Lights on Heights 

Project Title: Medication Information Search Engine

Project Goal: Developing a robust drug information system

Problem Statement

In the fast-paced pharmaceutical industry, a leading company faces a critical challenge: developing an advanced drug information retrieval system. This system aims to provide healthcare professionals with quick access to comprehensive information about drugs, including side effects, drug classes, activity levels, pregnancy categories, and more. As a highly skilled Machine Learning Engineer, your task is to build this system within a strict 1-week timeline.

Dataset:

The dataset provided contains a wealth of drug information, including generic names, drug classes, brand names, activity levels, prescription requirements, pregnancy categories, Controlled Substances Act (CSA) schedules, alcohol interactions, ratings, and crucially, the associated side effects of each drug. This dataset serves as the foundation for the drug information retrieval system.

Task:

Your mission is to develop a robust drug information retrieval system and present your solution within a 1-week timeframe. The assessment will focus on the following key aspects:

1. Data Pre-processing:

- Thoroughly cleanse and pre-process the dataset, ensuring data quality and removing any irrelevant or redundant information. Handle missing values appropriately and perform necessary data transformations.
- Extract relevant features without extensive feature engineering to expedite the process.

2. Information Retrieval Model Development:

- Develop a powerful machine learning or natural language processing model capable of understanding the semantic context of drug information.
- Train the model using suitable techniques, such as word embeddings or transformer architectures, to capture semantic relationships effectively.
- The model should provide accurate and efficient retrieval of drug information based on user queries, considering multiple criteria.

3. API Development and Deployment:

- Design and implement a high-quality RESTful API that seamlessly integrates with the information retrieval model. Use frameworks like Flask or FastAPI to develop the API.

- Provide API endpoints enabling users to submit search queries and receive relevant drug information based on their criteria.
- Deploy the API endpoint application on hosting services.

Example API Request:

```
{
  "query": "I want information on paracetamol",
  "focus": ["side_effects", "drug_classes"]
}
```

Example API Response:

```
{
  "results": [
    {
      "drug_name": "Paracetamol",
      "generic_name": "Acetaminophen",
      "side_effects": ["Headache", "Nausea", "Stomach pain"],
      "drug_classes": ["Analgesic", "Antipyretic"],
      "brand_name": ["Tynelol", "Mixapin"],
      "CSA": "A",
      .........
    }
  ]
}
```

4. Documentation:

- Prepare concise documentation that explains the data pre-processing steps, information retrieval model development, and API integration. Clearly articulate the decisions made, challenges encountered, and insights gained during the development process.
- Include code snippets, explanations of the model architecture, and API endpoint details. Provide clear instructions on how to interact with the API and interpret the results.
- Include instructions for setting up and utilizing the developed system.

Deliverables Summary:

The deliverables for the case study assessment include:

1. Pre-processed Dataset and Dataset Schema: A cleaned and transformed dataset in Parquet format, along with a well-defined schema outlining the data types and structure of each column. This demonstrates proper data pre-processing and ensures data consistency and integrity.

2. Jupyter Notebook: A comprehensive notebook that showcases the development process, and step-by-step implementation of the solution, and provides explanations of the code and insights gained throughout the assessment.

3. Trained Model and Word Embedding: A machine learning model or pre-trained model fine-tuned for drug side effect searching with the generated word embeddings, documented with details on architecture, training process, and performance evaluation.

4. Hosted Functional Backend API: Design and implementation of a RESTful API that integrates with the trained model and word embeddings, allowing users to input drug attributes and obtain drug information. Link to the hosted API should also be added to the documentation. Clear documentation on API usage and request/response formats should be provided.

5. Documentation: Concise README documentation covering data pre-processing steps, schema definition, model development details, and backend API implementation. It should provide guidance for interacting with the API and highlight key features.

How to Submit: Submit a link to your GitHub repository as a response to this mail.