Take Home Challenge for Junior AI Researcher

Objective:

Your task is to develop an embedding model for a provided dataset sample, enabling efficient search functionality across the dataset. The model should facilitate querying the dataset using both textual and numerical descriptors, demonstrating versatility in handling different data types and interpreting ordinal relationships.

Dataset Overview:
You will work with a sample of 100 rows across 19 columns from a shipment dataset. The dataset encompasses various attributes, including at least one date column, one categorical column, and one numerical column.

Here is a link to the dataset:
https://drive.google.com/file/d/1Xax5Q7axT4i8pp8XPRnvGIH4q1avPVPW/view?usp=sharing

Task Details:

Embedding Model Creation:

- Select any 4 columns from the dataset, ensuring you include at least one date column, one categorical column, and one numerical column.
- Develop an embedding model that can:
- Accept text as input and output the corresponding column name, value, and all rows matching that condition (Column=Value).
- Handle both numerical and textual data inputs, with the capability to interpret ordinal text (e.g., "below A", "small", "zero", "none", "highest", "Higher than A and smaller than B") for numerical columns and ordinal searches (e.g., min value, max value, most common value) for categorical columns with ordinal values (Low, High, Moderate).
- You may create separate embedding models for numerical and textual values if deemed appropriate.

Model Requirements:

- The model should be based on open-source technologies, not relying on proprietary models such as those provided by OpenAI.
- Ensure the model's response time is approximately ~10 tokens/second on an NVIDIA A100 GPU or similar hardware.

- Your model should accurately deduce column names and values from textual queries (e.g., "apparel products" should map to Column ='Product_Category', Value ='Apparel'), handle synonyms, be case insensitive, and allow for typographical errors in input.

Submission Format:

Provide your solution as Python code capable of accepting an array of strings (each string representing a textual query) and outputting the results in JSON format.

Example input format: [‘apparel product’]
Example output format: {"column_name": "Product_Category", "value": "Apparel", "row_ids": ["row1", "row2", ...]}.

If your embedding model requires downloading from the Hugging Face hub or another repository, include a link to the repository within your submission.

Evaluation Criteria:

- Functionality: The model's ability to accurately interpret and respond to both numerical and textual queries, including ordinal relationships.
- Performance: Response time in line with the specified hardware requirements.
- Code Quality: Clarity, structure, and documentation of your Python code.
- Innovation and Problem-solving: Creativity in model development and problem-solving strategies for embedding and query interpretation.

Additional Notes:

Your solutions will be discussed during the technical interview
For the final assessment, your solution will be evaluated against a larger dataset (100,000 rows, 50 columns). Ensure your model is scalable and robust against a dataset of this size.