databricks

Databricks 試験ガイド

Databricks 認定生成 AI エンジニア アソシエイト

試験ガイドについてフィードバックを送る

この試験ガイドの目的

この試験ガイドでは、試験の準備に役立てていただくために試験の概要と試験の対象範囲について説明します。試験内容が変更された場合 (およびこれらの変更が試験に反映される際) には、本書の内容も更新されますので、それに合わせて準備を行ってください。このバージョンは、**2024**年**6**月**1**日時点の現在のライブ バージョンを対象としています。試験を受ける**2**週間前に、ご自身の試験が最新版であることを再度ご確認ください。

対象者についての説明

Databricks 認定生成 AI エンジニア アソシエイト認定試験では、Databricks を使用して LLM 対応ソリューションを設計および実装する個人の能力を評価します。これには、複雑な要件を管理可能なタスクに分割するための問題の分解や、包括的なソリューションを開発するための現在の生成 AI ランドスケープから適切なモデル、ツール、アプローチを選択することが含まれます。また、意味的類似性検索のためのベクトル検索や、モデルとソリューションのデプロイメントのためのモデルサービング、ソリューション ライフサイクル管理のための MLflow、データ ガバナンスのための Unity Catalog など、Databricks 固有のツールも評価します。この試験に合格した個人は、Databricks とそのツールセットを最大限に活用した、パフォーマンスの高い RAG アプリケーションと LLM チェーンを構築してデプロイできることが期待されます。

この試験について

- 問題数: 45 問の多肢選択問題または複数選択問題
- 制限時間: 90 分
- 受験料: $200
- 実施方法: オンライン（監督付き）
- 試験への資料の持ち込み: 不可
- 前提条件: なし。Databricks 関連コースを受講し、6 か月の実務経験を得てから受験することが強く推奨されます。また、このドキュメントの「推奨される準備」も参照してください。
- 有効期間: 2 年間。
- 再認定: 認定資格を維持するには、2 年ごとに再認定を受ける必要があります。再認定を受けるには、その時点で実施されている完全な試験を受ける必要があります。試験のウェブページの「試験の準備」セクションを確認して、再度試験を受ける準備をしてください。

推奨される準備

- 特に「Databricks で生成 AI エンジニアリング」など、生成 AI の学習者としての役割に関連する現在のすべての Databricks アカデミー コース
- 現在の LLM とその機能に関する知識
- プロンプト エンジニアリング、プロンプト生成、評価に関する知識
- LangChain、Hugging Face Transformers など、関連する現在のオンライン ツールやサービスに関する知識
- RAG アプリケーションと LLM チェーンの開発をサポートする Python とそのライブラリに関する実用的な知識
- データ準備、モデル チェーンなどの現在の API に関する実用的な知識
- 関連する Databricks ドキュメント リソース

試験の概要

セクション **1:** アプリケーションの設計

- 特定の形式の応答を引き出すプロンプトを設計する
- 特定のビジネス要件を満たすためのモデル タスクを選択する
- 目的のモデルの入力と出力のためのチェーン コンポーネントを選択する
- ビジネス ユースケースの目標に基づき、AI パイプラインに期待される入力と出力を記述する
- 多段階推論のための知識の収集やアクションの実行を行うツールを定義して順序付けする

セクション **2:** データの準備

- 特定のドキュメント構造とモデルの制約に対して、チャンク化方法を適用する
- RAG アプリケーションの品質を低下させるソース ドキュメント内の無関係なコンテンツをフィルタリングする
- 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ための適切な Python パッケージを選択する
- チャンク化されたテキストを Unity Catalog の Delta Lake テーブルに書き込むための操作とシーケンスを定義する
- 特定の RAG アプリケーションに必要な知識と品質を提供するソース ドキュメントを特定する
- 特定のモデル タスクに合致するプロンプトと応答のペアを特定する
- ツールとメトリクスを使用して検索パフォーマンスを評価する

セクション **3:** アプリケーション開発

- データ取得の特定のニーズに対応するデータの抽出に必要なツールを作成する
- 生成 AI アプリケーションで使用する LangChain または類似のツールを選択する
- プロンプト形式によってモデルの出力と結果がどのように変わるかを特定する
- 応答を定性的に評価して、品質や安全性といった共通の問題を特定する
- モデルと検索の評価に基づいてチャンク化方法を選択する

- 主要なフィールド、用語、意図に基づいて、ユーザー入力から追加コンテキストによってプロンプトを拡張する
- LLM の応答をベースラインから目的の出力に調整するプロンプトを作成する
- LLM ガードレールを実装してネガティブな結果を回避する
- ハルシネーションや非公開データの漏洩を最小限に抑えるメタプロンプトを記述する
- 使用可能な関数を公開するエージェント プロンプト テンプレートを作成する
- 開発するアプリケーションの属性に基づいて最適な LLM を選択する
- ソース ドキュメント、予想されるクエリー、最適化戦略に基づいてエンベディング モデルのコンテキスト長を選択する
- モデルのメタデータ/モデル カードに基づいて、モデル ハブまたは Marketplace からタスクのためのモデルを選択する
- エクスペリメントで生成された共通のメトリクスに基づいて、特定のタスクのための最適なモデルを選択する

セクション **4:** アプリケーションのアンサンブルとデプロイ

- 前処理と後処理で pyfunc モデルを使用してチェーンをコード化する
- モデル サービング エンドポイントからリソースへのアクセスを制御する
- 要件に沿ってシンプルなチェーンをコード化する
- LangChain を使用してシンプルなチェーンをコード化する
- RAG アプリケーションの構築に必要な基本要素 (モデル フレーバー、埋め込みモデル、リトリーバー、依存関係、入力例、モデル シグネチャ) を選択する
- MLflow を使用してモデルを Unity Catalog に登録する
- 基本的な RAG アプリケーションのエンドポイントをデプロイするために必要なステップを順序付ける
- ベクトル検索インデックスを作成してクエリーを実行する
- 基盤モデル API を活用する LLM アプリケーションの提供方法を特定する
- RAG アプリケーションの機能を提供するために必要なリソースを特定する

セクション **5:** ガバナンス

- マスキング手法をガードレールとして使用して、パフォーマンスの目標を達成する
- 生成 AI アプリケーションへの悪意のあるユーザー入力から保護するためのガードレール手法を選択する
- RAG アプリケーションにフィードするデータ ソースで問題のあるテキストを軽減するための代替手段を推奨する
- データ ソースの法的/ライセンス要件を使用して法的リスクを回避する

セクション **6:** 評価とモニタリング

- 一連の定量的評価メトリクスに基づいて LLM のオプション (サイズとアーキテクチャ) を選択する
- 特定の LLM デプロイメント シナリオをモニタリングするための主要なメトリクスを選択する
- MLflow を使用して RAG アプリケーションでのモデルのパフォーマンスを評価する
- 推論ロギングを使用して、デプロイ済みの RAG アプリケーションのパフォーマンスを評価する
- Databricks の機能を使用して RAG アプリケーションの LLM コストを制御する

サンプル問題

これらの質問は、受験者がこの試験で出題される質問の全般的なスタイルや形式を理解できるよう、実際の問題に似せて作成されています。試験ガイドに記載されている試験の目的が含まれており、目的に沿ったサンプル問題も提供されます。この試験ガイドには、試験の出題対象であるすべての目的が記載されています。認定試験の準備を行う際は、この試験ガイドの「試験の概要」を確認することをお勧めします。

問題 **1**

目的: 特定のドキュメント構造とモデルの制約にチャンク化方法を適用する

ある生成 AI エンジニアが、最大 1 億個のレコードを格納可能なベクター データベースに 1億 5,000 万個のエンベディングを読み込もうとしています。

レコード数を減らすために実行できる 2 つのアクションはどれですか?

A. ドキュメントのチャンク サイズを増やす
B. チャンク間のオーバーラップを減らす
C. ドキュメントのチャンク サイズを小さくする
D. チャンク間のオーバーラップを増やす
E. より小さな埋め込みモデルを使用する

問題 **2**

目的: 特定の *RAG* アプリケーションに必要な知識と品質を提供するソース ドキュメントを特定する。

ある生成 AI エンジニアが、自動車部品の販売支援に向けて開発中の、顧客向け生成 AI アプリケーションからの応答を評価しています。このアプリケーションでは、質問に回答するために、顧客に account_id と transaction_id を明示的に入力してもらう必要があります。最初のリリース後、顧客からのフィードバックは、アプリケーションは注文と請求の詳細に適切に回答したが、出荷と到着予定日の質問に正確に回答できなかったというものでした。

次のレシーバーのうち、これらの質問に回答するアプリケーションの機能向上につながるものはどれですか?

A. すべての自動車部品の会社の配送ポリシーと支払い条件を含むベクトル ストアを作成する
B. 請求データと配送予定日とともに入力されるプライマリ キーとして transaction_id を使って Feature Store テーブルを作成する
C. 到着予定日のサンプル データをチューニング データセットとして提供し、定期的にモデルをファインチューニングすることで最新の出荷情報を維持する

D. チャット プロンプトを修正して注文日時を入力し、それに 14 日を追加するようモデルに指示する (どんな配送方法でも 14 日を超えないと想定されるため)

問題 **3**

目的: 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ための適切な *Python* パッケージを選択する。

ある生成 AI エンジニアが、.jpeg や .png などの画像形式のファイルとして保存された、スキャン済みのソース ドキュメントから取得したコンテキストに依存する RAG アプリケーションを構築しています。この生成 AI エンジニアは、最小限のコード行数でこのソリューションを開発したいと考えています。

これらのソース ドキュメントからテキストを抽出するために使用すべき Python パッケージはどれですか?

A. BeautifulSoup
B. scrapy
C. pytesseract
D. pyquery

問題 **4**

目的: ソース ドキュメント、予想されるクエリー、最適化戦略に基づいてエンベディング モデルのコンテキスト長を選択する

ある生成 AI エンジニアが LLM ベースのアプリケーションを構築しています。そのリトリーバーのドキュメントは、それぞれ最大 512 個のトークンにチャンク化されています。この生成 AI エンジニアは、このアプリケーションでは品質よりもコストとレイテンシーのほうが重要であることを理解しています。選択肢としていくつかのコンテキスト長レベルがあります。

このニーズを満たすものはどれですか?

A. コンテキスト長 512: 最小モデルは 0.13GB、エンベディングの次元は 384
B. コンテキスト長 514: 最小モデルは 0.44GB、エンベディングの次元は 768
C. コンテキスト長 2048: 最小モデルは 11GB、エンベディングの次元は 2560
D. コンテキスト長 32768: 最小モデルは 14GB、エンベディングの次元は 4096

問題 **5**

目的: 開発するアプリケーションの属性に基づいて最適な *LLM* を選択する

ある生成 AI エンジニアが、約 1 段落分のメモ フィールドをその意図を示す 1 文程度の要約に更新でき、さらにそのアプリケーションのフロントエンドに適合するアプリケーションを構築したいと考えています。

このアプリケーションの潜在的な LLM を評価するには、どの自然言語処理タスク カテゴリーを使用すべきですか?

A. text2textの生成
B. センテンサイザー
C. テキスト分類
D. 要約

解答
問題 1: A、B
問題 2: B
問題 3: C
問題 4: A
問題 5: D