**Tracera (ESG Flo) Assessment Round – ML Interns**

Tracera (ESG Flo) emerged from Bain & Company's Venture Incubator, where advisors consistently observed their clients struggling with sustainability compliance and data collection. Our software optimizes data extraction, validation, and transformation, to make ESG reporting seamless.

The current problem revolves around data extraction from documents. Specifically, the task is to extract values for the following fields: ***Account Number****,* ***Meter Number****,* ***Start Date****,* ***End Date****,* ***Usage****,* and ***Cost***, from a set of 12 provided documents. You will also find a **ground\_truth.csv** file, which contains the actual values for comparison with your system’s extractions.

**Metric**: Use Exact Match (Accuracy) at the field level—this metric measures whether the extracted value exactly matches the ground truth value for each field. For example, accuracy will be assessed for extracting the correct Account Number.

**Things to consider:**

1. Documents may vary in format.
2. Some documents may contain multiple values for the fields mentioned above.
3. Keep in mind the document's region (US, EU, or Asia) and the corresponding data and number formatting conventions.
4. If a document does not have a value for a particular field, represent the missing value with a hyphen (-).

**Submission Guidelines:**

* Submit a **ZIP file** containing all code, required files, and outputs needed to run your solution end-to-end on our side.
* Include a single **CSV or Excel file** containing all extracted values across all processed documents.
* **Submissions that cannot be executed as-is will be considered invalid.**
* Include a **Jupyter Notebook (.ipynb)** or **Google Colab notebook** with your implementation. Python scripts are also acceptable, but notebooks are preferred for ease of review and debugging.
* Provide a **clear README** with step-by-step instructions to set up and run your code. If using a virtual environment, list all dependencies and ensure we can reproduce your environment without manual fixes.

Sample code to calculate exact match accuracy metric.

# Function to calculate column-wise accuracy

def calculate\_column\_accuracy(ground\_truth\_df, model\_output\_df, columns):

accuracies = {}

for column in columns:

correct\_count = (ground\_truth\_df[column] == model\_output\_df[column]).sum()

total\_count = len(ground\_truth\_df)

accuracy = (correct\_count / total\_count) \* 100

accuracies[column] = accuracy

return accuracies

# Specify columns to check accuracy

columns\_to\_check = ["Account Number", "Meter Number", "From Date", "To Date", "Usage", "Cost"]

# Calculate accuracies

column\_accuracies = calculate\_column\_accuracy(ground\_truth\_df, model\_output\_df, columns\_to\_check)

# Print the results

for column, accuracy in column\_accuracies.items():

print(f"Accuracy for {column}: {accuracy:.2f}%")