

Databricks Databricks-Generative-AI-Engineer-Associate Zertifizierungsprüfung, Databricks-Generative-AI- Engineer-Associate Deutsch



Außerdem sind jetzt einige Teile dieser ZertPruefung Databricks-Generative-AI-Engineer-Associate Prüfungsfragen kostenlos erhältlich: https://drive.google.com/open?id=1uFIYKYhKLova_quffHxmzrUzHraBEuc

Die Databricks Databricks-Generative-AI-Engineer-Associate Zertifizierungsprüfung ist heutzutage in der konkurrenzfähigen IT-Branche immer beliebter geworden. Immer mehr Leute haben die Databricks Databricks-Generative-AI-Engineer-Associate Prüfung abgelegt. Aber ihre Schwierigkeit nimmt doch nicht ab. Es ist schwer, die Databricks Databricks-Generative-AI-Engineer-Associate Prüfung zu bestehen, weil sie sowieso eine autoritäre Prüfung ist, die Computerfachkenntnisse und die Fähigkeiten zur Informationstechnik prüft. Viele Leute haben viel Zeit und Energie auf die Databricks Databricks-Generative-AI-Engineer-Associate Zertifizierungsprüfung aufgewendet.

Databricks Databricks-Generative-AI-Engineer-Associate Prüfungsplan:

Thema	Einzelheiten
Thema 1	<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.
Thema 2	<ul style="list-style-type: none"> • 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.

Thema 3	<ul style="list-style-type: none"> • Governance: Generative AI Engineers who take the exam get knowledge about masking techniques, guardrail techniques, and legal • licensing requirements in this topic.
Thema 4	<ul style="list-style-type: none"> • 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.
Thema 5	<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.

>> Databricks Databricks-Generative-AI-Engineer-Associate Zertifizierungsprüfung <<

Databricks-Generative-AI-Engineer-Associate Fragen & Antworten & Databricks-Generative-AI-Engineer-Associate Studienführer & Databricks-Generative-AI-Engineer-Associate Prüfungsvorbereitung

Als ein professioneller Lieferant der IT Zertifizierungsprüfungssoftwares, bieten wir nicht nur die Produkte wie Databricks Databricks-Generative-AI-Engineer-Associate Prüfungsunterlagen, deren Qualität und Wirkung garantiert werden, sondern auch hochqualifizierter 24/7 Kundendienst. Wenn Sie neben Databricks Databricks-Generative-AI-Engineer-Associate noch Prüfungsunterlagen anderer Prüfungen suchen oder Fragen für den Kauf haben, können Sie direkt auf unserer Website online fragen. Innerhalb einem Jahr nach dem Kauf der Databricks Databricks-Generative-AI-Engineer-Associate Prüfungssoftware, geben wir Ihnen Bescheid, sobald die Databricks Databricks-Generative-AI-Engineer-Associate Prüfungsunterlagen aktualisiert haben.

Databricks Certified Generative AI Engineer Associate Databricks-Generative-AI-Engineer-Associate Prüfungsfragen mit Lösungen (Q62-Q67):

62. Frage

Generative AI Engineer at an electronics company just deployed a RAG application for customers to ask questions about products that the company carries. However, they received feedback that the RAG response often returns information about an irrelevant product.

What can the engineer do to improve the relevance of the RAG's response?

- A. Implement caching for frequently asked questions
- B. Use a different LLM to improve the generated response
- C. Assess the quality of the retrieved context
- D. Use a different semantic similarity search algorithm

Antwort: C

Begründung:

In a Retrieval-Augmented Generation (RAG) system, the key to providing relevant responses lies in the quality of the retrieved context. Here's why option A is the most appropriate solution:

* Context Relevance: The RAG model generates answers based on retrieved documents or context. If the retrieved information is about an irrelevant product, it suggests that the retrieval step is failing to select the right context. The Generative AI Engineer must first assess the quality of what is being retrieved and ensure it is pertinent to the query.

* Vector Search and Embedding Similarity: RAG typically uses vector search for retrieval, where embeddings of the query are matched against embeddings of product descriptions. Assessing the semantic similarity search process ensures that the closest matches are actually relevant to the query.

* Fine-tuning the Retrieval Process: By improving the retrieval quality, such as tuning the embeddings or adjusting the retrieval strategy, the system can return more accurate and relevant product information.

* Why Other Options Are Less Suitable:

* B (Caching FAQs): Caching can speed up responses for frequently asked questions but won't improve the relevance of the retrieved content for less frequent or new queries.

* C (Use a Different LLM): Changing the LLM only affects the generation step, not the retrieval process, which is the core issue

here.

* D (Different Semantic Search Algorithm): This could help, but the first step is to evaluate the current retrieval context before replacing the search algorithm.

Therefore, improving and assessing the quality of the retrieved context (option A) is the first step to fixing the issue of irrelevant product information.

63. Frage

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

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

Antwort: D

Begründung:

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.

64. Frage

A Generative AI Engineer interfaces with an LLM with prompt/response behavior that has been trained on customer calls inquiring about product availability. The LLM is designed to output "In Stock" if the product is available or only the term "Out of Stock" if not. Which prompt will work to allow the engineer to respond to call classification labels correctly?

- **A. You will be given a customer call transcript where the customer asks about product availability. The outputs are either "In Stock" or "Out of Stock". Format the output in JSON, for example: {"call_id": "123", "label": "In Stock"}.**
- B. You will be given a customer call transcript where the customer inquires about product availability. Respond with "In Stock" if the product is available or "Out of Stock" if not.
- C. Respond with "Out of Stock" if the customer asks for a product.
- D. Respond with "In Stock" if the customer asks for a product.

Antwort: A

Begründung:

* **Problem Context:** The Generative AI Engineer needs a prompt that will enable an LLM trained on customer call transcripts to classify and respond correctly regarding product availability. The desired response should clearly indicate whether a product is "In Stock" or "Out of Stock," and it should be formatted in a way that is structured and easy to parse programmatically, such as JSON.

* **Explanation of Options:**

* Option A: Respond with "In Stock" if the customer asks for a product. This prompt is too generic and does not specify how to handle the case when a product is not available, nor does it provide a structured output format.

* Option B: This option is correctly formatted and explicit. It instructs the LLM to respond based on the availability mentioned in the customer call transcript and to format the response in JSON.

This structure allows for easy integration into systems that may need to process this information automatically, such as customer service dashboards or databases.

* Option C: Respond with "Out of Stock" if the customer asks for a product. Like option A, this prompt is also insufficient as it only covers the scenario where a product is unavailable and does not provide a structured output.

* Option D: While this prompt correctly specifies how to respond based on product availability, it lacks the structured output format, making it less suitable for systems that require formatted data for further processing.

Given the requirements for clear, programmatically usable outputs, Option B is the optimal choice because it provides precise instructions on how to respond and includes a JSON format example for structuring the output, which is ideal for automated systems or further data handling.

65. Frage

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=OpenAI()
)

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

```
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(prompt=prompt)

```

- C. `llm.generate("funny")`

```

prompt_template = "Tell me a {adjective} joke"

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

llm = LLMChain(llm=OpenAI(), prompt=prompt)

```
- D. `llm.generate(prompt_variables={"adjective": "funny"})`

Antwort: A

Begründung:

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.

66. Frage

A Generative AI Engineer has developed an LLM application to answer questions about internal company policies. The Generative AI Engineer must ensure that the application doesn't hallucinate or leak confidential data.

Which approach should NOT be used to mitigate hallucination or confidential data leakage?

- **A. Fine-tune the model on your data, hoping it will learn what is appropriate and not**
- B. Limit the data available based on the user's access level
- C. Add guardrails to filter outputs from the LLM before it is shown to the user
- D. Use a strong system prompt to ensure the model aligns with your needs.

Antwort: A

Begründung:

When addressing concerns of hallucination and data leakage in an LLM application for internal company policies, fine-tuning the model on internal data with the hope it learns data boundaries can be problematic:

- * **Risk of Data Leakage:** Fine-tuning on sensitive or confidential data does not guarantee that the model will not inadvertently include or reference this data in its outputs. There's a risk of overfitting to the specific data details, which might lead to unintended leakage.
- * **Hallucination:** Fine-tuning does not necessarily mitigate the model's tendency to hallucinate; in fact, it might exacerbate it if the training data is not comprehensive or representative of all potential queries.

Better Approaches:

- * A, C, and D involve setting up operational safeguards and constraints that directly address data leakage and ensure responses are

aligned with specific user needs and security levels.

Fine-tuning lacks the targeted control needed for such sensitive applications and can introduce new risks, making it an unsuitable approach in this context.

67. Frage

.....

Nach den Forschungen in den letzten Jahren sind die Fragen und Antworten zur Databricks Databricks-Generative-AI-Engineer-Associate Zertifizierungsprüfung von ZertPruefung den realen Prüfung sehr ähnlich. ZertPruefung verspricht, dass Sie zum ersten Mal die Databricks Databricks-Generative-AI-Engineer-Associate (Databricks Certified Generative AI Engineer Associate) Zertifizierungsprüfung 100% bestehen können.

Databricks-Generative-AI-Engineer-Associate Deutsch: https://www.zertpruefung.ch/Databricks-Generative-AI-Engineer-Associate_exam.html

- Databricks-Generative-AI-Engineer-Associate Kostenlos Downladen □ Databricks-Generative-AI-Engineer-Associate Zertifikatsdemo □ Databricks-Generative-AI-Engineer-Associate Kostenlos Downladen □ Suchen Sie auf der Webseite “ www.zertfragen.com ” nach □ Databricks-Generative-AI-Engineer-Associate □ und laden Sie es kostenlos herunter □ □Databricks-Generative-AI-Engineer-Associate Prüfungsübungen
- Die seit kurzem aktuellsten Databricks Certified Generative AI Engineer Associate Prüfungsunterlagen, 100% Garantie für Ihren Erfolg in der Databricks Databricks-Generative-AI-Engineer-Associate Prüfungen! □ Suchen Sie jetzt auf “ www.itzert.com ” nach ➔ Databricks-Generative-AI-Engineer-Associate □ und laden Sie es kostenlos herunter □ □Databricks-Generative-AI-Engineer-Associate Musterprüfungsfragen
- Databricks-Generative-AI-Engineer-Associate Zertifikatsfragen □ Databricks-Generative-AI-Engineer-Associate Kostenlos Downladen □ Databricks-Generative-AI-Engineer-Associate Musterprüfungsfragen □ Suchen Sie auf 《 www.zertpruefung.ch 》 nach [Databricks-Generative-AI-Engineer-Associate] und erhalten Sie den kostenlosen Download mühelos □Databricks-Generative-AI-Engineer-Associate Lernressourcen
- Databricks Databricks-Generative-AI-Engineer-Associate: Databricks Certified Generative AI Engineer Associate braindumps PDF - Testking echter Test □ Öffnen Sie die Website 《 www.itzert.com 》 Suchen Sie ➔ Databricks-Generative-AI-Engineer-Associate □□□ Kostenloser Download □Databricks-Generative-AI-Engineer-Associate Lernressourcen
- Databricks-Generative-AI-Engineer-Associate Prüfungsaufgaben □ Databricks-Generative-AI-Engineer-Associate Testking □ Databricks-Generative-AI-Engineer-Associate Prüfungsübungen □ Öffnen Sie ✓ www.deutschpruefung.com □✓□ geben Sie (Databricks-Generative-AI-Engineer-Associate) ein und erhalten Sie den kostenlosen Download ☀️Databricks-Generative-AI-Engineer-Associate Deutsch Prüfungsfragen
- Databricks-Generative-AI-Engineer-Associate: Databricks Certified Generative AI Engineer Associate Dumps - PassGuide Databricks-Generative-AI-Engineer-Associate Examen □ Suchen Sie einfach auf “ www.itzert.com ” nach kostenloser Download von ➔ Databricks-Generative-AI-Engineer-Associate □ □Databricks-Generative-AI-Engineer-Associate Zertifikatsfragen
- Databricks-Generative-AI-Engineer-Associate Deutsche Prüfungsfragen □ Databricks-Generative-AI-Engineer-Associate Lernressourcen □ Databricks-Generative-AI-Engineer-Associate Prüfungsübungen □ Suchen Sie auf ☀️ www.zertfragen.com □☀️□ nach kostenlosem Download von □ Databricks-Generative-AI-Engineer-Associate □ □ □Databricks-Generative-AI-Engineer-Associate Prüfungsaufgaben
- Databricks-Generative-AI-Engineer-Associate Zertifizierungsfragen, Databricks Databricks-Generative-AI-Engineer-Associate PrüfungFragen □ Suchen Sie jetzt auf ☀️ www.itzert.com □☀️□ nach ▷ Databricks-Generative-AI-Engineer-Associate ◁ und laden Sie es kostenlos herunter □Databricks-Generative-AI-Engineer-Associate Testking
- Das neueste Databricks-Generative-AI-Engineer-Associate, nützliche und praktische Databricks-Generative-AI-Engineer-Associate pass4sure Trainingsmaterial □ Öffnen Sie die Webseite ⇒ www.zertpruefung.ch ⇐ und suchen Sie nach kostenloser Download von 【 Databricks-Generative-AI-Engineer-Associate 】 □Databricks-Generative-AI-Engineer-Associate Testantworten
- Databricks-Generative-AI-Engineer-Associate Testantworten □ Databricks-Generative-AI-Engineer-Associate Testking □ Databricks-Generative-AI-Engineer-Associate Testantworten □ Suchen Sie auf ▶ www.itzert.com ◀ nach ➡ Databricks-Generative-AI-Engineer-Associate □ und erhalten Sie den kostenlosen Download mühelos 📖 Databricks-Generative-AI-Engineer-Associate Prüfungs-Guide
- Databricks-Generative-AI-Engineer-Associate Databricks Certified Generative AI Engineer Associate neueste Studie Torrent - Databricks-Generative-AI-Engineer-Associate tatsächliche prep Prüfung □ ▶ www.zertpruefung.ch ◀ ist die beste Webseite um den kostenlosen Download von (Databricks-Generative-AI-Engineer-Associate) zu erhalten □Databricks-Generative-AI-Engineer-Associate Kostenlos Downladen
- berthaysdk764188.mycoolwiki.com, bookmarklinking.com, henriqgnb907771.59bloggers.com, safalrf166212.tdlwiki.com, bookmark-dofollow.com, esocialmall.com, easiestbookmarks.com, nevedvf590537.wikimillions.com, worldsocialindex.com,

vinnyhdqj050387.wiki-cms.com, Disposable vapes

Außerdem sind jetzt einige Teile dieser ZertPruefung Databricks-Generative-AI-Engineer-Associate Prüfungsfragen kostenlos erhältlich: https://drive.google.com/open?id=1uFIYKYhKLova_quffHxmzrUzHraBEuc