

有效的考試資料Databricks-Generative-AI-Engineer-Associate參考資料保證助您輕鬆通過Databricks Databricks-Generative-AI-Engineer-Associate考試無憂



P.S. PDFExamDumps在Google Drive上分享了免費的、最新的Databricks-Generative-AI-Engineer-Associate考試題庫：<https://drive.google.com/open?id=1BPkmYHY24n4wCBayJMm7lcAOqf47JSmU>

選擇捷徑、使用技巧是為了更好地獲得成功。如果你想獲得一次就通過Databricks-Generative-AI-Engineer-Associate認證考試的保障，那麼PDFExamDumps的Databricks-Generative-AI-Engineer-Associate考古題是你唯一的、也是最好的選擇。這絕對是一個讓你禁不住讚美的考古題。你不可能找到比它更好的考試相關的資料了。這個考古題可以讓你更準確地瞭解考試的出題點，從而讓你更有目的地學習相關知識。另外，如果你實在沒有準備考試的時間，那麼你只需要記好這個考古題裏的試題和答案。因為這個考古題包括了真實考試中的所有試題，所以只是這樣你也可以通過考試。

Databricks Databricks-Generative-AI-Engineer-Associate 考試大綱：

主題	簡介
主題 1	<ul style="list-style-type: none"> Data Preparation: Generative AI Engineers covers a chunking strategy for a given document structure and model constraints. The topic also focuses on filter extraneous content in source documents. Lastly, Generative AI Engineers also learn about extracting document content from provided source data and format.
主題 2	<ul style="list-style-type: none"> Assembling and Deploying Applications: In this topic, Generative AI Engineers get knowledge about coding a chain using a pyfunc mode, coding a simple chain using langchain, and coding a simple chain according to requirements. Additionally, the topic focuses on basic elements needed to create a RAG application. Lastly, the topic addresses sub-topics about registering the model to Unity Catalog using MLflow.
主題 3	<ul style="list-style-type: none"> Application Development: In this topic, Generative AI Engineers learn about tools needed to extract data, Langchain similar tools, and assessing responses to identify common issues. Moreover, the topic includes questions about adjusting an LLM's response, LLM guardrails, and the best LLM based on the attributes of the application.

>> Databricks-Generative-AI-Engineer-Associate參考資料 <<

準確的Databricks-Generative-AI-Engineer-Associate參考資料 - 在PDFExamDumps平臺最好

Databricks的Databricks-Generative-AI-Engineer-Associate認證考試是現在IT領域非常有人氣的考試。取得這個考試的認證資格對想晉升的人們來說是最好的，也是最可以看到效果的選擇。而且，借助這個考試你也可以提高自己的技能，掌握更多的對工作有用的技術。這樣你就可以更好地完成你的工作，向別人展示你的超強的能力。也只有這樣你才可以獲得更多的发展机会。

最新的 Generative AI Engineer Databricks-Generative-AI-Engineer-Associate 免費考試真題 (Q68-Q73):

問題 #68

What is an effective method to preprocess prompts using custom code before sending them to an LLM?

- **A. Write a MLflow PyFunc model that has a separate function to process the prompts**
- B. It is better not to introduce custom code to preprocess prompts as the LLM has not been trained with examples of the preprocessed prompts
- C. Rather than preprocessing prompts, it's more effective to postprocess the LLM outputs to align the outputs to desired outcomes
- D. Directly modify the LLM's internal architecture to include preprocessing steps

答案： A

解題說明：

The most effective way to preprocess prompts using custom code is to write a custom model, such as an MLflow PyFunc model. Here's a breakdown of why this is the correct approach:

MLflow PyFunc Models:

MLflow is a widely used platform for managing the machine learning lifecycle, including experimentation, reproducibility, and deployment. A PyFunc model is a generic Python function model that can implement custom logic, which includes preprocessing prompts.

Preprocessing Prompts:

Preprocessing could include various tasks like cleaning up the user input, formatting it according to specific rules, or augmenting it with additional context before passing it to the LLM. Writing this preprocessing as part of a PyFunc model allows the custom code to be managed, tested, and deployed easily.

Modular and Reusable:

By separating the preprocessing logic into a PyFunc model, the system becomes modular, making it easier to maintain and update without needing to modify the core LLM or retrain it.

Why Other Options Are Less Suitable:

A (Modify LLM's Internal Architecture): Directly modifying the LLM's architecture is highly impractical and can disrupt the model's performance. LLMs are typically treated as black-box models for tasks like prompt processing.

B (Avoid Custom Code): While it's true that LLMs haven't been explicitly trained with preprocessed prompts, preprocessing can still improve clarity and alignment with desired input formats without confusing the model.

C (Postprocessing Outputs): While postprocessing the output can be useful, it doesn't address the need for clean and well-formatted inputs, which directly affect the quality of the model's responses.

Thus, using an MLflow PyFunc model allows for flexible and controlled preprocessing of prompts in a scalable way, making it the most effective method.

問題 #69

A Generative AI Engineer at an automotive company would like to build a question-answering chatbot to help customers answer specific questions about their vehicles. They have:

A catalog with hundreds of thousands of cars manufactured since the 1960s
Historical searches with user queries and successful matches
Descriptions of their own cars in multiple languages
They have already selected an open-source LLM and created a test set of user queries. They need to discard techniques that will not help them build the chatbot. Which do they discard?

- A. Implementing metadata filtering based on car models and years
- B. Adding few-shot examples for response generation
- C. Fine-tuning an embedding model on automotive terminology
- **D. Setting chunk size to match the model's context window to maximize coverage**

答案： D

解題說明：

According to Generative AI engineering standards for Retrieval-Augmented Generation (RAG), chunking strategy is a critical

optimization variable. Setting the chunk size to match the model's maximum context window (e.g., 4k or 8k tokens) is a poor practice and should be discarded. Large chunks introduce significant "noise" into the LLM's context, as only a small portion of a massive chunk usually contains the answer to a specific query. This leads to the "lost in the middle" phenomenon where LLMs struggle to extract relevant information from bloated contexts. Furthermore, large chunks reduce the precision of the vector search. Standard best practices involve using smaller, semantically meaningful chunks (typically 256-512 tokens) with overlap to maintain context. In contrast, metadata filtering (B) is essential for narrowing searches to specific car years, fine-tuning embeddings (C) improves retrieval accuracy for domain-specific technical terms, and few-shot examples (D) guide the LLM's output format and tone.

問題 #70

A Generative AI Engineer has created a RAG application to look up answers to questions about a series of fantasy novels that are being asked on the author's web forum. The fantasy novel texts are chunked and embedded into a vector store with metadata (page number, chapter number, book title), retrieved with the user's query, and provided to an LLM for response generation. The Generative AI Engineer used their intuition to pick the chunking strategy and associated configurations but now wants to more methodically choose the best values.

Which TWO strategies should the Generative AI Engineer take to optimize their chunking strategy and parameters? (Choose two.)

- A. Create an LLM-as-a-judge metric to evaluate how well previous questions are answered by the most appropriate chunk. Optimize the chunking parameters based upon the values of the metric.
- B. Add a classifier for user queries that predicts which book will best contain the answer. Use this to filter retrieval.
- C. Pass known questions and best answers to an LLM and instruct the LLM to provide the best token count. Use a summary statistic (mean, median, etc.) of the best token counts to choose chunk size.
- D. Choose an appropriate evaluation metric (such as recall or NDCG) and experiment with changes in the chunking strategy, such as splitting chunks by paragraphs or chapters. Choose the strategy that gives the best performance metric.
- E. Change embedding models and compare performance.

答案: A,D

解題說明:

To optimize a chunking strategy for a Retrieval-Augmented Generation (RAG) application, the Generative AI Engineer needs a structured approach to evaluating the chunking strategy, ensuring that the chosen configuration retrieves the most relevant information and leads to accurate and coherent LLM responses. Here's why C and E are the correct strategies:

Strategy C: Evaluation Metrics (Recall, NDCG)

Define an evaluation metric: Common evaluation metrics such as recall, precision, or NDCG (Normalized Discounted Cumulative Gain) measure how well the retrieved chunks match the user's query and the expected response.

Recall measures the proportion of relevant information retrieved.

NDCG is often used when you want to account for both the relevance of retrieved chunks and the ranking or order in which they are retrieved.

Experiment with chunking strategies: Adjusting chunking strategies based on text structure (e.g., splitting by paragraph, chapter, or a fixed number of tokens) allows the engineer to experiment with various ways of slicing the text. Some chunks may better align with the user's query than others.

Evaluate performance: By using recall or NDCG, the engineer can methodically test various chunking strategies to identify which one yields the highest performance. This ensures that the chunking method provides the most relevant information when embedding and retrieving data from the vector store.

Strategy E: LLM-as-a-Judge Metric

Use the LLM as an evaluator: After retrieving chunks, the LLM can be used to evaluate the quality of answers based on the chunks provided. This could be framed as a "judge" function, where the LLM compares how well a given chunk answers previous user queries.

Optimize based on the LLM's judgment: By having the LLM assess previous answers and rate their relevance and accuracy, the engineer can collect feedback on how well different chunking configurations perform in real-world scenarios.

This metric could be a qualitative judgment on how closely the retrieved information matches the user's intent.

Tune chunking parameters: Based on the LLM's judgment, the engineer can adjust the chunk size or structure to better align with the LLM's responses, optimizing retrieval for future queries.

By combining these two approaches, the engineer ensures that the chunking strategy is systematically evaluated using both quantitative (recall/NDCG) and qualitative (LLM judgment) methods. This balanced optimization process results in improved retrieval relevance and, consequently, better response generation by the LLM.

問題 #71

A Generative AI Engineer is building a system which will answer questions on latest stock news articles. Which will NOT help with ensuring the outputs are relevant to financial news?

- A. Incorporate manual reviews to correct any problematic outputs prior to sending to the users
- **B. Increase the compute to improve processing speed of questions to allow greater relevancy analysis**
- C. Implement a profanity filter to screen out offensive language
- D. Implement a comprehensive guardrail framework that includes policies for content filters tailored to the finance sector.

答案: B

解題說明:

In the context of ensuring that outputs are relevant to financial news, increasing compute power (option B) does not directly improve the relevance of the LLM-generated outputs. Here's why:

Compute Power and Relevancy:

Increasing compute power can help the model process inputs faster, but it does not inherently improve the relevance of the answers. Relevancy depends on the data sources, the retrieval method, and the filtering mechanisms in place, not on how quickly the model processes the query.

What Actually Helps with Relevancy:

Other methods, like content filtering, guardrails, or manual review, can directly impact the relevance of the model's responses by ensuring the model focuses on pertinent financial content. These methods help tailor the LLM's responses to the financial domain and avoid irrelevant or harmful outputs.

Why Other Options Are More Relevant:

A (Comprehensive Guardrail Framework): This will ensure that the model avoids generating content that is irrelevant or inappropriate in the finance sector.

C (Profanity Filter): While not directly related to financial relevancy, ensuring the output is clean and professional is still important in maintaining the quality of responses.

D (Manual Review): Incorporating human oversight to catch and correct issues with the LLM's output ensures the final answers are aligned with financial content expectations.

Thus, increasing compute power does not help with ensuring the outputs are more relevant to financial news, making option B the correct answer.

問題 #72

A Generative AI Engineer is building a Generative AI system that suggests the best matched employee team member to newly scoped projects. The team member is selected from a very large team. The match should be based upon project date availability and how well their employee profile matches the project scope. Both the employee profile and project scope are unstructured text. How should the Generative AI Engineer architect their system?

- A. Create a tool to find available team members given project dates. Create a second tool that can calculate a similarity score for a combination of team member profile and the project scope. Iterate through the team members and rank by best score to select a team member.
- B. Create a tool for finding available team members given project dates. Embed all project scopes into a vector store, perform a retrieval using team member profiles to find the best team member.
- C. Create a tool for finding team member availability given project dates, and another tool that uses an LLM to extract keywords from project scopes. Iterate through available team members' profiles and perform keyword matching to find the best available team member.
- **D. Create a tool for finding available team members given project dates. Embed team profiles into a vector store and use the project scope and filtering to perform retrieval to find the available best matched team members.**

答案: D

解題說明:

* Problem Context: The problem involves matching team members to new projects based on two main factors:

* Availability: Ensure the team members are available during the project dates.

* Profile-Project Match: Use the employee profiles (unstructured text) to find the best match for a project's scope (also unstructured text).

The two main inputs are the employee profiles and project scopes, both of which are unstructured. This means traditional rule-based systems (e.g., simple keyword matching) would be inefficient, especially when working with large datasets.

* Explanation of Options: Let's break down the provided options to understand why D is the most optimal answer.

* Option A suggests embedding project scopes into a vector store and then performing retrieval using team member profiles. While

embedding project scopes into a vector store is a valid technique, it skips an important detail: the focus should primarily be on embedding employee profiles because we're matching the profiles to a new project, not the other way around.

* Option B involves using a large language model (LLM) to extract keywords from the project scope and perform keyword matching on employee profiles. While LLMs can help with keyword extraction, this approach is too simplistic and doesn't leverage advanced retrieval techniques like vector embeddings, which can handle the nuanced and rich semantics of unstructured data. This approach may miss out on subtle but important similarities.

* Option C suggests calculating a similarity score between each team member's profile and project scope. While this is a good idea, it doesn't specify how to handle the unstructured nature of data efficiently. Iterating through each member's profile individually could be computationally expensive in large teams. It also lacks the mention of using a vector store or an efficient retrieval mechanism.

* Option D is the correct approach. Here's why:

* Embedding team profiles into a vector store: Using a vector store allows for efficient similarity searches on unstructured data.

Embedding the team member profiles into vectors captures their semantics in a way that is far more flexible than keyword-based matching.

* Using project scope for retrieval: Instead of matching keywords, this approach suggests using vector embeddings and similarity search algorithms (e.g., cosine similarity) to find the team members whose profiles most closely align with the project scope.

* Filtering based on availability: Once the best-matched candidates are retrieved based on profile similarity, filtering them by availability ensures that the system provides a practically useful result.

This method efficiently handles large-scale datasets by leveraging vector embeddings and similarity search techniques, both of which are fundamental tools in Generative AI engineering for handling unstructured text.

* Technical References:

* Vector embeddings: In this approach, the unstructured text (employee profiles and project scopes) is converted into high-dimensional vectors using pretrained models (e.g., BERT, Sentence-BERT, or custom embeddings). These embeddings capture the semantic meaning of the text, making it easier to perform similarity-based retrieval.

* Vector stores: Solutions like FAISS or Milvus allow storing and retrieving large numbers of vector embeddings quickly. This is critical when working with large teams where querying through individual profiles sequentially would be inefficient.

* LLM Integration: Large language models can assist in generating embeddings for both employee profiles and project scopes. They can also assist in fine-tuning similarity measures, ensuring that the retrieval system captures the nuances of the text data.

* Filtering: After retrieving the most similar profiles based on the project scope, filtering based on availability ensures that only team members who are free for the project are considered.

This system is scalable, efficient, and makes use of the latest techniques in Generative AI, such as vector embeddings and semantic search.

問題 #73

.....

當前 Databricks 作為企業資訊解決方案的重要性及緊要性與日俱增，相關的工作機會將會越來越多，對技術能力的要求也越來越被企業作為面試的一個標準，所以不管在哪個行業，Databricks 工作者都必須不斷自我學習、接受訓練課程或是參加各式的專業認證來充實自己，使自己在工作上可以更加得心應手。而通過了 Databricks Databricks-Generative-AI-Engineer-Associate 認證考試，證明你的 IT 專業知識很強，有很強的能力，可以勝任一份很好的工作。

Databricks-Generative-AI-Engineer-Associate 在線考題: https://www.pdfexamdumps.com/Databricks-Generative-AI-Engineer-Associate_valid-braindumps.html

- 完美的 Databricks-Generative-AI-Engineer-Associate 參考資料和資格考試的領導者和最新更新的 Databricks Databricks Certified Generative AI Engineer Associate 開啟 (www.vcesoft.com) 輸入 { Databricks-Generative-AI-Engineer-Associate } 並獲取免費下載 Databricks-Generative-AI-Engineer-Associate 測試引擎
- Databricks-Generative-AI-Engineer-Associate 考試題庫 Databricks-Generative-AI-Engineer-Associate 考試大綱 Databricks-Generative-AI-Engineer-Associate 最新考題 在 www.newdumpspdf.com 上搜索 “Databricks-Generative-AI-Engineer-Associate” 並獲取免費下載 Databricks-Generative-AI-Engineer-Associate 考試證照
- Databricks-Generative-AI-Engineer-Associate 參考資料 | 準備通過 Databricks Certified Generative AI Engineer Associate 快人一步 tw.fast2test.com 網站搜索 Databricks-Generative-AI-Engineer-Associate 並免費下載 Databricks-Generative-AI-Engineer-Associate 熱門考題
- Databricks-Generative-AI-Engineer-Associate 考古題推薦 Databricks-Generative-AI-Engineer-Associate 題庫 Databricks-Generative-AI-Engineer-Associate 考古題推薦 到 (www.newdumpspdf.com) 搜尋 Databricks-Generative-AI-Engineer-Associate 以獲取免費下載考試資料 Databricks-Generative-AI-Engineer-Associate 考試資料
- 完整包括的 Databricks-Generative-AI-Engineer-Associate 參考資料 | 第一次嘗試輕鬆學習並通過考試和一流的 Databricks-Generative-AI-Engineer-Associate: Databricks Certified Generative AI Engineer Associate 立即打開 www.vcesoft.com 並搜索 Databricks-Generative-AI-Engineer-Associate 以獲取免費下載 Databricks-Generative-AI-Engineer-Associate 最新考古題

- Databricks-Generative-AI-Engineer-Associate 信息資訊 □ Databricks-Generative-AI-Engineer-Associate 最新考古題 □ Databricks-Generative-AI-Engineer-Associate 熱門考題 □ ▷ www.newdumpspdf.com ◁ 上搜索 ⇒ Databricks-Generative-AI-Engineer-Associate ◀ 輕鬆獲取免費下載 Databricks-Generative-AI-Engineer-Associate 考試大綱
- 選擇 Databricks-Generative-AI-Engineer-Associate 參考資料表示您已通過 Databricks Certified Generative AI Engineer Associate 無憂 □ 透過 ⇒ www.vcesoft.com □ □ □ 輕鬆獲取 ⇒ Databricks-Generative-AI-Engineer-Associate □ □ □ 免費下載 Databricks-Generative-AI-Engineer-Associate 考古題推薦
- Databricks-Generative-AI-Engineer-Associate 參考資料和 Newdumpspdf - 認證考試材料的領導者和 Databricks Databricks Certified Generative AI Engineer Associate □ 透過“www.newdumpspdf.com”輕鬆獲取【Databricks-Generative-AI-Engineer-Associate】免費下載 Databricks-Generative-AI-Engineer-Associate PDF
- 完美的 Databricks-Generative-AI-Engineer-Associate 參考資料和資格考試的領導者和最新更新的 Databricks Databricks Certified Generative AI Engineer Associate □ 「www.newdumpspdf.com」上搜索 ► Databricks-Generative-AI-Engineer-Associate □ 輕鬆獲取免費下載 Databricks-Generative-AI-Engineer-Associate 題庫
- 快速下載 Databricks-Generative-AI-Engineer-Associate 參考資料 | 第一次嘗試輕鬆學習並通過考試，優秀的 Databricks Databricks Certified Generative AI Engineer Associate □ 開啟 ☀ www.newdumpspdf.com □ ☀ □ 輸入 □ Databricks-Generative-AI-Engineer-Associate □ 並獲取免費下載 Databricks-Generative-AI-Engineer-Associate 考試大綱
- Databricks-Generative-AI-Engineer-Associate 信息資訊 □ Databricks-Generative-AI-Engineer-Associate 題庫 □ Databricks-Generative-AI-Engineer-Associate 最新考題 □ ⇒ www.vcesoft.com □ 上的 ► Databricks-Generative-AI-Engineer-Associate ◀ 免費下載只需搜尋 Databricks-Generative-AI-Engineer-Associate 考古題推薦
- zayngdaw170929.webbuzzfeed.com, www.stes.tyc.edu.tw, networkbookmarks.com, socialnetworkadsinfo.com, bookmarkcitizen.com, heidinwmz825114.plpwiki.com, www.stes.tyc.edu.tw, joanlpmm565253.angelinsblog.com, www.stes.tyc.edu.tw, anitaerx159071.blogdemls.com, Disposable vapes

此外，這些 PDFExamDumps Databricks-Generative-AI-Engineer-Associate 考試題庫的部分內容現在是免費的：<https://drive.google.com/open?id=1BPkmYHY24n4wCBayJMm7lcAOqf47JSmU>