Databricks-Generative-AI-Engineer-Associate的中関連問題 & Databricks-Generative-AI-Engineer-Associate赤本勉強

2025年GoShikenの最新Databricks-Generative-AI-Engineer-Associate PDFダンプおよびDatabricks-Generative-AI-Engineer-Associate試験エンジンの無料共有：https://drive.google.com/open?id=1cTNRZBd6UPqH9D6YvcOW8kPXDbf7VJhB

現在の仕事と現在の生活に飽きていますか？ 便利な証明書を入手してください！ Databricks-Generative-AI-Engineer-Associate学習ガイドは、目標を達成するのに役立つ最高の製品です。 試験に合格し、Databricks-Generative-AI-Engineer-Associate学習教材で認定を取得すると、大企業で満足のいく仕事に応募し、高い給与と高い利益で上級職に就くことができます。 優れたDatabricks Databricks-Generative-AI-Engineer-Associateスタディガイドにより、受験者は、余分な時間とエネルギーを無駄にせずに効率的にテストを準備するための明確な学習方向を得ることができます。

Databricks Databricks-Generative-AI-Engineer-Associate 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• 評価と監視: このトピックでは、LLM の選択と主要なメトリックについて説明します。さらに、Generative AI エンジニアはモデルのパフォーマンスの評価について学習します。最後に、このトピックには推論ログと Databricks 機能の使用に関するサブトピックが含まれています。
トピック 2	• データ準備: Generative AI エンジニアは、特定のドキュメント構造とモデル制約のチャンキング戦略について説明します。このトピックでは、ソース ドキュメント内の不要なコンテンツのフィルター処理にも重点を置いています。最後に、Generative AI エンジニアは、提供されたソース データと形式からドキュメント コンテンツを抽出する方法についても学習します。
トピック 3	• アプリケーション開発: このトピックでは、Generative AI エンジニアは、データの抽出に必要なツール、Langchain • 類似ツール、一般的な問題を特定するための応答の評価について学習します。さらに、このトピックには、LLM の応答の調整、LLM ガードレール、およびアプリケーションの属性に基づいた最適な LLM に関する質問が含まれています。

トピック４	• ガバナンス:試験を受けるジェネレーティブ AI エンジニアは、このトピックのマスキング手法、ガードレール手法、および法的 • ライセンス要件に関する知識を習得します。

>> Databricks-Generative-AI-Engineer-Associate的中関連問題 <<

Databricks-Generative-AI-Engineer-Associate赤本勉強 & Databricks-Generative-AI-Engineer-Associate復習範囲

多くの時間とお金がいらなくて２０時間だけあって楽に一回にDatabricksのDatabricks-Generative-AI-Engineer-Associate認定試験を合格できます。GoShikenが提供したDatabricksのDatabricks-Generative-AI-Engineer-Associate試験問題と解答が真実の試験の練習問題と解答は最高の相似性があります。

Databricks Certified Generative AI Engineer Associate 認定 Databricks-Generative-AI-Engineer-Associate 試験問題 (Q39-Q44):

質問 # 39
A Generative AI Engineer I using the code below to test setting up a vector store:

```
from databricks.vector_search.client import VectorSearchClient

vsc = VectorSearchClient()

vsc.create_endpoint(
     name= "vector_search_test",
     endpoint_type= "STANDARD"
)
```

Assuming they intend to use Databricks managed embeddings with the default embedding model, what should be the next logical function call?

- A. vsc.create_direct_access_index()
- B. vsc.get_index()
- C. vsc.similarity_search()
- D. vsc.create_delta_sync_index()

正解：D

解説：
Context: The Generative AI Engineer is setting up a vector store using Databricks' VectorSearchClient. This is typically done to enable fast and efficient retrieval of vectorized data for tasks like similarity searches.
Explanation of Options:
* Option A: vsc.get_index(): This function would be used to retrieve an existing index, not create one, so it would not be the logical next step immediately after creating an endpoint.
* Option B: vsc.create_delta_sync_index(): After setting up a vector store endpoint, creating an index is necessary to start populating and organizing the data. The create_delta_sync_index() function specifically creates an index that synchronizes with a Delta table, allowing automatic updates as the data changes. This is likely the most appropriate choice if the engineer plans to use dynamic data that is updated over time.
* Option C: vsc.create_direct_access_index(): This function would create an index that directly accesses the data without synchronization. While also a valid approach, it's less likely to be the next logical step if the default setup (typically accommodating changes) is intended.
* Option D: vsc.similarity_search(): This function would be used to perform searches on an existing index; however, an index needs to be created and populated with data before any search can be conducted.
Given the typical workflow in setting up a vector store, the next step after creating an endpoint is to establish an index, particularly one that synchronizes with ongoing data updates, henceOption B.

質問 # 40
A Generative AI Engineer is building a RAG application that will rely on context retrieved from source documents that are currently in PDF format. These PDFs can contain both text and images. They want to develop a solution using the least amount of lines of code.
Which Python package should be used to extract the text from the source documents?

- A. unstructured
- B. numpy
- C. beautifulsoup
- D. flask

正解：C

解説：
* Problem Context: The engineer needs to extract text from PDF documents, which may contain both text and images. The goal is to find a Python package that simplifies this task using the least amount of code.
* Explanation of Options:
* Option A: flask: Flask is a web framework for Python, not suitable for processing or extracting content from PDFs.
* Option B: beautifulsoup: Beautiful Soup is designed for parsing HTML and XML documents, not PDFs.
* Option C: unstructured: This Python package is specifically designed to work with unstructured data, including extracting text from PDFs. It provides functionalities to handle various types of content in documents with minimal coding, making it ideal for the task.
* Option D: numpy: Numpy is a powerful library for numerical computing in Python and does not provide any tools for text extraction from PDFs.
Given the requirement,Option C(unstructured) is the most appropriate as it directly addresses the need to efficiently extract text from PDF documents with minimal code.

質問 # 41
A Generative AI Engineer has been asked to design an LLM-based application that accomplishes the following business objective: answer employee HR questions using HR PDF documentation.
Which set of high level tasks should the Generative AI Engineer's system perform?

- A. Use an LLM to summarize HR documentation. Provide summaries of documentation and user query into an LLM with a large context window to generate a response to the user.
- B. Split HR documentation into chunks and embed into a vector store. Use the employee question to retrieve best matched chunks of documentation, and use the LLM to generate a response to the employee based upon the documentation retrieved.
- C. Calculate averaged embeddings for each HR document, compare embeddings to user query to find the best document. Pass the best document with the user query into an LLM with a large context window to generate a response to the employee.
- D. Create an interaction matrix of historical employee questions and HR documentation. Use ALS to factorize the matrix and create embeddings. Calculate the embeddings of new queries and use them to find the best HR documentation. Use an LLM to generate a response to the employee question based upon the documentation retrieved.

正解：B

解説：
To design an LLM-based application that can answer employee HR questions using HR PDF documentation, the most effective approach is option D. Here's why:
* Chunking and Vector Store Embedding:HR documentation tends to be lengthy, so splitting it into smaller, manageable chunks helps optimize retrieval. These chunks are then embedded into avector store(a database that stores vector representations of text). Each chunk of text is transformed into an embeddingusing a transformer-based model, which allows for efficient similarity-based retrieval.
* Using Vector Search for Retrieval:When an employee asks a question, the system converts their query into an embedding as well. This embedding is then compared with the embeddings of the document chunks in the vector store. The most semantically similar chunks are retrieved, which ensures that the answer is based on the most relevant parts of the documentation.
* LLM to Generate a Response:Once the relevant chunks are retrieved, these chunks are passed into the LLM, which uses them as context to generate a coherent and accurate response to the employee's question.
* Why Other Options Are Less Suitable:
* A (Calculate Averaged Embeddings): Averaging embeddings might dilute important information. It doesn't provide enough granularity to focus on specific sections of documents.
* B (Summarize HR Documentation): Summarization loses the detail necessary for HR-related queries, which are often specific. It

would likely miss the mark for more detailed inquiries.
* C (Interaction Matrix and ALS): This approach is better suited for recommendation systems and not for HR queries, as it's focused on collaborative filtering rather than text-based retrieval.
Thus, option D is the most effective solution for providing precise and contextual answers based on HR documentation.

質問 # 42
A Generative AI Engineer is testing a simple prompt template in LangChain using the code below, but is getting an error.

```
from langchain.chains import LLMChain
from langchain_community.llms import OpenAI
from langchain_core.prompts import PromptTemplate

prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
    input_variables=["adjective"],
    template=prompt_template
)

llm = LLMChain(prompt=prompt)
llm.generate([{"adjective": "funny"}])
```

Assuming the API key was properly defined, what change does the Generative AI Engineer need to make to fix their chain?

- A.
```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
    input_variables=["adjective"],
    template=prompt_template
)

llm = LLMChain(prompt=prompt.format("funny"))
llm.generate()
```
- B.
```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
    input_variables=["adjective"],
    template=prompt_template
)

llm = LLMChain(llm=OpenAI(), prompt=prompt)
llm.generate([{"adjective": "funny"}])
```
- C.
```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
    input_variables=["adjective"],
    template=prompt_template
)

llm = LLMChain(prompt=prompt)
llm.generate("funny")
```

```
prompt_template = "Tell me a {adjective} joke"

prompt = PromptTemplate(
    input_variables=["adjective"],
    template=prompt_template
    llm=OpenAI()
)

llm = LLMChain(prompt=prompt)
```

- D. llm.generate([{"adjective": "funny"}])

正解：D

解説：
To fix the error in the LangChain code provided for using a simple prompt template, the correct approach is Option C. Here's a detailed breakdown of why Option C is the right choice and how it addresses the issue:
* Proper Initialization: In Option C, the LLMChain is correctly initialized with the LLM instance specified as OpenAI(), which likely represents a language model (like GPT) from OpenAI. This is crucial as it specifies which model to use for generating responses.
* Correct Use of Classes and Methods:
* The PromptTemplate is defined with the correct format, specifying that adjective is a variable within the template. This allows dynamic insertion of values into the template when generating text.
* The prompt variable is properly linked with the PromptTemplate, and the final template string is passed correctly.
* The LLMChain correctly references the prompt and the initialized OpenAI() instance, ensuring that the template and the model are properly linked for generating output.
Why Other Options Are Incorrect:
* Option A: Misuses the parameter passing in generate method by incorrectly structuring the dictionary.
* Option B: Incorrectly uses prompt.format method which does not exist in the context of LLMChain and PromptTemplate configuration, resulting in potential errors.
* Option D: Incorrect order and setup in the initialization parameters for LLMChain, which would likely lead to a failure in recognizing the correct configuration for prompt and LLM usage.
Thus, Option C is correct because it ensures that the LangChain components are correctly set up and integrated, adhering to proper syntax and logical flow required by LangChain's architecture. This setup avoids common pitfalls such as type errors or method misuses, which are evident in other options.

質問 # 43
A Generative AI Engineer has been asked to build an LLM-based question-answering application. The application should take into account new documents that are frequently published. The engineer wants to build this application with the least cost and least development effort and have it operate at the lowest cost possible.
Which combination of chaining components and configuration meets these requirements?

- A. For the application a prompt, a retriever, and an LLM are required. The retriever output is inserted into the prompt which is given to the LLM to generate answers.
- B. For the application a prompt, an agent and a fine-tuned LLM are required. The agent is used by the LLM to retrieve relevant content that is inserted into the prompt which is given to the LLM to generate answers.
- C. The LLM needs to be frequently with the new documents in order to provide most up-to-date answers.
- D. For the question-answering application, prompt engineering and an LLM are required to generate answers.

正解：A

解説：
Problem Context: The task is to build an LLM-based question-answering application that integrates new documents frequently with minimal costs and development efforts.
Explanation of Options:
* Option A: Utilizes a prompt and a retriever, with the retriever output being fed into the LLM. This setup is efficient because it dynamically updates the data pool via the retriever, allowing the LLM to provide up-to-date answers based on the latest documents without needing tofrequently retrain the model. This method offers a balance of cost-effectiveness and functionality.
* Option B: Requires frequent retraining of the LLM, which is costly and labor-intensive.
* Option C: Only involves prompt engineering and an LLM, which may not adequately handle the requirement for incorporating new documents unless it's part of an ongoing retraining or updating mechanism, which would increase costs.
* Option D: Involves an agent and a fine-tuned LLM, which could be overkill and lead to higher development and operational costs.
Option Ais the most suitable as it provides a cost-effective, minimal development approach while ensuring the application remains

up-to-date with new information.

質問 # 44
.....

DatabricksのDatabricks-Generative-AI-Engineer-Associateのオンラインサービスのスタディガイドを買いたかったら、GoShikenを買うのを薦めています。GoShikenは同じ作用がある多くのサイトでリーダーとしているサイトで、最も良い品質と最新のトレーニング資料を提供しています。弊社が提供したすべての勉強資料と他のトレーニング資料はコスト効率の良い製品で、サイトが一年間の無料更新サービスを提供します。ですから、弊社のトレーニング製品はあなたが試験に合格することを助けにならなかったら、全額で返金することを保証します。

Databricks-Generative-AI-Engineer-Associate赤本勉強: https://www.goshiken.com/Databricks/Databricks-Generative-AI-Engineer-Associate-mondaishu.html

- Databricks-Generative-AI-Engineer-Associate赤本勉強 □ Databricks-Generative-AI-Engineer-Associateテスト資料 ☎ Databricks-Generative-AI-Engineer-Associate模擬資料 □ サイト➥ www.japancert.com □で ➡ Databricks-Generative-AI-Engineer-Associate □□□問題集をダウンロードDatabricks-Generative-AI-Engineer-Associate模擬資料
- 正確的-ハイパスレートのDatabricks-Generative-AI-Engineer-Associate的中関連問題試験-試験の準備方法 Databricks-Generative-AI-Engineer-Associate赤本勉強 □ ウェブサイト《 www.goshiken.com 》を開き、➤ Databricks-Generative-AI-Engineer-Associate □を検索して無料でダウンロードしてくださいDatabricks-Generative-AI-Engineer-Associate復習範囲
- 正確的-ハイパスレートのDatabricks-Generative-AI-Engineer-Associate的中関連問題試験-試験の準備方法 Databricks-Generative-AI-Engineer-Associate赤本勉強 □ （ www.it-passports.com ）で☀ Databricks-Generative-AI-Engineer-Associate □☀□を検索して、無料でダウンロードしてくださいDatabricks-Generative-AI-Engineer-Associate合格率書籍
- 一番いいDatabricks-Generative-AI-Engineer-Associate的中関連問題 - 資格試験におけるリーダーオファー - 正確的なDatabricks Databricks Certified Generative AI Engineer Associate □ ▹ www.goshiken.com ◃にて限定無料の { Databricks-Generative-AI-Engineer-Associate }問題集をダウンロードせよDatabricks-Generative-AI-Engineer-Associateリンクグローバル
- Databricks-Generative-AI-Engineer-Associateテスト資料 □ Databricks-Generative-AI-Engineer-Associate専門知識訓練 □ Databricks-Generative-AI-Engineer-Associate日本語認定 □ ➠ www.passtest.jp □で➤ Databricks-Generative-AI-Engineer-Associate □を検索して、無料でダウンロードしてくださいDatabricks-Generative-AI-Engineer-Associate一発合格
- Databricks-Generative-AI-Engineer-Associate合格率書籍 □ Databricks-Generative-AI-Engineer-Associate関連資格知識 □ Databricks-Generative-AI-Engineer-Associate赤本勉強 □ 「 www.goshiken.com 」は、⇛ Databricks-Generative-AI-Engineer-Associate ⇚を無料でダウンロードするのに最適なサイトですDatabricks-Generative-AI-Engineer-Associateテスト資料
- 一番優秀なDatabricks-Generative-AI-Engineer-Associate的中関連問題試験-試験の準備方法-高品質なDatabricks-Generative-AI-Engineer-Associate赤本勉強 □ 《 Databricks-Generative-AI-Engineer-Associate 》を無料でダウンロード➤ www.it-passports.com □で検索するだけDatabricks-Generative-AI-Engineer-Associate受験練習参考書
- Databricks-Generative-AI-Engineer-Associate合格対策 □ Databricks-Generative-AI-Engineer-Associate一発合格 □ Databricks-Generative-AI-Engineer-Associate一発合格 □⇛ www.goshiken.com ⇚に移動し、（ Databricks-Generative-AI-Engineer-Associate ）を検索して、無料でダウンロード可能な試験資料を探しますDatabricks-Generative-AI-Engineer-Associate模擬資料
- Databricks-Generative-AI-Engineer-Associateリンクグローバル □ Databricks-Generative-AI-Engineer-Associateリンクグローバル □ Databricks-Generative-AI-Engineer-Associate関連資格知識 □ 検索するだけで□ www.japancert.com □から➽ Databricks-Generative-AI-Engineer-Associate □を無料でダウンロードDatabricks-Generative-AI-Engineer-Associate試験関連赤本
- 信頼できるDatabricks Databricks-Generative-AI-Engineer-Associate的中関連問題 - 合格スムーズDatabricks-Generative-AI-Engineer-Associate赤本勉強 | 有難いDatabricks-Generative-AI-Engineer-Associate復習範囲 □⇛ www.goshiken.com ⇚を入力して▸ Databricks-Generative-AI-Engineer-Associate ◂を検索し、無料でダウンロードしてくださいDatabricks-Generative-AI-Engineer-Associate日本語認定
- Databricks-Generative-AI-Engineer-Associate を効率よく取得したい人に一番お勧めしたい一冊です。 □ サイト➤ www.pass4test.jp □で➠ Databricks-Generative-AI-Engineer-Associate □問題集をダウンロードDatabricks-Generative-AI-Engineer-Associate日本語認定
- classesarefun.com, study.stcs.edu.np, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt,

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, daotao.wisebusiness.edu.vn, shortcourses.russellcollege.edu.au, Disposable vapes

2025年GoShikenの最新Databricks-Generative-AI-Engineer-Associate PDFダンプおよびDatabricks-Generative-AI-Engineer-Associate試験エンジンの無料共有：https://drive.google.com/open?id=1cTNRZBd6UPqH9D6YvcOW8kPXDbf7VJhB