## **Data Science Technical Assessment**

**Background**

Raising The Village (RTV) partners with last-mile communities to achieve a $2/day income target within 24 months. This assessment evaluates your ability to support this mission through data science solutions.

**Overview**

* Duration: **6 hours total**
* Format: Two independent tasks
* Dataset: Single file containing structured demographic/economic data and unstructured program feedback
* Submission: GitHub repository with documented code and instructions

**Data Dictionary**

Program Codes for **most\_recommend\_rtv\_program** and **least\_recommend\_rtv\_program**

* 1: Agriculture & Nutrition
* 2: WASH
* 3: Water
* 4: Access to Health
* 5: VSLAs
* 99: None

**Key Variables**

* HH Income + Production/Day (USD): Daily household income
* most\_recommend\_rtv\_program: Preferred program (check the data dictionary)
* least\_recommend\_rtv\_program: Least preferred program (check the data dictionary)
* most\_recommend\_rtv\_program\_reason: Corresponding text explanations for program preferences
* least\_recommend\_rtv\_program\_reason: Corresponding text explanations for program preferences

**Task 1: Risk Assessment Model (3 hours)**

**Objective**

Develop a predictive model to identify households at risk of not achieving the $2/day income target, enabling targeted interventions by program managers.

**Requirements**

1. Data Preparation & Exploration

* Clean and prepare the dataset
* Analyze income distribution and relevant factors
* Process text fields

1. Model Development

* Engineer relevant features
* Train a classification model
* Evaluate performance with appropriate metrics

1. MLOps Design

* Design automated pipeline for data processing
* Include model retraining strategy
* Outline monitoring approach

**Expected Deliverables**

* Jupyter notebooks with analysis and model development
* Python scripts for production implementation
* Documentation of approach and results

**Task 2: RAG Chatbot Development (3 hours)**

**Objective**

Create a chatbot system that analyzes community feedback from program recommendation responses to provide insights about RTV programs, helping stakeholders understand participant experiences and improve program effectiveness.

**Data Sources for RAG System**

The chatbot's knowledge base consists of four text fields from the dataset:

* `**most\_recommend\_rtv\_program**`: Preferred program (check the data dictionary)
* `**least\_recommend\_rtv\_program**`: Least preferred program (check the data dictionary)
* `**most\_recommend\_rtv\_program\_reason**`: Text explanations for preferred programs
* `**least\_recommend\_rtv\_program\_reason**`: Text explanations for least preferred programs

These fields contain the community feedback your system will analyze.

**Requirements**

1. **Retrieval System Development**

* Generate embeddings for both reason fields
* Implement vector storage solution for the community feedback embeddings
* Create semantic search functionality that can find relevant feedback across both positive and negative responses

1. **Chatbot Implementation**

* Develop retrieval logic to fetch relevant community feedback
* Integrate with generative model (open-source or free API e.g gemini) to synthesize responses based on retrieved feedback
* Create simple interface for stakeholder queries

1. **Example Functionality**

The system should handle queries about community feedback such as:

* "What reasons do communities give for recommending Agriculture & Nutrition programs?"
* "What are common concerns expressed about WASH programs?"
* "Show me feedback from communities that preferred Health Access programs"

1. **MLOps Considerations**

* Design pipeline for processing new community feedback
* Plan for regular embedding updates as new feedback is collected
* Include monitoring of retrieval relevance

**Expected Deliverables**

* Jupyter notebooks demonstrating:
  + Text preprocessing of reason fields
  + Embedding generation and storage
  + Retrieval system implementation
  + Example query responses
* Python scripts for production deployment
* Documentation of design decisions and system architecture