

Databricks-Generative-AI-Engineer-Associate最新試験情報 & Databricks-Generative-AI-Engineer-Associateテストトレーニング



P.S. JPNTTestがGoogle Driveで共有している無料かつ新しいDatabricks-Generative-AI-Engineer-Associateダンブ：<https://drive.google.com/open?id=177oZUwpEWLwt62R9GWHM23mGyNwiRFmh>

我々はDatabricks Databricks-Generative-AI-Engineer-Associate問題集をリリースされる以来、たくさんの好評を博しました。試験に合格したお客様は「Databricks-Generative-AI-Engineer-Associate問題集のオンライン版を利用して、模擬試験を繰り返して受けました。無事試験に合格しました。JPNTTestから大変助かりました。」と感謝します。あなたの支持こそ我々は最も高品質のDatabricks Databricks-Generative-AI-Engineer-Associate問題集を開発して努力します。

1年以内にDatabricks-Generative-AI-Engineer-Associateテスト準備を更新し、必要なものを無料でダウンロードします。1年後、購入者がサービスの保証を延長してお金を節約できるようにしたい場合、Databricksクライアントに50%の割引特典を提供します。あなたが古いクライアントである場合、Databricks-Generative-AI-Engineer-Associate試験トレントを購入する際に特定の割引を享受できるため、より多くのサービスとより多くのメリットを享受できます。このアップデートでは、最新かつ最も有用なDatabricks-Generative-AI-Engineer-Associate準備トレントを提供できます。さらに学習して、Databricks Certified Generative AI Engineer AssociateのDatabricks-Generative-AI-Engineer-Associate試験に合格することができます。

>> Databricks-Generative-AI-Engineer-Associate最新試験情報 <<

Databricks Databricks-Generative-AI-Engineer-Associateテストトレーニング & Databricks-Generative-AI-Engineer-Associate技術内容

話と行動の距離はどのぐらいありますか。これは人の心によることです。意志が強い人にとって、行動は目と鼻の先にあるのです。あなたはきっとこのような人でしょう。DatabricksのDatabricks-Generative-AI-Engineer-Associate認定試験に申し込んだ以上、試験に合格しなければならないです。これもあなたの意志が強いことを表示する方法です。JPNTTestが提供したトレーニング資料はインターネットで最高のものです。Databricks

の Databricks-Generative-AI-Engineer-Associate 認定試験に合格したいのなら、JPNTest の Databricks の Databricks-Generative-AI-Engineer-Associate 試験 トレーニング 資料を利用してください。

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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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。
トピック 3	<ul style="list-style-type: none"> アプリケーションの設計: このトピックでは、特定の形式の応答を引き出すプロンプトの設計に焦点を当てています。また、特定のビジネス要件を達成するためのモデル タスクの選択にも焦点を当てています。最後に、このトピックでは、必要なモデル入力と出力のチェーン コンポーネントについて説明します。
トピック 4	<ul style="list-style-type: none"> 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 5	<ul style="list-style-type: none"> 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。

Databricks Certified Generative AI Engineer Associate 認定 Databricks-Generative-AI-Engineer-Associate 試験問題 (Q18-Q23):

質問 # 18

A Generative AI Engineer is tasked with deploying an application that takes advantage of a custom MLflow Pyfunc model to return some interim results.

How should they configure the endpoint to pass the secrets and credentials?

- A. Pass variables using the Databricks Feature Store API
- B. Use `spark.conf.set()`
- C. Add credentials using environment variables
- D. Pass the secrets in plain text

正解: C

解説:

Context: Deploying an application that uses an MLflow Pyfunc model involves managing sensitive information such as secrets and credentials securely.

Explanation of Options:

* Option A: Use `spark.conf.set()`: While this method can pass configurations within Spark jobs, using it for secrets is not recommended because it may expose them in logs or Spark UI.

* Option B: Pass variables using the Databricks Feature Store API: The Feature Store API is designed for managing features for machine learning, not for handling secrets or credentials.

* Option C: Add credentials using environment variables: This is a common practice for managing credentials in a secure manner, as

environment variables can be accessed securely by applications without exposing them in the codebase.

* Option D: Pass the secrets in plain text: This is highly insecure and not recommended, as it exposes sensitive information directly in the code.

Therefore, Option C is the best method for securely passing secrets and credentials to an application, protecting them from exposure.

質問 # 19

A Generative AI Engineer is deciding between using LSH (Locality Sensitive Hashing) and HNSW (Hierarchical Navigable Small World) for indexing their vector database. Their top priority is semantic accuracy. Which approach should the Generative AI Engineer use to evaluate these two techniques?

- A. Compare the cosine similarities of the embeddings of returned results against those of a representative sample of test inputs
- B. Compare the Recall-Oriented-Understudy for Gisting Evaluation (ROUGE) scores of returned results for a representative sample of test inputs
- C. Compare the Bilingual Evaluation Understudy (BLEU) scores of returned results for a representative sample of test inputs
- D. Compare the Levenshtein distances of returned results against a representative sample of test inputs

正解: A

質問 # 20

A Generative AI Engineer is designing a chatbot for a gaming company that aims to engage users on its platform while its users play online video games.

Which metric would help them increase user engagement and retention for their platform?

- A. Lack of relevance
- B. Randomness
- C. Diversity of responses
- D. Repetition of responses

正解: C

解説:

In the context of designing a chatbot to engage users on a gaming platform, diversity of responses (option B) is a key metric to increase user engagement and retention. Here's why:

Diverse and Engaging Interactions:

A chatbot that provides varied and interesting responses will keep users engaged, especially in an interactive environment like a gaming platform. Gamers typically enjoy dynamic and evolving conversations, and diversity of responses helps prevent monotony, encouraging users to interact more frequently with the bot.

Increasing Retention:

By offering different types of responses to similar queries, the chatbot can create a sense of novelty and excitement, which enhances the user's experience and makes them more likely to return to the platform.

Why Other Options Are Less Effective:

A (Randomness): Random responses can be confusing or irrelevant, leading to frustration and reducing engagement.

C (Lack of Relevance): If responses are not relevant to the user's queries, this will degrade the user experience and lead to disengagement.

D (Repetition of Responses): Repetitive responses can quickly bore users, making the chatbot feel uninteresting and reducing the likelihood of continued interaction.

Thus, diversity of responses (option B) is the most effective way to keep users engaged and retain them on the platform.

質問 # 21

An AI developer team wants to fine-tune an open-weight model to have exceptional performance on a code generation use case.

They are trying to choose the best model to start with. They want to minimize model hosting costs and are using Hugging Face model cards and spaces to explore models. Which TWO model attributes and metrics should the team focus on to make their selection?

- A. Number of model downloads last month
- B. MTEB Leaderboard
- C. Chatbot Arena Leaderboard
- D. Big Code Models Leaderboard

- E. Number of model parameters

正解: D、E

解説:

To optimize for code generation performance and hosting costs, a Generative AI engineer must look at specific metrics. Big Code Models Leaderboard (A): This is the industry-standard benchmark for code-specific LLMs (like StarCoder or CodeLlama). It measures performance on tasks like HumanEval and MBPP, providing a direct indicator of how well the model handles programming logic.

Number of model parameters (B): This is the primary driver of hosting costs. Larger models (e.g., 70B) require more GPU memory (VRAM) and more expensive compute instances (like A100s/H100s) than smaller models (e.g., 7B or 13B). To minimize costs, the team should look for the smallest model that achieves a high score on the Big Code Leaderboard.

Note: MTEB (C) is for embeddings, and Chatbot Arena (D) is for general-purpose chat, neither of which is the primary metric for specialized code generation fine-tuning.

質問 # 22

A Generative AI Engineer is building a production-ready LLM system which replies directly to customers.

The solution makes use of the Foundation Model API via provisioned throughput. They are concerned that the LLM could potentially respond in a toxic or otherwise unsafe way. They also wish to perform this with the least amount of effort.

Which approach will do this?

- A. Ask users to report unsafe responses
- B. Add some LLM calls to their chain to detect unsafe content before returning text
- C. Host Llama Guard on Foundation Model API and use it to detect unsafe responses
- D. Add a regex expression on inputs and outputs to detect unsafe responses.

正解: C

解説:

The task is to prevent toxic or unsafe responses in an LLM system using the Foundation Model API with minimal effort. Let's assess the options.

* Option A: Host Llama Guard on Foundation Model API and use it to detect unsafe responses

* Llama Guard is a safety-focused model designed to detect toxic or unsafe content. Hosting it via the Foundation Model API (a Databricks service) integrates seamlessly with the existing system, requiring minimal setup (just deployment and a check step), and leverages provisioned throughput for performance.

* Databricks Reference: "Foundation Model API supports hosting safety models like Llama Guard to filter outputs efficiently" ("Foundation Model API Documentation," 2023).

* Option B: Add some LLM calls to their chain to detect unsafe content before returning text

* Using additional LLM calls (e.g., prompting an LLM to classify toxicity) increases latency, complexity, and effort (crafting prompts, chaining logic), and lacks the specificity of a dedicated safety model.

* Databricks Reference: "Ad-hoc LLM checks are less efficient than purpose-built safety solutions" ("Building LLM Applications with Databricks").

* Option C: Add a regex expression on inputs and outputs to detect unsafe responses

* Regex can catch simple patterns (e.g., profanity) but fails for nuanced toxicity (e.g., sarcasm, context-dependent harm), requiring significant manual effort to maintain and update rules.

* Databricks Reference: "Regex-based filtering is limited for complex safety needs" ("Generative AI Cookbook").

* Option D: Ask users to report unsafe responses

* User reporting is reactive, not preventive, and places burden on users rather than the system. It doesn't limit unsafe outputs proactively and requires additional effort for feedback handling.

* Databricks Reference: "Proactive guardrails are preferred over user-driven monitoring" ("Databricks Generative AI Engineer Guide").

Conclusion: Option A (Llama Guard on Foundation Model API) is the least-effort, most effective approach, leveraging Databricks' infrastructure for seamless safety integration.

質問 # 23

.....

多くのサイトの中で、どこかのDatabricksのDatabricks-Generative-AI-Engineer-Associate試験問題集は最も正確性が高いですか。無論JPNTestのDatabricksのDatabricks-Generative-AI-Engineer-Associate問題集が一番頼りになります。

