databricks

Guia do exame da Databricks

Databricks Certified Generative AI Engineer Associate

Fornecer feedback sobre o Guia do exame

Finalidade deste Guia do exame

A finalidade deste guia é apresentar uma visão geral do exame e das questões para ajudar você a determinar se está preparado. 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 cobre a versão atualmente ativa a partir de 1º de junho de 2024. Volte duas semanas antes de fazer o exame para ter certeza de que você tem a versão mais atual.**

Descrição do público

O exame de certificação de Databricks Certified Generative AI Engineer Associate avalia a capacidade de um indivíduo de projetar e implementar soluções habilitadas para LLM usando Databricks. Isso inclui a decomposição de problemas para dividir requisitos complexos em tarefas gerenciáveis, bem como a escolha de modelos, ferramentas e abordagens apropriadas do atual cenário de IA generativa para o desenvolvimento de soluções abrangentes. Ele também avalia ferramentas específicas do Databricks, como Vector Search para pesquisas de similaridade semântica, Serviço de Modelos para implantar modelos e soluções, MLflow para gerenciar o ciclo de vida da solução e Unity Catalog para governança de dados. Espera-se que as pessoas aprovadas neste exame criem e implantem aplicativos RAG de alto desempenho e cadeias LLM que aproveitem ao máximo Databricks e seu conjunto de ferramentas.

Sobre o exame

- Número de itens: 45 questões de múltipla escolha ou múltipla seleção
- Limite de tempo: 90 minutos
- Taxa de inscrição: $200
- Método de realização: supervisionado on-line
- Materiais de consulta: nenhum é permitido
- Pré-requisito: nenhum exigido; participação no curso e seis meses de experiência prática em Databricks são 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.

Preparação recomendada

- Todos os cursos atuais da Databricks Academy relacionados à função de aprendiz de IA Generativa, especificamente, Engenharia de IA Generativa com Databricks
- Conhecimento dos LLMs atuais e suas capacidades
- Conhecimento de engenharia de prompts, geração de prompts e avaliação
- Conhecimento de ferramentas e serviços on-line relacionados atuais, como LangChain, Hugging Face Transformers, etc.
- Conhecimento funcional de Python e suas bibliotecas que têm suporte para o aplicativo RAG e o desenvolvimento da cadeia LLM
- Conhecimento prático de APIs atuais para preparação de dados, encadeamento de modelos, etc.
- Recursos relevantes da documentação da Databricks

Visão geral do exame

Seção 1: Desenvolvimento de aplicações

- Projetar um prompt que provoque uma resposta formatada especificamente
- Selecionar tarefas de modelo para atender a um determinado requisito de negócios
- Selecionar componentes da cadeia para uma entrada e saída de modelo desejada
- Traduzir metas de caso de uso de negócios em uma descrição das entradas e saídas desejadas para o pipeline de IA
- Definir e ordenar ferramentas que reúnam conhecimento ou realizem ações para raciocínio em vários estágios

Seção 2: Preparação de dados

- Aplicar uma estratégia de segmentação para uma determinada estrutura de documento e restrições de modelo
- Filtrar conteúdo desnecessário em documentos de origem que degrada a qualidade de um aplicativo RAG
- Escolher o pacote Python apropriado para extrair o conteúdo do documento dos dados e do formato de origem fornecidos
- Definir operações e sequência para gravar determinado texto segmentado em tabelas Delta Lake no Unity Catalog
- Identificar os documentos de origem necessários que fornecem o conhecimento e a qualidade necessários para um determinado aplicativo RAG
- Identificar pares de prompt/resposta que se alinham a uma determinada tarefa do modelo
- Usar ferramentas e métricas para avaliar o desempenho de recuperação

Seção 3: Desenvolvimento de aplicações

- Criar ferramentas necessárias para extração de dados para uma determinada necessidade de recuperação de dados
- Selecionar ferramentas LangChain/similares para uso em um aplicativo de IA generativa
- Identificar como os formatos de prompt podem alterar as saídas e os resultados do modelo
- Avaliar qualitativamente as respostas para identificar problemas comuns, como qualidade e segurança
- Selecionar estratégia de segmentação com base no modelo e na avaliação de recuperação
- Aumentar um prompt com contexto adicional a partir da entrada de um usuário com base em campos, termos e intenções principais
- Criar um prompt que ajusta a resposta de um LLM de uma linha de base para uma saída desejada
- Implementar grades de proteção de LLM para evitar resultados negativos
- Escrever metaprompts que minimizem alucinações ou vazamento de dados privados
- Criar modelos de prompt de agente expondo funções disponíveis
- Selecionar o melhor LLM com base nos atributos da aplicação a ser desenvolvida
- Selecionar um comprimento de contexto do modelo de incorporação com base nos documentos de origem, nas consultas esperadas e na estratégia de otimização
- Selecionar um modelo de um hub de modelos ou marketplace para uma tarefa com base em metadados de modelo/cartões de modelo
- Selecionar o melhor modelo para uma determinada tarefa com base em métricas comuns geradas em experimentos

Seção 4: Montagem e implantação de aplicações

- Codificar uma cadeia usando um modelo pyfunc com pré e pós-processamento
- Controlar o acesso ao recurso a partir de endpoints de servir modelo
- Codificar uma cadeia simples de acordo com os requisitos
- Codificar uma cadeia simples usando LangChain
- Escolher os elementos básicos necessários para criar um aplicativo RAG: tipo de modelo, modelo de incorporação, recuperador, dependências, exemplos de entrada, assinatura de modelo
- Registrar o modelo no Unity Catalog usando MLflow
- Sequenciar os passos necessários para implantar um endpoint para um aplicativo RAG básico
- Criar e consultar um índice de Busca Vetorial
- Identificar como atender a um aplicativo de LLM que aproveita as APIs do Modelo de Base
- Identificar recursos necessários para fornecer funcionalidades para um aplicativo RAG

Seção 5: Governança

- Usar técnicas de mascaramento como medidas de segurança para atender a um objetivo de desempenho
- Selecionar técnicas de medidas de segurança para proteção contra entradas de usuários mal-intencionados para um aplicativo de IA generativa
- Recomendar uma alternativa para mitigação de texto problemático em uma fonte de dados que alimenta um aplicativo RAG
- Usar requisitos legais/de licenciamento para fontes de dados para evitar riscos legais

Seção 6: Avaliação e monitoramento

- Selecionar uma escolha de LLM (tamanho e arquitetura) com base em um conjunto de métricas de avaliação quantitativa
- Selecionar as principais métricas a serem monitoradas para um cenário específico de implantação de LLM
- Avaliar o desempenho do modelo em um aplicativo RAG usando MLflow
- Usar o log de inferência para avaliar o desempenho do aplicativo RAG implantado
- Usar recursos Databricks para controlar os custos de LLM para aplicativos RAG

Exemplos de perguntas

Essas perguntas são semelhantes aos itens de perguntas reais 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 a estrutura do exame no guia do exame.

Pergunta 1

Objetivo: Aplicar uma estratégia de agrupamento para uma determinada estrutura de documento e restrições de modelo

Um engenheiro de IA generativa está carregando 150 milhões de incorporações em um banco de dados vetorial que aceita um máximo de 100 milhões.

Quais DUAS ações ele pode tomar para reduzir a contagem de registros?

A. Aumentar o tamanho do bloco do documento
B. Diminuir a sobreposição entre blocos
C. Diminuir o tamanho do bloco do documento
D. Aumentar a sobreposição entre blocos
E. Usar um modelo de incorporação menor

Pergunta 2

Objetivo: Identificar os documentos de origem necessários que forneçam o conhecimento e a qualidade necessários para um determinado aplicativo RAG.

Um engenheiro de IA generativa está avaliando as respostas de um aplicativo de IA generativa voltado para o cliente que ele está desenvolvendo para ajudar na venda de peças automotivas. O aplicativo exige que o cliente insira `explicitamente account_id` e `transaction_id` para responder a perguntas. Após o lançamento inicial, o feedback dos clientes foi de que o aplicativo se saiu bem em responder aos detalhes do pedido e do faturamento, mas não respondeu com precisão às perguntas sobre frete e data prevista de chegada.

Qual dos captadores a seguir melhoraria a capacidade do aplicativo de responder a essas perguntas?

A. Criar uma loja vetorial que inclua as políticas de envio da empresa e as condições de pagamento para todas as peças automotivas
B. Criar uma tabela repositório de recursos com `transaction_id` como chave principal preenchida com dados de fatura e data de entrega esperada
C. Fornecer dados de exemplo para datas de chegada esperadas como um dataset de ajuste e, em seguida, ajustar periodicamente o modelo para que ele atualize os informação de envio
D. Alterar o prompt de bate-papo para inserir quando o pedido foi feito e instruir o modelo a adicionar 14 dias a isso, pois nenhum método de envio deve exceder 14 dias

Pergunta 3

Objetivo: Escolher o pacote Python apropriado para extrair conteúdo para documentos com os dados e o formato de origem fornecidos.

Um engenheiro de IA generativa está criando um aplicativo RAG que contará com o contexto recuperado de documentos de origem que foram digitalizados e salvos como arquivos de imagem em formatos como .jpeg ou .png. Ele quer desenvolver uma solução usando a menor quantidade de linhas de código.

Qual pacote Python deve ser usado para extrair o texto dos documentos de origem?

A. beautifulsoup
B. scrapy
C. pytesseract
D. pyquery

Pergunta 4

Objetivo: Selecionar um comprimento de contexto do modelo de incorporação com base nos documentos de origem, nas consultas esperadas e na estratégia de otimização

Um engenheiro de IA generativa está criando um aplicativo baseado em LLM. Os documentos para seu retriever foram divididos com um máximo de 512 tokens cada. O engenheiro de IA generativa sabe que o custo e a latência são mais importantes do que a qualidade para essa aplicação. Ele tem vários níveis de comprimento de contexto para escolher.

Qual atenderá sua necessidade?

A. Tamanho do contexto 512: o menor modelo é de 0,13 GB e uma dimensão de incorporação de 384
B. Comprimento do contexto 514: o menor modelo é de 0,44 GB e uma dimensão de incorporação de 768
C. Comprimento do contexto 2048: o menor modelo é de 11 GB e uma dimensão de incorporação de 2560
D. Tamanho do contexto 32768: o menor modelo é de 14 GB e uma dimensão de incorporação de 4096

Pergunta 5

Objetivo: Selecione o melhor LLM com base nos atributos da aplicação a ser desenvolvida

Um Engenheiro de IA Generativa gostaria de criar um aplicativo que possa atualizar um campo de memorando com cerca de um parágrafo de comprimento para apenas um único resumo que mostre a intenção do campo de memorando, mas que se encaixe no front-end do aplicativo.

Com qual categoria de tarefa de Processamento de Linguagem Natural ele deve avaliar potenciais LLMs para essa aplicação?

A. text2text Generation
B. Sentencizer
C. Classificação de textos
D. Sumarização

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