Epoch Problem Statements

Instructions-

- Choose any one of the following four problem statements
- Follow the hackathon ethics, code of conduct
- Plagiarism will be checked

Audio/Speech Processing Domain:

1) **Multilingual unit to WAV vocoder**
   The goal is to train a low-footprint GAN vocoder for generating multilingual audios from discrete unit/token representations from a generative audio model.

2) **End-to-end speech translation**
   The aim is to translate speech from one language directly to another language using generative models without any text transcription in the latent space. You are required to translate from **English** to **Hindi** language for this task.

   Dataset
   You are required to process the audio from the following video:

   RIISE-2022-Video

Data Science Domain:

3) **Implement the following Algorithms:**
   Please read this survey. You must understand the definition of 'coreset' and the operations in algo-1 & algo-2 (ignore theoretical analysis). Implement the algorithms in Python.

   Regarding attachments:

   - **Uniform_Sampling.py** constructs a fixed size coreset, uniformly at random, and returns the relative error performance.
   - The file **wkpp.p**y is weighted kMeans++; this is required to report the performance of coresets (see Uniform_Sampling.py for reference).
   - Use the **skeleton.py** to write your code, or take a hint from Uniform_Sampling.py and write your version of the code. Your code must return the relative error performance of both practical and uniform sampling-based coresets. You may use any Python libraries to compute the survey's internal operations in algo-1 & algo-2.

   Download the attachments

   Attachments on Github: Github Repository Link

   You can download the attachments from either of the two links.

Advanced task(Bonus)
Based on your understanding of the coreset from the survey paper or the skeleton file, showcase how image segmentation looks like on the coreset compared to image segmentation on the original file. An image file has been attached for your reference.

4) **Efficient And Fair Line Construction**
Consider a set of n houses with coordinates given by latitude (x) and longitude (y), defined as a set P = {p_1, p_2, . . . , p_n}. The task is to construct a gas pipeline in a straight line that serves all of them, where a straight line ℓ = {a, b} is defined by a direction vector a of unit norm and a point b on the line.

Distance Definition
The distance from a point p to a line ℓ ={a,b} is given by:

$$\text{dist}(p, \ell) := \left\|(I - aa^T) \cdot (p - b)\right\|^2$$

Objective 1: Efficient Line (Do not use any library function to find the line)
A line is efficient if it minimizes the following cost:

$$\sum_{i \in [n]} \|\text{dist}(p_i, \ell)\|$$

Design an algorithm that computes an efficient line or a line which is almost efficient, meaning the cost is a local minimum.

Objective 2: Fair Line (Do not use any library function to find the line)
A line is fair if it minimizes the maximum distance to any house:

$$\min_{\ell} \max_{i \in [n]} \|\text{dist}(p_i, \ell)\|$$

Design an algorithm that computes a fair line or a line which is almost fair, meaning the above cost is a local minimum.

Objective 3: Multiple Efficient Lines (Do not use any library function to find the lines)
For a set of k lines L = {ℓ_1, . . . , ℓ_k}, the set is efficient if it minimizes:

$$\sum_{i \in [n]} \min_{\ell \in \mathbf{L}} \|\text{dist}(p_i, \ell)\|$$

Design an algorithm that computes k efficient lines or a set of lines which is almost efficient, meaning the above cost is a local minimum.

Dataset
Use the latitude and longitude from the California dataset provided by ‘sklearn‘ to solve these problems.
Calfornia Housing Dataset