Machine Learning Technical Challenge (Doutor-IE)

1. Overview

1.1 Objective

The objective of this challenge is to develop a fault detection model for an automotive scanner. A collection of datasets are provided in the data corpus of this challenge.

1.2 Data Corpus

The datasets contain real cases collected from users across Brazil. Additionally, a special test case was recorded where a fault was intentionally induced.

Structure:

```
ml_technical_challenge/
├─ challenge_explanation.pdf
└─ datasets/
   ├─ references/
   |      └─logs.csv            # csv file containing reference cases
   └─ fault_example.csv     # csv file containing cases where a fault should be detected
```

- **fault_example.csv:** recorded during a test with an induced fault.
    - Condition: air-fuel mixture became rich at idle.
    - Expected faulty parameters:
        - *Short-Term Fuel Trim (STFT):* low values (around -10% to -15%).
        - *Oxygen Sensor 1, Bank 1:* indicating a rich mixture (0.8V to 0.9V).
        - *MAP Sensor:* values outside expected range.
    - Vehicle also had a faulty battery, resulting in low battery voltage, which may be detectable by the algorithm.
- **Key parameters for modeling:**
    - Coolant Temperature Sensor

- Calculated Load
- Engine RPM
- Altitude

These variables provide the necessary context to characterize engine operation and detect abnormal behavior under varying conditions.

> **Note:** For the dataset to be considered valid for training, the column **"number of trouble codes"** must be equal to zero.

---

2. Project Requirements

2.1 Functional Requirements

The service must perform the following operations:

1. Create an application capable of outputting fault detection results given a scan input.
2. Utilize the **reference dataset** as the baseline of normal vehicle behavior.
3. Implement ETL processes to prepare and clean the data.
4. Train a machine learning model to detect deviations from normal conditions.
5. Ensure the model can identify the induced fault case (rich mixture at idle + low battery voltage).
6. Provide an interface (API, script, Jupyter Notebook, etc) that receives scan results and outputs the detected faults.

2.2 Documentation

The project should include extensive documentation explaining:

1. The overall approach and reasoning for the chosen model.
2. ETL process details.
3. How to test the solution.
4. Challenges faced and proposed solutions.

5. Summary and potential future improvements.

2.3 Deadline

- **Estimated coding time:** 8 hours
- After the 8-hour deadline, no changes should be made in the public repository.

---

3. Validation and Evaluation

3.1 Test Cases

The solution will be evaluated using both reference datasets and the induced fault scenario.

Example validation cases:

- **User Input:** Logs with normal operating conditions.
  Expected Output: “No fault detected.”
- **User Input:** `fault_example.csv` (rich mixture + low battery).
  Expected Output: “Fault detected – rich air-fuel mixture at idle and low battery voltage.”
- **User Input:** Logs with missing or invalid values.
  Expected Output: “Invalid input – unable to evaluate.”

*the expected output model is just figurative - you can format as preferred as long as the results are clearly formatted.