Task Description: Machine Learning Optimization and Multimodal Challenge

Overview:

This task consists of **two steps**. Both steps involve predicting a **target variable** using **tabular data**. In the second step, you will also work with **text** and **image data** to improve the model's performance.

You will be provided with a CSV file containing several features, including `target`, `source_id`, `quantity`, and a `description` column. Additionally, there is a folder with images corresponding to the descriptions in the tabular data. The objective is to build models that optimize prediction of the **target** variable using the available data, applying the appropriate preprocessing and modeling techniques.

Step 1: Optimization with Tabular Data

For this step, you will work solely with the **tabular data**. The `description` column should be ignored for this step and dropped.

Key Features:

- **`source_id`**: Represents different data sources, which vary in the quality of their labeling.
- **`quantity`**: Represents the number of items to produce. The **target** is expected to decrease as the quantity increases, assuming similar items.

Your Task:

- You can assume that some basic feature engineering has been performed prior to this to have a workable dataset, and as you have absolutely no domain knowledge, this will be a pure optimization task.
- The data hasn't been cleaned up completely though so feel free to perform EDA and feature engineering as you judge necessary.
- Build a machine learning model that predicts the **target** variable based on the available tabular features.
- Focus on optimizing the model's performance. We're mostly interested in seeing what your methodology looks like, how you approach ML models optimization and technology choices.
- Ensure that you **remove the `description` column** for this step.

Step 2: Multimodal Learning with Text and Images

In this step, you will incorporate the **`description`** column and its associated images from the `spacecraft_images` folder into your model.

Additional Data Sources:

- **`description`**: A textual feature that describes the items. You are expected to encode this text using a neural network of your choice (e.g., embeddings, LSTMs, transformers, etc.). Do not use an ordinal or any other categorical encoder. You can assume that in a general case, these are unique.
- **Images**: Each description has an associated image in the `spacecraft_images` folder. For example, if the `description` is "Some Peculiar Description", there will be a corresponding image file named `some_peculiar_description.jpg` (or similar, with varying file extensions).

Your Task:

- Encode the **`description`** column using a neural network and use the resulting embeddings as part of your model.
- Load and preprocess the images from the `spacecraft_images` folder. You should encode these images into latent representations (using a CNN or other suitable method) and integrate them with the tabular and textual data.
- Build a multimodal model that utilizes the tabular features, text embeddings, and image embeddings to predict the **target** variable.
- We're interested in seeing your ability to:
    - preprocess multiple data types, aggregate latent representations and use them for training.
    - write clean code and dockerize it.

Deliverables:

1. **Step 1**:
    - A model that predicts the target variable using tabular data only.
    - A brief description of your optimization process (e.g., feature engineering, model selection).
2. **Step 2**:
    - A multimodal model that incorporates tabular data, text embeddings, and image embeddings.
    - A training script that loads, preprocesses, and trains the model on all three data modalities (tabular, text, and images).
3. **Dockerfile**:

    - Provide a `Dockerfile` that builds an image containing all dependencies required to run your scripts.
    - Ensure that the Docker image runs both steps of the task in an end-to-end manner.

Evaluation Criteria:

- **Model Performance**: The performance of the models in predicting the target variable.
- **Code Quality**: Clean, maintainable code with appropriate documentation.
- **Docker Setup**: A functional Docker environment that reproduces the task workflow.

AI Tools usage:

We encourage candidates to use AI as a supplementary tool and whatever they would use as a part of their regular assisting tools setup, but we value a hands-on understanding of the task and original thought process when tackling complex problems.
A deeper level of problem-solving and critical thinking is important for this role, and ChatGPT alone won't be enough.

This is a toy, but quite hard to fully optimize, application (especially without domain knowledge), which is designed specifically for interviews, so focus on showcasing your ability to work with data, train models and use pre-trained ones instead of final accuracy (even if higher is better).

A report gathering your thought process / data visualizations / results analysis is a plus, it's not mandatory. It can be a notebook, a latex document…

Good luck!