최신Databricks-Generative-AI-Engineer-Associate최신핫덤프인증덤프공부자료

참고: Itexamdump에서 Google Drive로 공유하는 무료 2026 Databricks Databricks-Generative-AI-Engineer-Associate 시험 문제집이 있습니다: https://drive.google.com/open?id=1okleKR5ACEFVNdg17BrmJrZu05lX2WbS

IT인증자격증을 취득하려고 마음먹었으면 끝까지 도전해봐야 합니다. Databricks인증 Databricks-Generative-AI-Engineer-Associate시험이 아무리 어려워도Itexamdump의Databricks인증 Databricks-Generative-AI-Engineer-Associate덤프가 동반해주면 시험이 쉬워지는 법은 많이 알려져 있습니다. Itexamdump의Databricks인증 Databricks-Generative-AI-Engineer-Associate덤프는 100% 패스보장 가능한 덤프자료입니다.한번만 믿어주시고Itexamdump제품으로 가면 시험패스는 식은 죽 먹기처럼 간단합니다.

Databricks Databricks-Generative-AI-Engineer-Associate 시험요강:

주제	소개
주제 1	• Governance: Generative AI Engineers who take the exam get knowledge about masking techniques, guardrail techniques, and legal • licensing requirements in this topic.
주제 2	• 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.
주제 3	• Evaluation and Monitoring: This topic is all about selecting an LLM choice and key metrics. Moreover, Generative AI Engineers learn about evaluating model performance. Lastly, the topic includes sub-topics about inference logging and usage of Databricks features.
주제 4	• Design Applications: The topic focuses on designing a prompt that elicits a specifically formatted response. It also focuses on selecting model tasks to accomplish a given business requirement. Lastly, the topic covers chain components for a desired model input and output.
주제 5	• 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.

>> Databricks-Generative-AI-Engineer-Associate최신핫덤프 <<

Databricks-Generative-AI-Engineer-Associate최신핫덤프 완벽한 시험 최신 버전 덤프자료 다운

Itexamdump을 선택함으로 100%인증시험을 패스하실 수 있습니다. 우리는Databricks Databricks-Generative-AI-Engineer-Associate시험의 갱신에 따라 최신의 덤프를 제공할 것입니다. Itexamdump에서는 무료로 24시간 온라인상담이 있으며, Itexamdump의 덤프로Databricks Databricks-Generative-AI-Engineer-Associate시험을 패스하지 못한다면 우리는 덤프전액환불을 약속 드립니다.

최신 Generative AI Engineer Databricks-Generative-AI-Engineer-Associate 무료샘플문제 (Q13-Q18):

질문 # 13
After changing the response generating LLM in a RAG pipeline from GPT-4 to a model with a shorter context length that the company self-hosts, the Generative AI Engineer is getting the following error:
What TWO solutions should the Generative AI Engineer implement without changing the response generating model? (Choose two.)

- A. Use a smaller embedding model to generate
- B. Retrain the response generating model using ALiBi
- C. Reduce the number of records retrieved from the vector database
- D. Reduce the maximum output tokens of the new model
- E. Decrease the chunk size of embedded documents

정답：C,E

설명：
* Problem Context: After switching to a model with a shorter context length, the error message indicating that the prompt token count has exceeded the limit suggests that the input to the model is too large.
* Explanation of Options:
* Option A: Use a smaller embedding model to generate- This wouldn't necessarily address the issue of prompt size exceeding the model's token limit.
* Option B: Reduce the maximum output tokens of the new model- This option affects the output length, not the size of the input being too large.
* Option C: Decrease the chunk size of embedded documents- This would help reduce the size of each document chunk fed into the model, ensuring that the input remains within the model's context length limitations.
* Option D: Reduce the number of records retrieved from the vector database- By retrieving fewer records, the total input size to the model can be managed more effectively, keeping it within the allowable token limits.
* Option E: Retrain the response generating model using ALiBi- Retraining the model is contrary to the stipulation not to change the response generating model.
OptionsCandDare the most effective solutions to manage the model's shorter context length without changing the model itself, by adjusting the input size both in terms of individual document size and total documents retrieved.

질문 # 14
A Generative AI Engineer at a legal firm is designing a RAG system to analyze historical legal cases. The system needs to process millions of court opinions and legal documents, already organized by time and topic, to track how interpretations of specific laws have evolved over time. All of these documents are in plain-text. The engineer needs to choose a chunking method that would most effectively preserve continuity and the temporal nature of the cases. Which method do they choose?

- A. Implement windowed summarization with overlapping chunks.
- B. Implement paragraph level embeddings with each chunk.
- C. Implement a hierarchical tree structure, like RAPTOR, to group similar legal concepts.
- D. Implement sentence level embeddings with each chunk tagged with the time to enable metadata filtering.

정답：A

설명：
In the context of legal document analysis where the "evolution of interpretation" is the primary goal, preserving narrative continuity is paramount. Windowed summarization with overlapping chunks is the most effective method for this use case. Overlapping (e.g., 10-15% of the chunk size) ensures that sentences or concepts split at the boundary of one chunk are preserved in the next, preventing the loss of critical context that often occurs in legal jargon. Furthermore, windowed summarization allows the system to condense

long-form court opinions into manageable parts while maintaining the chronological "thread" of the argument. While sentence-level embeddings with metadata (D) are useful for filtering, they often lack the sufficient context required to understand the nuances of a legal ruling. A windowed approach provides the LLM with enough surrounding text to understand the "why" behind a legal evolution, rather than just the "when."

질문 # 15
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 Al Engineer architect their system?

- A. 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.
- B. 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.
- 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 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.

정답：B

설명：
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 Reference:
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.

질문 # 16
A Generative Al Engineer is setting up a Databricks Vector Search that will lookup news articles by topic within 10 days of the date specified An example query might be "Tell me about monster truck news around January 5th 1992". They want to do this with the least amount of effort.
How can they set up their Vector Search index to support this use case?

- A. Include metadata columns for article date and topic to support metadata filtering.
- B. pass the query directly to the vector search index and return the best articles.
- C. Create separate indexes by topic and add a classifier model to appropriately pick the best index.
- D. Split articles by 10 day blocks and return the block closest to the query.

정답：A

설명:
The task is to set up a Databricks Vector Search index for news articles, supporting queries like "monster truck news around January 5th, 1992," with minimal effort. The index must filter by topic and a 10-day date range. Let's evaluate the options.
* Option A: Split articles by 10-day blocks and return the block closest to the query
* Pre-splitting articles into 10-day blocks requires significant preprocessing and index management (e.g., one index per block). It's effort-intensive and inflexible for dynamic date ranges.
* Databricks Reference:"Static partitioning increases setup complexity; metadata filtering is preferred"("Databricks Vector Search Documentation").
* Option B: Include metadata columns for article date and topic to support metadata filtering
* Adding date and topic as metadata in the Vector Search index allows dynamic filtering (e.g., date
± 5 days, topic = "monster truck") at query time. This leverages Databricks' built-in metadata filtering, minimizing setup effort.
* Databricks Reference:"Vector Search supports metadata filtering on columns like date or category for precise retrieval with minimal preprocessing"("Vector Search Guide," 2023).
* Option C: Pass the query directly to the vector search index and return the best articles
* Passing the full query (e.g., "Tell me about monster truck news around January 5th, 1992") to Vector Search relies solely on embeddings, ignoring structured filtering for date and topic. This risks inaccurate results without explicit range logic.
* Databricks Reference:"Pure vector similarity may not handle temporal or categorical constraints effectively"("Building LLM Applications with Databricks").
* Option D: Create separate indexes by topic and add a classifier model to appropriately pick the best index
* Separate indexes per topic plus a classifier model adds significant complexity (index creation, model training, maintenance), far exceeding "least effort." It's overkill for this use case.
* Databricks Reference:"Multiple indexes increase overhead; single-index with metadata is simpler"("Databricks Vector Search Documentation").
Conclusion: Option B is the simplest and most effective solution, using metadata filtering in a single Vector Search index to handle date ranges and topics, aligning with Databricks' emphasis on efficient, low-effort setups.

질문 # 17
A Generative Al Engineer is using an LLM to classify species of edible mushrooms based on text descriptions of certain features. The model is returning accurate responses in testing and the Generative Al Engineer is confident they have the correct list of possible labels, but the output frequently contains additional reasoning in the answer when the Generative Al Engineer only wants to return the label with no additional text.
Which action should they take to elicit the desired behavior from this LLM?

- A. Use zero shot prompting to instruct the model on expected output format
- B. Use zero shot chain-of-thought prompting to prevent a verbose output format
- C. Use a system prompt to instruct the model to be succinct in its answer
- D. Use few snot prompting to instruct the model on expected output format

정답：C

설명:

The LLM classifies mushroom species accurately but includes unwanted reasoning text, and the engineer wants only the label. Let's assess how to control output format effectively.
Option A: Use few shot prompting to instruct the model on expected output format Few-shot prompting provides examples (e.g., input: description, output: label). It can work but requires crafting multiple examples, which is effort-intensive and less direct than a clear instruction.
Databricks Reference: "Few-shot prompting guides LLMs via examples, effective for format control but requires careful design" ("Generative AI Cookbook").
Option B: Use zero shot prompting to instruct the model on expected output format Zero-shot prompting relies on a single instruction (e.g., "Return only the label") without examples. It's simpler than few-shot but may not consistently enforce succinctness if the LLM's default behavior is verbose.
Databricks Reference: "Zero-shot prompting can specify output but may lack precision without examples" ("Building LLM Applications with Databricks").
Option C: Use zero shot chain-of-thought prompting to prevent a verbose output format Chain-of-Thought (CoT) encourages step-by-step reasoning, which increases verbosity-opposite to the desired outcome. This contradicts the goal of label-only output.
Databricks Reference: "CoT prompting enhances reasoning but often results in detailed responses" ("Databricks Generative AI Engineer Guide").
Option D: Use a system prompt to instruct the model to be succinct in its answer A system prompt (e.g., "Respond with only the species label, no additional text") sets a global instruction for the LLM's behavior. It's direct, reusable, and effective for controlling output style across queries.
Databricks Reference: "System prompts define LLM behavior consistently, ideal for enforcing concise outputs" ("Generative AI Cookbook," 2023).
Conclusion: Option D is the most effective and straightforward action, using a system prompt to enforce succinct, label-only responses, aligning with Databricks' best practices for output control.

질문 # 18
.....

- 퍼펙트한 Databricks-Generative-AI-Engineer-Associate최신핫덤프 덤프샘플 다운로드 □ （ www.itdumpskr.com ）에서 검색만 하면「 Databricks-Generative-AI-Engineer-Associate 」를 무료로 다운로드할 수 있습니다Databricks-Generative-AI-Engineer-Associate최신 인증시험정보
- Databricks-Generative-AI-Engineer-Associate퍼펙트 덤프 최신버전 □ Databricks-Generative-AI-Engineer-Associate 인기자격증 덤프문제 □ Databricks-Generative-AI-Engineer-Associate시험문제집 □ 지금（ kr.fast2test.com ）에서《 Databricks-Generative-AI-Engineer-Associate 》를 검색하고 무료로 다운로드하세요Databricks-Generative-AI-Engineer-Associate인증시험 인기 덤프문제
- Databricks-Generative-AI-Engineer-Associate시험준비공부 □ Databricks-Generative-AI-Engineer-Associate합격보장 가능 덤프문제 □ Databricks-Generative-AI-Engineer-Associate인증시험대비 공부자료 □ 검색만 하면□ www.itdumpskr.com □에서「 Databricks-Generative-AI-Engineer-Associate 」무료 다운로드Databricks-Generative-AI-Engineer-Associate인증덤프 샘플문제
- Databricks-Generative-AI-Engineer-Associate완벽한 시험자료 □ Databricks-Generative-AI-Engineer-Associate완벽한 시험자료 □ Databricks-Generative-AI-Engineer-Associate인증시험대비 공부자료 □ 검색만 하면➡ www.passtip.net □□□에서➠ Databricks-Generative-AI-Engineer-Associate □무료 다운로드Databricks-Generative-AI-Engineer-Associate인증시험대비 공부자료
- www.dahanyijing.com, 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, bbs.t-firefly.com, 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, bbs.t-firefly.com, www.stes.tyc.edu.tw, rcmspace.com, wibki.com, 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, bbs.t-firefly.com, Disposable vapes

그리고 Itexamdump Databricks-Generative-AI-Engineer-Associate 시험 문제집의 전체 버전을 클라우드 저장소에서 다운로드할 수 있습니다: https://drive.google.com/open?id=1okleKR5ACEFVNdg17BrmJrZu05lX2WbS