**Project 5 - Team 3: Ellen, Deji, Answer, Manroop, Sukwinder, Ameera**

**Project Proposal: Classification of Rice using CNN**

**Datasource**: <https://www.kaggle.com/code/nmaleknasr/rice-image-classification-cnn-tensorflow>

**Objective statement:**

To develop a deep learning-based rice image classification model using TensorFlow that can accurately identify and categorize different types of rice grains from images. The project aims to leverage convolutional neural networks (CNNs) to process visual features of rice grains and classify them into predefined categories (e.g., Basmati, Jasmine, Brown rice) to assist in agricultural, industrial, or research applications.

**Project Scope (Brief):**  
The **Rice Image Classification** project focuses on implementing a machine learning model that classifies rice grain images into predefined categories with meaningful predictive power. The model development adheres to the following guidelines:

1. **Data Collection and Cleaning:**
   * Gather rice grain images from available datasets or create a custom dataset.
   * Clean the data by removing corrupted or irrelevant entries.
   * Normalize and standardize image data for consistent training.
2. **Model Implementation:**
   * Use a Python script to initialize, train, and evaluate a classification model using TensorFlow.
   * Apache Spark to retrieve and preprocess the data.
3. **Optimization and Evaluation:**
   * Document the model optimization process, including iterations in hyperparameters, architecture, and preprocessing steps.
   * Track and log model performance improvements in a CSV file or directly within the Python script.
4. **Deployment and Impact:**
   * Deploy the trained model to demonstrate real-world usage, such as sorting or quality control.
   * Present insights from the project, emphasizing the improvements achieved through optimization.

**Tools and Technologies Used:**

1. Programming Language:
   * Python
2. Frameworks and Libraries:
   * TensorFlow : For building, training, and evaluating the model.
   * NumPy & Pandas: For data manipulation and preprocessing.
   * SQLAlchemy or PySpark: For retrieving and handling data from SQL or Spark databases.
   * Scikit-learn: For metrics and additional data processing (e.g., train-test split, normalization).
3. Development Environment:
   * Google Colab : For interactive development and GPU support.
   * VSCode: For script-based implementation.
4. Version Control:
   * Git/GitHub: For tracking iterations in the project.
5. Visualization Tools:
   * Matplotlib & Seaborn: For visualizing data and performance metrics.
6. Data Storage and Retrieval:
   * SQL: Use PostgreSQL or MySQL for data retrieval.
   * Spark: If dealing with large-scale data, Apache Spark can handle distributed processing.

**Outcome**

We should be able to determine how well the model performs in training and validating the images.