**Problem Statement – 1 : Bounding Box Merging using Data Science algorithms**

**Objective:**

Given a ZIP (screening\_data\_csv.zip) folder with 51 images of an excel sheet with two columns of bounding box coordinates, where each row contains bounding box coordinates that are closely related and can potentially be merged into a single bounding box. The objective is to develop:

1. An OCR (Optical Character Recognition) based pipeline that can read the data from the images and save the data to a CSV file.
2. A data science model that can identify and merge these related bounding boxes automatically.

**Input:**

A directory of images that contains 51 images, each one is a portion of a full excel sheet containing two columns: List A and List B. Each column contains bounding box coordinates in the format [x1, y1, x2, y2]. The bounding box coordinates in each row of List A correspond to matched bounding box coordinates in the same row of List B, forming pairs that are closely related and can be merged.

**Expected Output:**

Given a list of coordinates from List A and List B, the solution should output matched pairs of bounding box coordinates (from List A and List B), indicating which bounding boxes from List A and List B can be combined into a single bounding box.

**Constraints:**

* Bounding box coordinates are represented as [x1, y1, x2, y2], where (x1, y1) represents the top-left corner and (x2, y2) represents the bottom-right corner of the bounding box.
* The model should accurately identify and merge bounding boxes that are closely related and can be combined into a single bounding box.

**Deliverables:**

* Code implementing the OCR for converting the data from images to CSV.
* Code implementing the data science model for merging closely related bounding boxes.
* Evaluation metrics to assess the accuracy and efficiency of the model in merging bounding boxes.
* [Optional]: Evaluation metrics to assess the OCR implementation.
* Documentation outlining the approach, algorithms used, input/output formats, and instructions for running the system on new datasets.

**Problem Statement -2: Sentiment Analysis Webapp**

Objective:

Given a dataset for sentiment analysis of tweets. The zip file contains train.csv and test.csv file corresponding to training and testing data. The objective here is to develop:

1. A sentiment analysis pipeline using the train data (You are free to use any kind of algorithm for the same. ), and tested on test data.
2. A UI based webapp that deploys the model from step-1 and can let the user to test the sentiment of new tweets. (A Streamlit app is preferred)

Input:

The data is a CSV with emoticons removed. Data file format has 6 fields:

1. The polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)
2. The id of the tweet (2087)
3. The time of the tweet ()
4. The age of the user who tweeted
5. The country of the user who tweeted
6. The text of the tweet (Lyx is cool)

Deliverables:

* Code implementing the sentiment analysis training and testing.
* Evaluation metrics to assess the accuracy of the model.
* Demo webapp which can read new sentences and assess the sentiment. Code for the same.