Janitri Data Science Challenge

Problem Statement 1: FHR and UC Analysis

Objective: You are tasked with analyzing a dataset containing FHR (Fetal Heart Rate) and UC (Uterine Contractions) data. Your goal is to plot the graphs, perform FHR analysis over time epochs, and analyze peaks in the UC graph to provide insights.

Task Details:

1. **Plotting Graphs**:
    - Plot two separate graphs:
        1. **Time vs FHR**: This should display the fetal heart rate over time.
        2. **Time vs UC**: This should display uterine contractions over time.
2. **FHR Analysis**:
    - Divide each minute of the NST (Non-Stress Test) reading into sixteen 3.75-second epochs.
    - For each epoch, calculate the **average FHR** in two ways:
        1. **Beats per minute (bpm)**.
        2. **Pulse interval (milliseconds)**. Example: An FHR of 120 bpm is equivalent to a pulse interval of 500 milliseconds.
    - Use the average pulse interval of each epoch for further analysis.
3. **UC Peak Detection**:
    - Using any suitable Python library (e.g., SciPy), perform peak detection on the UC graph.
    - For each detected peak:
        1. Find the **width of the peak at half its maximum height**.
        2. Count peaks where the width is more than 30 seconds.
    - Calculate and report the **average duration** of the counted UC peaks.

Dataset Description: FHR and UC Simulator Data

Overview: This dataset contains simulated data representing **Fetal Heart Rate (FHR)** and **Uterine Contractions (UC)**. The data is generated using a simulation tool and is designed to mimic real-world conditions observed during non-stress tests (NST) used in maternal and fetal health monitoring.

Data Frequency:

- Each data point is recorded at **250-millisecond intervals** (i.e., four data points per second), providing high-resolution temporal data for both FHR and UC.

Data Fields:

1. **Time (milliseconds)**: Represents the time in milliseconds since the start of the recording. The time intervals between consecutive rows are exactly 250 milliseconds.
2. **FHR (beats per minute)**: The fetal heart rate, measured in beats per minute (bpm). This field represents how frequently the fetal heart is beating over time.
3. **UC (TOCO)**: Uterine contractions, represented by a numeric value indicating the strength and occurrence of contractions. These values can be analyzed to detect peaks and durations of contractions over time.

Example Data:

Time (ms)	FHR (bpm)	UC (TOCO)
250	140	5
500	138	7
750	136	6
1000	140	8
1250	142	9

Deliverables

---

1. **Solutions:**
   - Plots for FHR and UC data.
   - Detailed analysis of FHR over epochs (both in bpm and pulse intervals).
   - Peak detection and analysis for UC, including the count of peaks wider than 30 seconds and the average duration of these peaks.

2. **Code Files:** Attach all your code files used to generate the solution.

3. **Writeup:** A detailed explanation of your solution. It may also contain other approaches that you think will work for the problem and some comparison between the approaches. We would also suggest you submit your best approach for solving problems even if you are not able to generate solutions for the problem.

Problem Statement 2: Image Segmentation Model

Objective: You are required to create an **image segmentation model** on any dataset of your choice. The goal is to accurately segment different regions within the images using a machine learning or deep learning approach.

Task Details:

1. **Dataset**:
    - Choose or use any publicly available dataset for image segmentation. (Examples: medical images, satellite images, or everyday objects).
    - You can also generate your own dataset if you prefer.
2. **Model Development**:
    - Build an image segmentation model using any technique of your choice:
        - You may use traditional methods like **thresholding** or **edge detection**.
        - Alternatively, you can use deep learning models like **U-Net**, **Mask R-CNN**, or any other convolutional neural network (CNN) architecture.
3. **Evaluation**:
    - Evaluate the segmentation performance using metrics such as **Intersection over Union (IoU)** or **Dice Coefficient**.
4. **Visualization**:
    - Provide visual outputs showing the original image alongside the segmented regions to demonstrate your model's effectiveness.

Deliverables:

1. **Model code**: Provide the Python code for the model and training pipeline.
2. **Report**: A brief report describing the dataset, the model used, and the evaluation results.
3. **Visualization**: Include segmented images alongside their original counterparts for comparison.

Bonus:

- Implement a pre-trained model on a complex dataset like **medical or satellite images** for advanced segmentation.

Download

You can get the Dataset from the following Source:

1. Use this link to get access to the Dataset.
2. Ask for a local copy if there are any issues in accessing the Data.

How to Submit

DOs

- Make sure the file_name of your solution file follows your_name_FDS.
- Make sure you attach the corresponding code files used for result generation.
- Make Necessary assumptions whenever needed and provide appropriate explanations.
- Please maintain coding standards.

DON'Ts

- Attach huge model/embeddings/cache files.

Submit

To make a submission, upload your zipped generated files as mentioned in the deliverables in this **google form** within 24 hours of receipt of the assignment.

https://forms.gle/rkmNnJtwkHdSy5GA6.

Points to ponder

Using a ML technique does not necessarily mean training a new model. Please feel free to use any pre-trained model for any task.

If you are not able to solve a problem, we would appreciate it if you submit your best approach to solve that problem instead of the solution.