****

**Work Integrated Learning Programmes Division**

**ML System Optimization**

## S2-23\_AIMLCZG516

**2023 -24**

**Assignment 2 – PS5 [Weightage 10%]**

**Instructions for Assignment Evaluation**

1. Please follow the naming convention as <Group no>\_<PS\_Number>.ipynb.

Eg – for group 1 with PS1 should be named as - Group1\_ PS1.ipynb.

1. Inside each jupyter notebook, you are required to mention your name, Group details and the Assignment dataset you will be working on.
2. Organize your code in separate sections for each task. Add comments to make the code readable.
3. Notebooks without output shall not be considered for evaluation.
4. Delete unnecessary error messages and long outputs.
5. Display the analysis of attributes in one frame rather than one after one. However, special treatment to attributes can be displayed separately.
6. Prepare a jupyter notebook (recommended - Google Colab) to build, train and evaluate the model.
7. Each group consists of up to 4 members. All members of the group will work on the same problem statement.
8. Each group should upload in CANVAS in respective locations under ASSIGNMENT Tab. Assignment submitted via means other than through CANVAS will not be graded.
9. Submission: Files should be uploaded on canvas without zipping them. One is ipynb file and other one html or pdf with output of the ipynb file.

**Problem Statement**

Principal Component Analysis (PCA) with cuML

Perform dimensionality reduction using cuML’s PCA.

**Criteria:**

* **Principal Component Analysis (PCA) with cuML** (6 Points):
  + Load a dataset of your choice (or use a synthetic one with at least 20 features and 100,000 samples).
  + Normalize the dataset (mean = 0, variance = 1).
  + Apply PCA using cuML to reduce the dimensionality to 3 principal components.
  + Plot the first two principal components in a scatter plot.
* **Functionality** (2 Points):
  + Compare the time taken by cuML’s PCA and scikit-learn’s PCA on the same dataset.
* **Code Quality and Efficiency** (2 Points):
  + Submit the script, the scatter plot, and a comparison of the execution times for cuML and scikit-learn.

**\*\*\*\*\*\*\*\*\***