Free Questions for Databricks-Generative-AI-Engineer-Associate

Shared by Merrill on 04-10-2024

Question 1

Question Type: **MultipleChoice**

A Generative AI Engineer developed an LLM application using the provisioned throughput Foundation Model API. Now that the application is ready to be deployed, they realize their volume of requests are not sufficiently high enough to create their own provisioned throughput endpoint. They want to choose a strategy that ensures the best cost-effectiveness for their application.

What strategy should the Generative AI Engineer use?

Options:

A- Switch to using External Models instead

B- Deploy the model using pay-per-token throughput as it comes with cost guarantees

C- Change to a model with a fewer number of parameters in order to reduce hardware constraint issues

D- Throttle the incoming batch of requests manually to avoid rate limiting issues

Answer:

B

Explanation:

Problem Context: The engineer needs a cost-effective deployment strategy for an LLM application with relatively low request volume.

Explanation of Options:

Option A: Switching to external models may not provide the required control or integration necessary for specific application needs.

Option B: Using a pay-per-token model is cost-effective, especially for applications with variable or low request volumes, as it aligns costs directly with usage.

Option C: Changing to a model with fewer parameters could reduce costs, but might also impact the performance and capabilities of the application.

Option D: Manually throttling requests is a less efficient and potentially error-prone strategy for managing costs.

Option B is ideal, offering flexibility and cost control, aligning expenses directly with the application's usage patterns.

Question 2

Question Type: MultipleChoice

A Generative AI Engineer has already trained an LLM on Databricks and it is now ready to be deployed.

Which of the following steps correctly outlines the easiest process for deploying a model on Databricks?

Options:

A- Log the model as a pickle object, upload the object to Unity Catalog Volume, register it to Unity Catalog using MLflow, and start a serving endpoint

B- Log the model using MLflow during training, directly register the model to Unity Catalog using the MLflow API, and start a serving endpoint

C- Save the model along with its dependencies in a local directory, build the Docker image, and run the Docker container

D- Wrap the LLM's prediction function into a Flask application and serve using Gunicorn

Answer:

B

Explanation:

Problem Context: The goal is to deploy a trained LLM on Databricks in the simplest and most integrated manner.

Explanation of Options:

Option A: This method involves unnecessary steps like logging the model as a pickle object, which is not the most efficient path in a Databricks environment.

Option B: Logging the model with MLflow during training and then using MLflow's API to register and start serving the model is straightforward and leverages Databricks' built-in functionalities for seamless model deployment.

Option C: Building and running a Docker container is a complex and less integrated approach within the Databricks ecosystem.

Option D: Using Flask and Gunicorn is a more manual approach and less integrated compared to the native capabilities of Databricks and MLflow.

Option B provides the most straightforward and efficient process, utilizing Databricks' ecosystem to its full advantage for deploying models.

Question 3

Question Type: **MultipleChoice**

A Generative AI Engineer has created a RAG application which can help employees retrieve answers from an internal knowledge base, such as Confluence pages or Google Drive. The prototype application is now working with some positive feedback from internal company testers. Now the Generative Al Engineer wants to formally evaluate the system's performance and understand where to focus their efforts to further improve the system.

How should the Generative AI Engineer evaluate the system?

Options:

A- Use cosine similarity score to comprehensively evaluate the quality of the final generated answers.

B- Curate a dataset that can test the retrieval and generation components of the system separately. Use MLflow's built in evaluation metrics to perform the evaluation on the retrieval and generation components.

C- Benchmark multiple LLMs with the same data and pick the best LLM for the job.

D- Use an LLM-as-a-judge to evaluate the quality of the final answers generated.

Answer:

B

Explanation:

Problem Context: After receiving positive feedback for the RAG application prototype, the next step is to formally evaluate the system to pinpoint areas for improvement.

Explanation of Options:

Option A: While cosine similarity scores are useful, they primarily measure similarity rather than the overall performance of an RAG system.

Option B: This option provides a systematic approach to evaluation by testing both retrieval and generation components separately. This allows for targeted improvements and a clear understanding of each component's performance, using MLflow's metrics for a structured and standardized assessment.

Option C: Benchmarking multiple LLMs does not focus on evaluating the existing system's components but rather on comparing different models.

Option D: Using an LLM as a judge is subjective and less reliable for systematic performance evaluation.

Option B is the most comprehensive and structured approach, facilitating precise evaluations and improvements on specific components of the RAG system.

Question 4

Question Type: MultipleChoice

A Generative Al Engineer has successfully ingested unstructured documents and chunked them by document sections. They would like to store the chunks in a Vector Search index. The current format of the dataframe has two columns: (i) original document file name (ii) an array of text chunks for each document.

What is the most performant way to store this dataframe?

Options:

A- Split the data into train and test set, create a unique identifier for each document, then save to a Delta table

B- Flatten the dataframe to one chunk per row, create a unique identifier for each row, and save to a Delta table

C- First create a unique identifier for each document, then save to a Delta table

D- Store each chunk as an independent JSON file in Unity Catalog Volume. For each JSON file, the key is the document section name and the value is the array of text chunks for that section

Answer:

B

Explanation:

Problem Context: The engineer needs an efficient way to store chunks of unstructured documents to facilitate easy retrieval and search. The current dataframe consists of document filenames and associated text chunks.

Explanation of Options:

Option A: Splitting into train and test sets is more relevant for model training scenarios and not directly applicable to storage for retrieval in a Vector Search index.

Option B: Flattening the dataframe such that each row contains a single chunk with a unique identifier is the most performant for storage and retrieval. This structure aligns well with how data is indexed and queried in vector search applications, making it easier to retrieve specific chunks efficiently.

Option C: Creating a unique identifier for each document only does not address the need to access individual chunks efficiently, which is critical in a Vector Search application.

Option D: Storing each chunk as an independent JSON file creates unnecessary overhead and complexity in managing and querying large volumes of files.

Option B is the most efficient and practical approach, allowing for streamlined indexing and retrieval processes in a Delta table environment, fitting the requirements of a Vector Search index.

Question 5

Question Type: MultipleChoice

A Generative Al Engineer is building a RAG application that answers questions about internal documents for the company SnoPen AI.

The source documents may contain a significant amount of irrelevant content, such as advertisements, sports news, or entertainment news, or content about other companies.

Which approach is advisable when building a RAG application to achieve this goal of filtering irrelevant information?

Options:

A- Keep all articles because the RAG application needs to understand non-company content to avoid answering questions about them.

B- Include in the system prompt that any information it sees will be about SnoPenAI, even if no data filtering is performed.

C- Include in the system prompt that the application is not supposed to answer any questions unrelated to SnoPen AI.

D- Consolidate all SnoPen AI related documents into a single chunk in the vector database.

Answer:

C

Explanation:

In a Retrieval-Augmented Generation (RAG) application built to answer questions about internal documents, especially when the dataset contains irrelevant content, it's crucial to guide the system to focus on the right information. The best way to achieve this is by including a clear instruction in the system prompt (option C).

System Prompt as Guidance: The system prompt is an effective way to instruct the LLM to limit its focus to SnoPen AI-related content. By clearly specifying that the model should avoid answering questions unrelated to SnoPen AI, you add an additional layer of control that helps the model stay on-topic, even if irrelevant content is present in the dataset.

Why This Approach Works: The prompt acts as a guiding principle for the model, narrowing its focus to specific domains. This prevents the model from generating answers based on irrelevant content, such as advertisements or news unrelated to SnoPen AI.

Why Other Options Are Less Suitable:

A (Keep All Articles): Retaining all content, including irrelevant materials, without any filtering makes the system prone to generating answers based on unwanted data.

B (Include in the System Prompt about SnoPen AI): This option doesn't address irrelevant content directly, and without filtering, the model might still retrieve and use irrelevant data.

D (Consolidating Documents into a Single Chunk): Grouping documents into a single chunk makes the retrieval process less efficient and won't help filter out irrelevant content effectively.

Therefore, instructing the system in the prompt not to answer questions unrelated to SnoPen AI (option C) is the best approach to ensure the system filters out irrelevant information.

Question 6

Question Type: MultipleChoice

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?

Options:

A- Implement a comprehensive guardrail framework that includes policies for content filters tailored to the finance sector.

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- Incorporate manual reviews to correct any problematic outputs prior to sending to the users

Answer:

B

Explanation:

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 Relevance: 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.

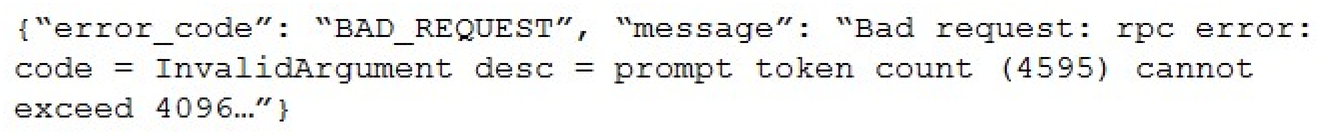
D (Manual Review): Incorporating human oversight to catch and correct issues with the LLM's output ensures the final answers are aligned with financial content expectations.

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

Question 7

Question Type: MultipleChoice

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:



```
{"error_code": "BAD_REQUEST", "message": "Bad request: rpc error:
code = InvalidArgument desc = prompt token count (4595) cannot
exceed 4096…"}
```

What TWO solutions should the Generative AI Engineer implement without changing the response generating model? (Choose two.)

Options:

A- Use a smaller embedding model to generate

B- Reduce the maximum output tokens of the new model

C- Decrease the chunk size of embedded documents

D- Reduce the number of records retrieved from the vector database

E- Retrain the response generating model using ALiBi

Answer:

C, D

Explanation:

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.

Options C and D are 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.

20%
DISCOUNT