação one-hot para recursos categóricos
- Identificar e explicar os tipos de modelos ou conjuntos de dados para os quais a codificação one-hot é ou não apropriada.
- Identificar cenários onde a transformação de log scale é apropriada

Seção 3: Desenvolvimento de Modelo

- Usar fundamentos de ML para selecionar o algoritmo apropriado para um determinado cenário de modelo
- Identificar métodos para mitigar o desequilíbrio de dados em dados de treinamento
- Comparar estimadores e transformadores
- Desenvolver um pipeline de treinamento
- Usar a operação `fmin do Hyperopt para ajustar os` hiperparâmetros de um modelo
- Executar busca aleatória, em grade ou bayesiana como um método para ajustar hiperparâmetros.
- Paralelizar modelos de nó único para ajuste de hiperparâmetros
- Descrever os benefícios e desvantagens de usar validação cruzada em vez de uma divisão de treino-validação.
- Realizar a validação cruzada como parte do ajuste de modelo.
- Identificar o número de modelos que estão sendo ensinados em conjunto com um processo de busca em grade e validação cruzada.
- Usar métricas de classificação comuns: F1, Log Loss, ROC/AUC, etc.
- Usar métricas de regressão comuns: RMSE, MAE, R ao quadrado, etc.
- Escolha a métrica mais apropriada para o objetivo de um determinado cenário
- Identificar a necessidade de exponenciar variáveis transformadas em log antes de calcular métricas de avaliação ou interpretar previsões
- Avaliar o impacto da complexidade do modelo e do tradeoff de viés e variância no desempenho do modelo

Seção 4: Implantação de modelos

- Identificar as diferenças e vantagens das abordagens de model serving: lotes, tempo real e streaming
- Implantar um modelo personalizado em um endpoint modelo
- Use pandas para realizar inferência em lote
- Identificar como a inferência de streaming é realizada com Delta Live Tables
- Implantar e consultar um modelo para inferência em tempo real
- Dividir dados entre endpoints para inferência em tempo real

Perguntas de exemplo

Essas perguntas são semelhantes às do teste e dão a você uma noção geral de como as perguntas são feitas neste exame. Incluem os objetivos do exame, como estão indicados no guia e oferecem um exemplo de pergunta que se alinhe ao objetivo. O guia do exame lista todos os objetivos que podem ser abordados em um exame. A melhor maneira de se preparar para um exame de certificação é revisar sua estrutura conforme descrita no guia.

Pergunta 1

Objetivo: Criar uma tabela no repositório de recursos no Unity Catalog.

Um cientista de dados quer criar uma tabela de recursos para usar em seus modelos. O time está trabalhando em um espaço de trabalho com o Unity Catalog habilitado e querem que essa tabela de recursos seja armazenada e governada por ele.

Qual é a maneira correta de criar esta tabela de recursos?

A. Criar uma tabela Delta com dados, como de costume, e use o **`método register_table do`** FeatureStoreClient **`em Python para registrá-la como uma tabela de recursos no Unity Catalog.`**
B. Criar uma tabela Delta vazia no Unity Catalog com a cláusula **`AS FEATURE STORE`** via SQL e, em seguida, gravar dados nela.
C. Usar o método **`create_table`** do FeatureEngineeringClient em Python para criar a tabela e, em seguida, gravar dados nela.
D. Criar uma tabela Delta com dados no Unity Catalog e usar o comando **`ALTER TABLE`** no SQL para configurá-la como uma tabela de recursos com a cláusula **`SET AS FEATURE STORE`** .

Pergunta 2

Objetivo: Imputar valores ausentes com a moda, média ou mediana

Um cientista de dados precisa imputar os valores ausentes em um recurso contínuo. O time quer fazer isso com o mínimo de esforço, mas com resultados corretos.

Qual estratégia devem usar?

A. Usar sklearn **`SimpleImputer,`** que seleciona automaticamente a melhor metodologia com base na distribuição de recursos
B. Examinar a distribuição dos valores e selecionar a imputação apropriada após a revisão
C. Use **`.mean()`**, que é a imputação mais apropriada em colunas contínuas
D. Usar **`.mode()`**, que é a imputação mais apropriada em colunas contínuas

Pergunta 3

Objetivo: Identificar métodos para mitigar o desequilíbrio de dados em dados de treinamento.

Um cientista de dados está trabalhando em um projeto de machine learning para desenvolver um modelo que prevê se um cliente abandonará um serviço de assinatura. O conjunto de dados é altamente desequilibrado, com apenas 10% das instâncias representando clientes que cancelam o serviço. Eles querem garantir que seu modelo identifique efetivamente a classe minoritária sem ser tendencioso em relação à classe majoritária.

Qual estratégia atenua diretamente o viés do modelo em direção aos clientes que não cancelam o serviço devido ao desequilíbrio de classes?

A. Normalizar os recursos para garantir que estejam na mesma escala, melhorando o desempenho do modelo.
B. Usar o aprendizado sensível ao custo atribuindo um custo de classificação incorreta mais alto à classe minoritária durante o treinamento do modelo.
C. Aumentar o tamanho do conjunto de dados de treinamento coletando mais dados sobre clientes que não cancelaram.
D. Usar um modelo mais simples para reduzir o overfitting, garantindo que ele generalize melhor para a classe minoritária.

Pergunta 4

Objetivo: Identificar o número de modelos que estão sendo treinados em conjunto com um processo de busca em grade e validação cruzada.

Um cientista de dados está ajustando um modelo de Máquina de Vetores de Suporte (SVM)

usando validação cruzada de 5 vezes e `GridSearchCV` no scikit-learn. A grade de parâmetros inclui três hiperparâmetros para otimizar: C com valores `[0.1, 1, 10]`, kernel com escolhas `['linear', 'rbf']` e gama com valores `[0.01, 0.1, 1]`.

Quantos modelos diferentes serão treinados no total?

A. 90
B. 18
C. 1
D. Nenhuma das opções anteriores.

Pergunta 5

Objetivo: Identificar como a inferência de streaming é realizada com Delta Live Tables.

Uma empresa tem uma plataforma de podcast com milhares de usuários. A empresa implementou um algoritmo de detecção de anomalias para detectar baixo engajamento no podcast com base em uma janela de 10 minutos de eventos do usuário, como ouvir, pausar e sair do podcast. Um engenheiro de machine learning quer implantar esse modelo em um pipeline de dados de produção que precisa lidar com até dezenas de milhares de eventos por segundo. Como o volume de eventos flutua ao longo do dia, o engenheiro precisa que a computação do pipeline seja redimensionada dinamicamente.

Qual abordagem de projeto de pipeline atende a esses requisitos?

A. Criar um pipeline do Delta Live Tables que aplique o algoritmo como um Spark UDF.
B. Criar um Job de Structured Streaming que aplique o algoritmo como um Spark UDF.
C. Criar um endpoint de model serving, criar um pipeline do Delta Live Tables que chame uma UDF personalizada que invoque o endpoint.
D. Criar um endpoint de model serving, criar um Job de Structured Streaming que chame uma UDF personalizada que invoque o endpoint.

Respostas
Pergunta 1: C
Pergunta 2: B
Pergunta 3: B
Pergunta 4: A
Pergunta 5: A