AN AGENTIC EXPLAINABLE AI SYSTEM FOR AGRICULTURAL CROP PRODUCTION PREDICTION AND SUITABILITY MAPPING

INSTRUCTIONS MANUAL

Submitted by

HARISH BALAJI V [RA2112701010005]
HARIHARAN K [RA2112701010016]

MASTER OF TECHNOLOGY (INTEGRATED)
in
ARTIFICIAL INTELLIGENCE

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203

MAY 2025

Instruction Manual: Integrated Agricultural AI Assistant

1. Introduction:

This document provides instructions for setting up and running the Final_Code.py script, an Integrated Agricultural AI Assistant built using Streamlit. This application offers two main functionalities:

1. Crop Production Analysis: Predicts crop production based on historical data using an AI agent workflow.
2. Crop Suitability Map: Visualizes predicted crop suitability for different districts using a machine learning model and geographic data.

2. Included Files:

The core component provided is the Python script:

- Final_Code.py: The main application code containing all logic for data loading, AI model integration (phi-agent, TensorFlow/Keras), map generation (Folium), and the Streamlit user interface.
- DataSet are attached in the Google Drive Folder.

3. Software and System Requirements:

To run this application, you need the following installed on your computer:

- Operating System: Windows, macOS, or Linux.
- Python: Version 3.8 or higher is recommended. You can download Python from python.org.
- Required Python Libraries: The application relies on several libraries. These need to be installed using pip, Python's package installer.
  - streamlit
  - pandas
  - numpy
  - folium
  - joblib
  - scikit-learn (sklearn)
  - google-generativeai
  - phidata
  - python-dotenv

    - warnings (Usually built-in)
    - os (Usually built-in)
    - time (Usually built-in)
    - json (Usually built-in)
    - logging (Usually built-in)
    - traceback (Usually built-in)
- Optional Python Libraries:
    - tensorflow or tensorflow-cpu/tensorflow-gpu: Required for the "Crop Suitability Map" feature. If not installed, this feature will be disabled, and a warning will be shown. Install one of these based on your system and GPU availability.
- API Keys:
    - Google Gemini API Key: Required for both the Production Analysis explainer agent and the Suitability Map explanation. The code currently uses a hardcoded key.
    - Groq API Key: Required for the Production Analysis explainer agent (using Llama-3.1). The code currently uses a hardcoded key.
- Data Files and Models: You must have the following files available at the exact paths specified in the Final_Code.py script (or modify the paths in the code to match your file locations):
    - C:\Users\haris\OneDrive\Major_project\dataset\New\ICRISAT-Averaged-Data_Set_agent.csv (Production Data)
    - C:\Users\haris\OneDrive\Major_project\dataset\New\cleaned_merged_data.csv (Suitability Data)
    - C:\Users\haris\OneDrive\Major_project\Code\New_Implementation\crop_recommendation_model_v3.h5 (Suitability Model - Keras)
    - C:\Users\haris\OneDrive\Major_project\Code\New_Implementation\label_encoder_v3.pkl (Suitability Label Encoder)
    - C:\Users\haris\OneDrive\Major_project\Code\New_Implementation\scaler_v3.pkl (Suitability Scaler)
    - combined_agri_workflows.db (Workflow storage - this file will be created automatically if it doesn't exist, typically in the same directory as Final_Code.py, unless DB_FILE path is absolute).

4. Setup and Installation

Follow these steps to set up your environment:

1. Install Python: If you don't have Python 3.8+ installed, download and run the installer from python.org. Make sure to check the option "Add Python to PATH" during installation.
2. Open Terminal or Command Prompt:
   - On Windows, search for "Command Prompt" or "PowerShell".
   - On macOS or Linux, open the "Terminal".
3. Install Required Libraries: Run the following command in your terminal:
4. pip install streamlit pandas numpy folium joblib scikit-learn google-generativeai python-dotenv
5. Install phidata
6. Install TensorFlow (Optional but Recommended): If you want the Suitability Map feature to work, install TensorFlow. Choose the appropriate version for your system (CPU or GPU). For most users, the CPU version is sufficient and easier to install:
7. pip install tensorflow # Installs the default version, usually GPU if CUDA is available # OR for CPU only:
   pip install tensorflow-cpu
8. Obtain API Keys: Get your Google Gemini API key (e.g., from Google AI Studio) and Groq API key (e.g., from Groq Console).
9. Place Data and Model Files: Ensure the data CSV files (ICRISAT-Averaged-Data_Set_agent.csv, cleaned_merged_data.csv) and the model files (crop_recommendation_model_v3.h5, label_encoder_v3.pkl, scaler_v3.pkl) are located at the exact paths specified in the Final_Code.py script. If your files are in different locations, you must edit the Final_Code.py script and update the paths for PRODUCTION_DATA_PATH, SUITABILITY_DATA_PATH, MODEL_PATH_SUITABILITY, ENCODER_PATH_SUITABILITY,
   and SCALER_PATH_SUITABILITY.
10. Update Hardcoded API Keys (Optional but recommended to check): Open Final_Code.py in a text editor. Find the lines for GEMINI_API_KEY and GROQ_API_KEY. Although they are hardcoded with placeholder values in the provided code snippet, verify they contain your actual valid API keys.
11. GEMINI_API_KEY = "YOUR_ACTUAL_GEMINI_API_KEY" # Replace

YOUR_ACTUAL_GEMINI_API_KEY

12. GROQ_API_KEY = "YOUR_ACTUAL_GROQ_API_KEY" # Replace YOUR_ACTUAL_GROQ_API_KEY

5. Execution Steps

Once the setup is complete and you have verified the file paths and API keys in the script, follow these steps to run the application:

1. Open Terminal or Command Prompt.
2. Navigate to the Directory: Use the cd command to go to the directory where you saved the Final_Code.py file.
   - Example: cd C:\Users\YourUsername\Documents\AgriApp (Replace with your actual path)
3. Run the Streamlit Application: Execute the following command:
   streamlit run Final_Code.py
4. Access the Application: After running the command, your default web browser should automatically open a new tab displaying the Streamlit application (usually at http://localhost:8501). If it doesn't open automatically, copy and paste the URL displayed in your terminal into your browser.
5. View Logs: Keep the terminal window open. It will display logs and messages from the application, which can be helpful for debugging if you encounter issues.
6. Stop the Application: To stop the application, go back to the terminal window where you ran the streamlit run command and press Ctrl + C.

6. Using the Application

The application runs in your web browser.

1. Navigate Tabs: Use the tabs at the top of the page to switch between the two features: "📈 Crop Production Analysis" and "🗺️ Crop Suitability Map".
2. Crop Production Analysis Tab:
   - Select a crop from the dropdown list.
   - Click the "Analyze [Selected Crop] Production" button.
   - Wait for the analysis to complete. The application will display a spinner and progress updates.
   - Review the results: Model performance metrics (MAE, R²), key factors influencing production (feature importance), sample predictions vs. actual values,

and an AI-generated explanation.

3. Crop Suitability Map Tab:
    - Click the "Generate Crop Suitability Map" button.
    - Wait for the map to load and populate. A spinner and progress updates will be shown.
    - Interact with the map:
        - Pan and zoom.
        - Hover over markers to see tooltips with brief information.
        - Click on markers to open popups with detailed suitability predictions for that location, including the data point it was based on.
        - Use the layer control icon (often in the top-right) to show/hide marker clusters.
        - Refer to the legend (bottom-left) for crop colors.
    - Read the AI-generated explanation below the map for guidance on how to interpret the map and its predictions.

7. Troubleshooting

- "ModuleNotFoundError: No module named '[module_name]'": You are missing a required Python library. Go back to Section 4, Step 3 (and Step 4 for TensorFlow) and install the missing library using pip install [module_name].
- "FileNotFoundError: [file_path]": The script cannot find one of the required data or model files.
    - Verify that the file exists at the path specified in the error message.
    - Check the Final_Code.py script and confirm the paths defined (e.g., PRODUCTION_DATA_PATH, MODEL_PATH_SUITABILITY) are correct for where you saved your files.
    - You will likely need to edit the script to correct these paths.
- "Error configuring Google Gemini API" / "Error configuring Groq API":
    - Check your internet connection.
    - Verify that your API keys are correct.
    - Ensure the API keys are placed in the Final_Code.py script as shown in Section 4, Step 7 (or configure st.secrets if you choose that method).
    - Check the terminal for more detailed error messages from the API calls.
- "TensorFlow/Keras not found. Crop Suitability Map feature will be disabled.": This is a

warning, not an error. The Suitability Map feature will simply not work. If you want this feature, install TensorFlow as described in Section 4, Step 4.

- "Insufficient data..." / "Target '{target}' has <= 1 unique value..." (Production Analysis): The selected crop data in the production CSV is not suitable for training a model (e.g., too few data points, production values are all the same). Try selecting a different crop.
- "Prediction Failed" markers on Suitability Map: Indicates that the suitability prediction for that specific district failed, possibly due to issues with the nearest data point or an internal model error. The popup/tooltip might provide more details. The AI explanation will mention errors encountered.
- Blank Screen or Page Not Loading: Check the terminal where you ran streamlit run. Python errors will be printed there.