Big Data Analytics Project Report
BAX 423- 001

By:

Mahnoor Shahid, Avi Reissberg, Siyi (Susie) Zhang, Tingwei (Elsiey) Hu, Peilu Han

Group11-Big-Data-Project-AgriAid

1. Business objective

Our top business priority is to empower farmers by reducing the uncertainty caused by unpredictable weather and limited access to reliable agricultural insights. We aim to support farmers in making data-driven decisions about crop selection and resource allocation to optimize planting strategies. This initiative is focused on improving crop yields and overall farmer livelihoods. By simply entering their location coordinates (latitude and longitude), farmers will receive personalized recommendations for the top three crops to grow along with predicted yield estimates, enabling them to make timely and informed decisions.

2. Key actionable business initiative

To bridge the gap between data and decision-making in agriculture, our initiative focuses on building a smart, scalable solution tailored to smallholder farmers. Below are the key actions driving the development and deployment of **AgriAid**.

- Develop and scale a farming agent called AgriAid that delivers precise, location and crop-specific weather forecasts (temperature and rainfall) using real-time meteorological data
- Implement a crop yield prediction model by integrating climate data, soil composition, and historical agricultural yields to forecast expected yields for different crops at specific locations.
- Farmers simply enter their latitude and longitude and can optionally assign relative importance to crops (e.g., if 1 kg of wheat = 5 kg of maize, input 5x for wheat and 1x for maize). Based on this and local environmental data, the assistant predicts and recommends the top 3 most profitable crops to grow, along with their expected yields.
- Expand global data coverage, moving beyond the initial India-centric deployment to support smallholder farmers in other regions worldwide.
- Integrate a natural language interface (NLP) to allow farmers to interact with the assistant in simple, conversational language, making recommendations and decision support more accessible and user-friendly.

The most powerful initiative is the creation of AgriAid, a smart farming agent that leverages a farmer's geographic coordinates (latitude and longitude), historical weather conditions, and soil property estimates to recommend the most suitable crops to grow along with predicted yield estimates tailored to their specific environment.

3. Metrics of Success

To evaluate the success of our initiative, we will focus on both technical performance and real-world impact for smallholder farmers. The key metrics include:

- **Prediction Accuracy:** Measured by metrics like R^2 score, RMSE, and MAE for crop yield prediction across different locations and crop types.

- **Crop Recommendation Effectiveness:** Success will be reflected in whether the top 3 recommended crops are aligned with the highest actual yields or profitability in a given location.
- **User Adoption Rate:** Number of farmers actively using the tool, particularly across new regions beyond the initial India rollout.
- **Decision Impact:** Farmer feedback on whether the tool helped them make better planting decisions, reduce risk, or improve outcomes (via surveys or engagement data).
- **Response Time & Accessibility:** Average time from user input (e.g., lat/lon) to recommendation delivery; ability to access insights via low-tech channels like SMS or voice in local languages.

Together, these metrics ensure we are evaluating both model performance and the practical value delivered to the end user.

3.1 Prioritization of metrics

To effectively measure the success of our initiative, we have prioritized a focused set of metrics that balance model performance, user adoption, and real world impact on farmer decision-making. The top three metrics are:

1. **Prediction Accuracy (R^2, RMSE, MAE):**
   Measures how well the model predicts crop yields essential for building farmer trust and ensuring that recommendations are reliable.
2. **Farmer Adoption & Usage Rate:**
   Reflects how many farmers are actively using the tool. High usage indicates that the solution is accessible, understandable, and seen as valuable.
3. **Decision Impact / Farmer Feedback:**
   Captures whether the recommendations actually help farmers improve crop planning and yield outcomes which is the core business goal.

3.2 Hypothesized Impact of the Business Initiative

We hypothesize that by providing smallholder farmers with personalized, data-driven crop and yield recommendations based on their location, soil composition, and local weather conditions, our initiative will significantly improve both decision quality and farming outcomes.

Specifically:

- We expect to achieve a prediction accuracy (R^2) of 0.7 or higher, indicating a strong correlation between predicted and actual crop yields.
- We anticipate that at least 60% of farmers who receive recommendations will adopt one of the top 3 suggested crops.

- Over a single planting cycle, we estimate a 10–20% improvement in average yield for farmers who follow the system's guidance compared to those who don't, based on historical baselines.

4. Role of analytics

Analytics played a central role in enabling, refining, and evaluating our business initiative in the following ways:

- **Enabling:** We used analytics to build predictive models (e.g., XGBoost and TabNet) that estimate crop yield based on soil, weather, and geolocation data. This enabled us to translate raw environmental inputs into actionable recommendations for farmers.
- **Refining:** By analyzing model performance (RMSE, MAE, R^2), comparing algorithms, and visualizing feature importance, we continuously refined our feature selection, preprocessing pipeline, and model choice to improve accuracy and reliability across diverse locations.
- **Evaluating:** We evaluated the effectiveness of our solution by validating predictions against actual yield data, using both statistical metrics and location-specific insights. This helped us assess whether the solution could scale, deliver real-world value, and support farmers' decision-making in various agro-climatic zones.

5. Thinking through the analytics

5.1 Data

We are relying on existing, publicly available datasets to support our analytics and model development efforts.To create the first iteration of the MVP we used the following data sets.

- **Weather Data**: We use the "Global Daily Climate Data" dataset from Kaggle, which contains historical weather observations (temperature, precipitation, humidity) from major cities around the world. https://www.kaggle.com/datasets/guillemservera/global-daily-climate-data?select=countries.csv
- **Yield Data**: Our crop yield modeling is powered by the "Crop Yield Prediction Dataset" from Kaggle, which includes yield outcomes along with environmental features across multiple countries and years. https://www.kaggle.com/datasets/patelris/crop-yield-prediction-dataset?resource=download&select=yield_df.csv

While scaling our AI agent, we used the following data sets:

- **Soil Data**: We use global soil property estimates from SoilGrids, which provides key parameters like pH, clay content, and organic carbon, essential for assessing crop suitability.
  https://soilgrids.org/

- **Yield Data**: Yield data and yield gap estimates by region and crop, based on research plots and simulations https://www.yieldgap.org/India and https://code.earthengine.google.com/
- **Weather:** Provides agro-climatic parameters (e.g., temperature, rainfall, solar radiation, humidity) derived from satellite data and reanalysis models. https://power.larc.nasa.gov/ and https://code.earthengine.google.com/

5.2 Data processing:

MVP Phase – Weather and Yield Data Processing Using Spark

In the MVP stage, we built a simplified yet functional data pipeline using PySpark to process city-level weather and yield data for India. The purpose was to validate the feasibility of using climate data to predict crop yield before scaling globally.

Weather Data Processing:

We used the Global Daily Climate Data dataset from Kaggle, which includes daily weather records (temperature, precipitation, etc.) for major cities.

Using PySpark, we grouped the data by city and year and computed key seasonal features:

- Average of maximum temperature
- Average of minimum temperature
- Total rainfall during the growing period

The cleaned data was converted into a pandas DataFrame for further modeling.

Yield Prediction:

We trained a Feedforward Neural Network (FNN) using features such as rainfall, temperature, pesticide use, and crop type.

The model was evaluated using MSE, MAE, and R^2, and achieved the following performance on the validation set:

- MSE ≈ 2.01e+09
- MAE ≈ 31,336
- $R^2 \approx 0.72$

We also visualized predicted vs. actual yields to confirm model fit and identify areas for improvement.

This MVP pipeline demonstrated the viability of using limited data to make accurate yield predictions, paving the way for the scaled-up global version of AgriAid.

Then to scale it, we collected the soil, weather, and yield data from publicly available global sources, and undertook a series of preprocessing and transformation steps to convert them into a structured format. Each dataset was cleaned, standardized, and spatially aligned to enable integration into a unified training table for predictive modeling.
The following steps were taken.

Soil Data (SoilGrids GeoTIFFs)

Source: Global SoilGrids raster files (.tif) for variables like nitrogen, pH, clay, sand, bulk density, potassium, and silt
Processing Steps:

- Read each raster using rasterio, and reproject coordinates from EPSG:4326 to the raster CRS (e.g., ESRI:54052).
- Created a regular grid of (lat, lon) points.
- Sampled raster values at each grid point using raster.sample().
- Parallelized with PySpark for global scale (partitioned spatial grid and opened each raster once per worker).
- Final Output: Spark DataFrame with one row per (latitude, longitude) containing all soil variables.

Weather Data (ERA5 Monthly NetCDF)

Source: ERA5 monthly reanalysis datasets (.nc) from the Copernicus Climate Data Store
Key Variables used:

- 2m_temperature, total_precipitation, surface_net_solar_radiation, evaporation, soil_temperature_level_1

Processing Steps:

- Downloaded .nc files per year using cdsapi.
- Extracted and averaged data over time (yearly average).
- Removed unused dimensions (expver, level, step, etc.).
- Converted units (e.g., Kelvin to °C, J/m² to MJ/m²).
- Rounded coordinates to avoid float precision issues.
- Dropped duplicates based on (latitude, longitude, year).
- Final Output: Cleaned CSV with one row per (latitude, longitude, year) with weather features.

Yield Data (SPAM Global CSVs)

Source: SPAM yield data by crop and irrigation type (.csv files per year/tag)
Processing Steps:

- Read spam[year]V1r0_global_Y_[tag].csv files (for TA, TI, TR irrigation tags).

- Extracted yield columns for selected crops (Wheat, Rice, Maize, etc.).
- Renamed irrigation types (TA → combined, TI → irrigated, TR → rainfed).
- Normalized schema: (latitude, longitude, crop_type, irrigation_method, yield, year)
- Filtered out NaN and zero yield values.
- Final Output: Normalized yield DataFrame for all crops and irrigation types globally.

Combining all Sources for XGBoost Training

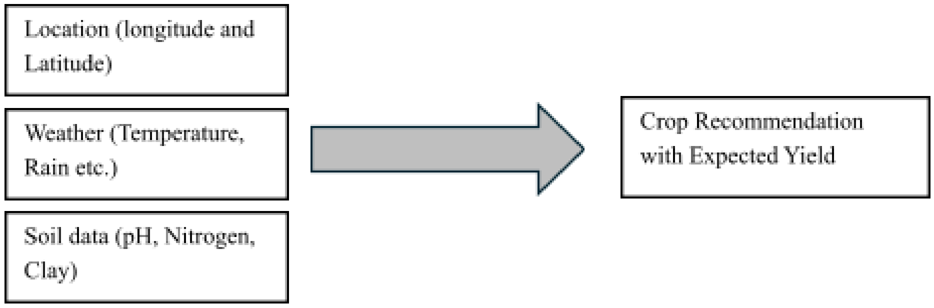
- Joined all three datasets on:
    - (latitude, longitude) for soil data
    - (latitude, longitude, year) for weather and yield data
- Ensured matching coordinate precision by rounding
- Removed rows with missing values post-join
- Encoded categorical features (crop_type, irrigation_method) using one-hot encoding
- Final feature table used for model training

These datasets provide rich, structured data that enable us to generate personalized crop and yield recommendations without designing or collecting our own data from scratch.

5.3 Outcome and Target Variables

Our analysis focuses on two primary target variables: crop yield, a continuous variable representing expected production per unit area, and recommended crop type**,** a categorical variable selected based on predicted yield and environmental suitability.

The explanatory variables used to generate these predictions include a range of soil parameters (such as nitrogen content, clay percentage, pH, soil humidity, and organic carbon), weather variables (including average temperature, precipitation, rainfall, and humidity), and geographic coordinates (latitude and longitude) to localize environmental conditions. Together, these features enable us to deliver personalized, data-driven crop recommendations tailored to each farmer's specific location.


Location (longitude and Latitude)
Weather (Temperature, Rain etc.)
Soil data (pH, Nitrogen, Clay)
Crop Recommendation with Expected Yield

5.4 Variation in Data Supporting Analytics

Our data exhibits substantial variation across both the target and explanatory variables, which is essential for addressing our core question: **Which crops are most suitable for a specific location, and what yields can be expected?** The target variable, crop yield, varies significantly by crop type, region, and environmental conditions, reflecting real-world differences in productivity. Our features also show strong variation. Soil parameters such as nitrogen content, clay percentage, and pH differ across geographies, while weather variables like precipitation, humidity, and temperature fluctuate over time and location. Additionally, the geographic coordinates (latitude and longitude) introduce spatial variation that enables highly localized predictions. This diversity in the data allows our model to learn meaningful patterns and generate accurate, location-specific crop recommendations and yield forecasts.

Analytical Framing: Explanatory vs. Predictive

We began by ensuring a clear understanding of the data sources and what each variable represents. Soil data was sourced from SoilGrids (global interpolated raster layers), weather data from ERA5 reanalysis (satellite-modeled and bias-corrected), and crop yield data from SPAM (gridded estimates derived from subnational agricultural surveys). Each dataset was validated for consistency in units, spatial resolution, coordinate systems, and temporal coverage.

While certain variables—such as precipitation or nitrogen—are biologically linked to crop outcomes, our model was not designed for causal inference. The goal was not to estimate the causal effect of a single variable but to predict yield outcomes given a realistic set of agro-climatic conditions. Some features, such as irrigation type or nutrient proxies, are actionable, allowing for targeted interventions, while others (e.g., temperature) are not directly manipulable but remain useful for guiding adaptive strategies like crop planning.

Our focus was firmly predictive. The objective was to build an accurate, generalizable model that could estimate crop yields across regions and conditions, empowering decision-makers in agriculture and food security. We prioritized data-driven models like XGBoost for their superior performance on structured data, while also using feature importance metrics to extract limited explanatory value. Interpretability was secondary to the model's predictive strength.

5.5 Impediments

While our analytics initiative has strong potential for impact, it is not without challenges. Below, we outline the key impediments we have faced and the strategies we've used to overcome them.

- Data Challenges

One of the primary hurdles in executing our analytics was dealing with heterogeneous and inconsistent data sources. We integrated data from multiple platforms, some temporal (e.g., historical weather data) and some static (e.g., soil composition), and had to standardize them to a unified format for analysis. This required careful alignment across location, time, and units of measurement.

We also encountered missing values, inconsistent formats, and sparse data in certain geographic regions, which posed a risk to model reliability. To address these challenges, we performed extensive data cleaning, applied imputation techniques to fill in missing values, and used external APIs (e.g., SoilGrids, ERA5) to enrich the dataset where local data was unavailable or unreliable.

- Modeling and Feature Engineering Challenges

Another major impediment was handling the mix of categorical and continuous variables. Features like crop type, irrigation method, and location category needed to be encoded properly (e.g., one-hot encoding or embeddings) to be usable in machine learning models. Ensuring consistency across encoding methods and avoiding data leakage during this process added technical complexity.

In addition, the target variable (crop yield) exhibited a wide range of values across regions and crops, making it harder for the model to converge during training. To stabilize training and improve accuracy, we applied normalization techniques to the target variable.

- Computational Limitations

We also faced resource constraints, particularly around GPU availability. Due to these limitations, we were unable to experiment with more computationally intensive models such as transformers or deep feedforward neural networks. Instead, we focused on more efficient models like XGBoost and TabNet, balancing performance with feasibility.

6. Executing the Analytics

6.1 Team Roles and Responsibilities for Executing Analytics

We are a team of five, each contributing specialized expertise across the goal, from data collection and preparation to model development, interface design, and project coordination. Our team works collaboratively across roles, with several members contributing to multiple areas of the project. The table below outlines each team member's primary responsibilities and areas of collaboration.

Team Member(s)	Primary Role	Key Responsibilities
Avi & Elsiey	Data Collection Lead / Data Engineer	Acquire and integrate soil, weather, and yield data; manage ETL workflows, raster-to-CSV conversions, and API connections
Peilu & Mahnoor	Data Scientists / ML Engineers	Build and tune predictive models; conduct feature engineering and evaluate performance using R^2, RMSE, and crop ranking accuracy.

Elsiey & Avi	Project Coordinator / Research Analyst	Oversee project timelines, synthesize model findings, define success metrics, and align analytics with business goals.
Mahnoor , Avi, Elsiey, Peilu, Susie	Business Analyst	Presentation and Report
Susie & Mahnoor	Product and Frontend Developers	Develop the user interface to deliver crop recommendations and yield predictions in a clear format.

Table 1.0 Team Roles and Responsibilities

6.2 Team Collaboration in Defining Metrics and Analytics Strategy

We involve all team members in defining success metrics and shaping the analytics strategy through regular collaborative planning sessions. Each member brings domain-specific insights, whether it's data quality, model performance, user experience, or project goals, to ensure that metrics are both technically sound and practically meaningful. Together, we review early outputs, discuss trade-offs, and refine metrics such as prediction accuracy, usability, and relevance of crop recommendations to ensure alignment with real-world impact. This cross-functional collaboration ensures our analytics are grounded in shared understanding and optimized for end-user value.

7. Implementation

7.1 Model Training

7.1.1 Input Features

For the MVP we did the following model training

Modeling Approach for MVP:

- We employed data-driven predictive models, specifically Feedforward Neural Networks (FNN), to capture complex, nonlinear relationships among climate factors, soil properties, and crop yield.
- Additionally, we used Prophet models for time-series forecasting of weather conditions, integrating historical trends to enhance predictive accuracy.
- Models were rigorously validated using cross-validation techniques and key metrics (MSE, R^2) to ensure robust predictive performance.

Statistical Evaluation of Results:

- The trained FNN model demonstrated robust predictive performance, achieving stable evaluation metrics across a 5-fold cross-validation.
- Statistical analyses further confirmed that climatic factors, notably precipitation and average temperature, significantly influence crop yields.

Cross-Validation and Performance Metrics:

- Conducted 5-fold cross-validation, achieving stable predictive performance with the following average metrics:

  Mean MSE: 2,013,497,497.60

  Mean MAE: 31,336.26

  Mean R^2: 0.72

- Individual fold results:

  Fold 1: MSE = 1,811,999,104, MAE = 30,070, $R^2 = 0.75$

  Fold 2: MSE = 2,138,616,704, MAE = 31,878, $R^2 = 0.71$

  Fold 3: MSE = 2,047,384,576, MAE = 31,639, $R^2 = 0.71$

  Fold 4: MSE = 1,887,906,560, MAE = 30,538, $R^2 = 0.73$

  Fold 5: MSE = 2,181,580,544, MAE = 32,557, $R^2 = 0.70$

These metrics confirm the model's robust predictive capabilities, ensuring reliability and practical utility in decision-making scenarios.

Modeling Approach for Scaled Version

Then to scale it we used the global data sets as stated above to develop a supervised learning model to predict crop yield (in tons/hectare) based on geospatial, agronomic, and climatic factors across multiple crop types and irrigation practices.

We started off by constructing a rich feature set by integrating multi-source geospatial data spanning soil properties, climatic conditions, and agricultural yields.

- Soil variables were extracted using rasterio, PySpark with vectorized sampling across global lat/lon grid
  - ▪ Features: nitrogen, phWater, clay, sand, bulk_density, potassium, silt
- Weather Variables (from ERA5 monthly NetCDF files via CDSAPI) extracted using xarray, averaged annually
  - ▪ Variables used: 2m_temperature, total_precipitation, surface_net_solar_radiation, evaporation, soil_temperature_level_1

    - Converted Kelvin → °C, J/m² → MJ/m²
    - PySpark was used to scale the transformation pipeline across multiple years and reduce memory bottlenecks when handling global gridded files.
- Yield and Agronomic Variables (from SPAM global CSVs):
    - Features: crop_type, irrigation_method, year, yield
    - Extracted and normalized using pandas, joined across tags TA, TI, TR

7.1.2 Preprocessing Steps

The following preprocessing steps were followed:

- Merged soil, weather, and yield datasets on (latitude, longitude, year) using PySpark DataFrames.
- One-hot encoded categorical features (crop_type, irrigation_method) using Spark SQL functions.
- Handled missing/null values and dropped duplicates using dropna() and dropDuplicates() in Spark.
- Exported final feature table using .coalesce(1).write.csv(...) for downstream modeling.
- Dropped rows with missing values in any key feature
- One-hot encoded categorical variables:
    1. crop_type (e.g., Wheat, Maize, Sugarcane)
    2. irrigation_method (combined, irrigated, rainfed)
- Normalized continuous inputs for TabNet using standard scaling
- Ensured spatial alignment by rounding lat/lon coordinates and deduplicating

7.1.3 Modeling Frameworks and Tools

We used a combination of tools and libraries tailored to each stage of the modeling pipeline. For data manipulation, we efficiently leveraged Pandas, xarray, rasterio, pyproj, and PySpark to handle large-scale tabular, geospatial, and raster data. For modeling, we trained two separate models, XGBoost, implemented using the xgboost Python library, and TabNet, built using the pytorch-tabnet library with a PyTorch backend. XGBoost was trained on CPU due to its lightweight and optimized implementation for structured data, while TabNet training was run on GPU to accelerate the deep learning computations and handle batch-based learning effectively. This tech stack allowed us to combine scalable data engineering with robust model development.

Task	Tool / Library Used
Data manipulation	Pandas, xarray, rasterio, pyproj, Pyspark
Model 1: XGBoost	xgboost Python library
Model 2: TabNet	pytorch-tabnet (PyTorch backend)
Hardware	CPU for XGBoost, GPU for TabNet

Table 2.0 Modelling Tools and Framework

7.1.4 Train-Test Split Strategy

Our strategy followed the following approach: we performed an 80/20 train-test split stratified by crop_type and year, and ensured that no (latitude, longitude, year) combinations were shared between the training and test sets to avoid data leakage.

7.1.5 Model Configuration & Training

We trained and fine-tuned both XGBoost and TabNet models on the unified dataset, using tailored configurations suited to each algorithm. Training included early stopping, hyperparameter tuning, and evaluation based on RMSE and R^2 to ensure robust performance.

1. XGBoost (Tree-based Gradient Boosting)

- Objective: "reg: squarederror"
- Hyperparameters:
    - max_depth = 6, eta = 0.1, n_estimators = 500
    - early_stopping_rounds = 20 on validation RMSE
- Cross-validation: 5-fold CV stratified by crop type
- Feature importance: gain and SHAP values used for interpretation

2. TabNet (Deep Learning on Tabular Data)

- Framework: PyTorch TabNet
- Training details:
    - Optimizer: Adam
    - Learning rate: 2e-2, batch size: 1024
    - Loss: Mean Squared Error (MSE)
    - Scheduler: ReduceLROnPlateau
    - Early stopping: patience = 15
- Requires normalized inputs, benefits from categorical embeddings

7.2 Model results

In our crop yield prediction task, we trained two models, XGBoost and TabNet using integrated soil, weather, and crop data. The results showed that the XGBoost achieved superior performance, with a Root Mean Squared Error (RMSE) of 11,113 and an R^2 score of 0.801, indicating strong predictive accuracy and good fit to the data. In contrast, TabNet recorded a higher RMSE of 14,833 and a lower R^2 of 0.646, suggesting weaker generalization and greater prediction error. XGBoost's tree-based architecture proved to be more effective for structured, tabular data and required less tuning, whereas TabNet's deep learning approach was more sensitive to hyperparameters and less suited for the available data scale without GPU acceleration. Overall, XGBoost emerged as the more robust and reliable model for this agricultural use case. Table 3.0 shows results summary.

Model	RMSE ↓	R² ↑	Summary
XGBoost	11,113	0.801	Best overall performance, strong generalization, and lower error
TabNet	14,833	0.646	Weaker fit, more sensitive to training conditions and hyperparameters

Table 3.0 Model Results Summary

7.2.1 Model Evaluation Visuals

The chart below displays the residuals (difference between actual and predicted yield) on the vertical axis against predicted yield on the horizontal axis for the XGBoost model. Ideally, residuals should be randomly scattered around zero with no clear pattern, indicating that the model captures the relationship well without systematic bias. In our case, the spread around zero suggests reasonable accuracy, though some larger errors exist in higher predicted ranges, likely due to yield outliers or data sparsity.

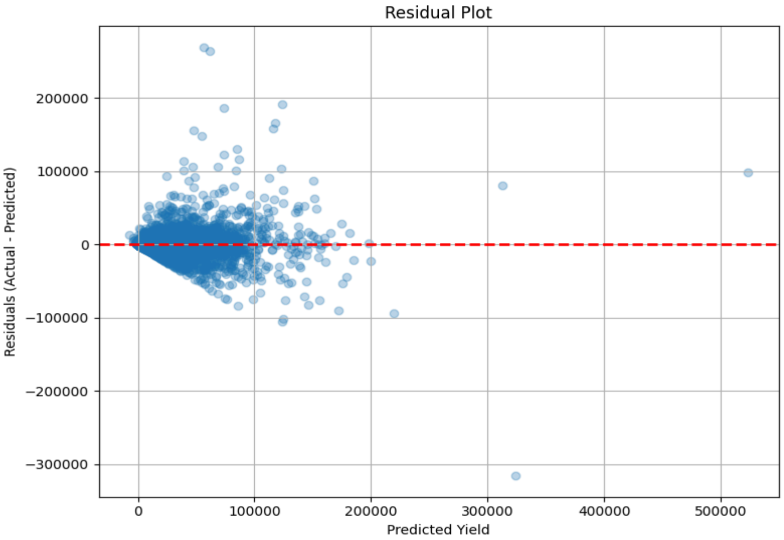


Fig 1.0 Residual and Prediction Analysis

Predicted vs Actual Yield Plot:

This scatter plot compares the predicted crop yields against actual observed yields, with the red dashed line representing perfect predictions (i.e., predicted = actual) for the XGBoost Model. Most points lie close to this diagonal line, indicating that the model performs well overall, especially for typical yield values. Deviations from the line highlight under- or over-predictions, which tend to increase slightly in extreme yield cases.

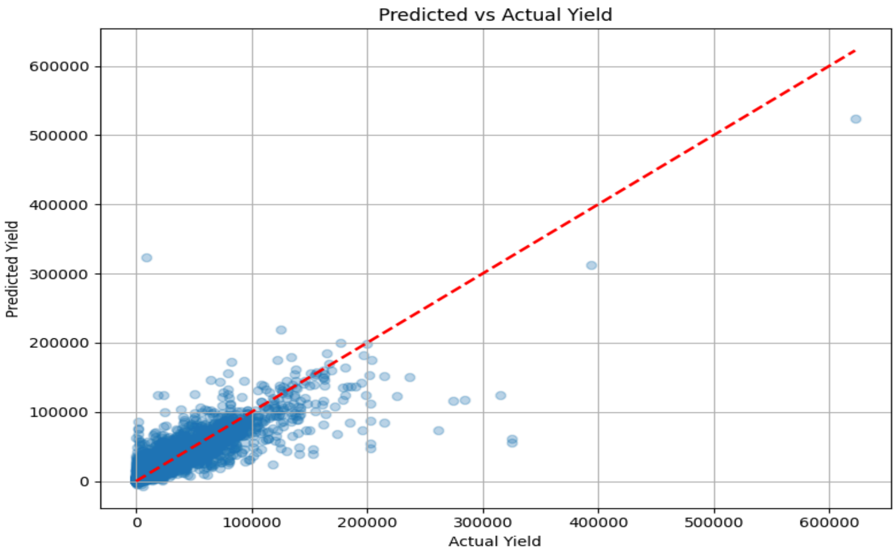


Fig 2.0 Predicted vs Actual Yield Analysis

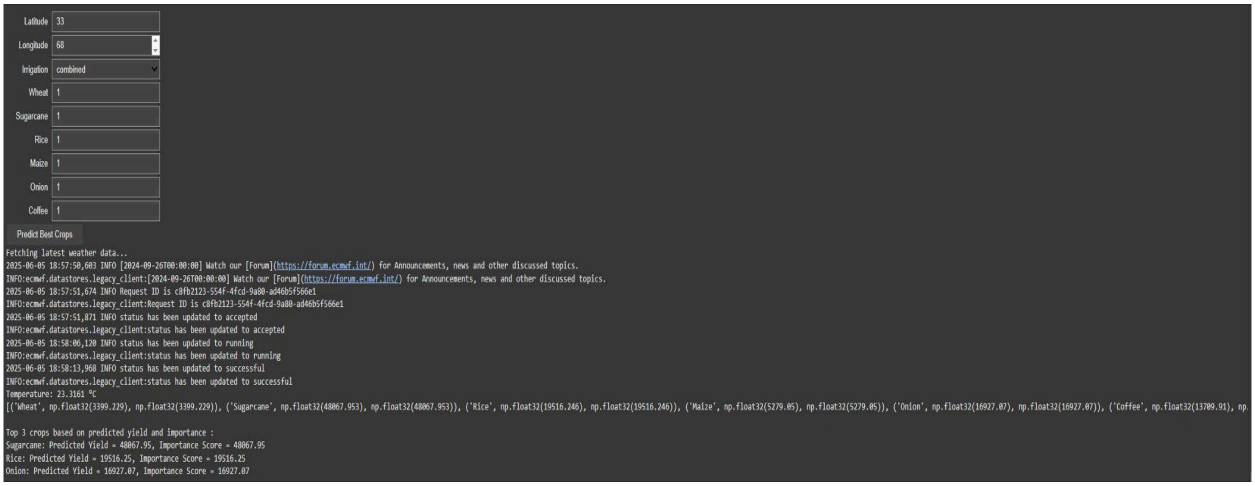
7.3 Decision Impact and Actionable Insights

Once we have the results, the analytics will directly influence which crops smallholder farmers choose to plant and how they allocate their limited resources (land, water, labor). Rather than relying on intuition or tradition, farmers will now receive data-backed recommendations for the top three crops best suited to their specific location, along with expected yield estimates. Because of the analytics developed, farmers will be able to make proactive and informed decisions, even under uncertain weather or soil conditions. This shifts their behavior from reactive trial-and-error farming to strategic, evidence-based planning, ultimately leading to better crop outcomes, reduced risk, and improved livelihoods. Additionally, our team will use these insights to refine and personalize future recommendations as more data is collected.

Additionally, the analytics will support risk mitigation. Early identification of potential low-yield periods helps in creating contingency plans, such as activating crop insurance or making strategic adjustments in the agricultural supply chain. These insights enable both farmers and stakeholders to prepare and respond more effectively to agronomic and climate-related risks.

7.4 UI interface

Below is the final user interface we developed for the AgriAid agent.



Latitude 33
Longitude 68
Irrigation combined
Wheat 1
Sugarcane 1
Rice 1
Maize 1
Onion 1
Coffee 1

```
Fetching latest weather data...
2025-06-05 18:57:50,603 INFO [2024-09-26T00:00:00] Watch our [Forum](https://forum.ecmwf.int/) for Announcements, news and other discussed topics.
INFO:ecmwf.datastores.legacy_client:[2024-09-26T00:00:00] Watch our [Forum](https://forum.ecmwf.int/) for Announcements, news and other discussed topics.
2025-06-05 18:57:51,674 INFO Request ID is c8fb2123-554f-4fcd-9a80-ad46b5f566e1
INFO:ecmwf.datastores.legacy_client:Request ID is c8fb2123-554f-4fcd-9a80-ad46b5f566e1
2025-06-05 18:57:51,871 INFO status has been updated to accepted
INFO:ecmwf.datastores.legacy_client:status has been updated to accepted
2025-06-05 18:58:06,120 INFO status has been updated to running
INFO:ecmwf.datastores.legacy_client:status has been updated to running
2025-06-05 18:58:13,968 INFO status has been updated to successful
INFO:ecmwf.datastores.legacy_client:status has been updated to successful
Temperature: 23.3161 °C
[('Wheat', np.float32(3399.229), np.float32(3399.229)), ('Sugarcane', np.float32(48067.953), np.float32(48067.953)), ('Rice', np.float32(19516.246), np.float32(19516.246)), ('Maize', np.float32(5279.05), np.float32(5279.05)), ('Onion', np.float32(16927.07), np.float32(16927.07)), ('Coffee', np.float32(13709.91), np.

Top 3 crops based on predicted yield and importance :
Sugarcane: Predicted Yield = 48067.95, Importance Score = 48067.95
Rice: Predicted Yield = 19516.25, Importance Score = 19516.25
Onion: Predicted Yield = 16927.07, Importance Score = 16927.07
```

Fig 3.0 UI interface

7.5 Embedding Analytics into Workflows for Effective Adoption

For the AgriAid project we designed the analytics pipeline to seamlessly integrate into the decision-making workflow of smallholder farmers. Initially, we built a lightweight MVP that used localized weather and soil data for yield prediction. Once validated, we scaled the solution globally using PySpark to process real-time satellite and geolocation data. To ensure adoption, we focused on embedding analytics outputs into an intuitive UI that allowed users to input farm-level coordinates and receive actionable recommendations.

We prioritized accessibility by supporting low-tech environments and multilingual interfaces, aligning our analytics outputs with how users already plan crops seasonally and locally. Furthermore, our integration strategy includes farmer feedback loops and local extension worker partnerships to validate model recommendations and improve trust and usability over time. By embedding analytics into a familiar workflow and co-designing with users, we significantly improve the chances of real-world adoption and impact.

8. Scale

8.1 Organizational Challenges to Scaling and Their Solutions

Calling our analytics-driven initiative involves a range of organizational challenges across data availability, technical infrastructure, team capacity, and user adoption culture.

- **Data Challenges**: As we expand to new geographies, we anticipate encountering incomplete, inconsistent, or lower-resolution soil and weather data, which may affect model performance. To mitigate this, we plan to build partnerships with global data providers, integrate satellite-derived datasets, and implement dynamic data quality checks during ingestion.
- **Systems and Infrastructure**: Our current solution is designed to run efficiently on local computers, but to support real-time, global scalability, we will need to transition to a more cloud-native architecture. This includes using services like Google Cloud Platform

or AWS for scalable storage and compute, along with containerization (e.g., Docker) for flexible deployment.

- **People and Team Capacity**: While our five-member team is agile and collaborative, sustained scale-up will require dedicated roles for DevOps, data governance, user support, and localization. We plan to expand the team gradually and consider training agricultural extension workers to serve as frontline interpreters of the tool for farmers.
- **Culture and Adoption**: In many regions, farmers rely heavily on experience or peer advice. To promote adoption, we are designing the system to embed seamlessly into farmers' current decision-making workflows, using SMS-style communication, local language support, and intuitive interfaces. We also aim to partner with agricultural NGOs and cooperatives to build trust and encourage usage.

8.2 Improvements

Our analytics initiative is designed as a continuous learning system, not a one-time deployment. As we scale, we will implement a structured plan for ongoing improvement across three key dimensions: model performance, user experience, and business impact.

1. **Model Refinement**: We will continuously monitor model performance using live data and regularly retrain models with newly collected soil, weather, and yield information. As more farmers use the tool, the growing dataset will help us improve prediction accuracy and regional adaptability.
2. **User Feedback Integration**: We plan to integrate simple feedback loops into the user interface (e.g., "Was this recommendation helpful?"). These responses, along with qualitative insights from partner organizations, will inform enhancements to our recommendation logic and prioritization of new features.
3. **Feature Expansion**: Based on user demand and agricultural trends, we aim to add new capabilities such as seasonal risk alerts (e.g., early flood or drought warnings), crop price forecasting and localized planting schedules and input usage advice.
4. **Impact Evaluation**: We will establish internal metrics dashboards to track yield improvements, adoption rates, and user engagement over time. This will guide data-driven decisions on where to expand, what to refine, and how to maximize value for farmers.

By embedding analytics into an iterative, user-centered development process, we ensure that the platform stays relevant, accurate, and continuously aligned with the real needs of smallholder farmers.

9. Conclusion

For this project we developed AgriAid, a data-driven tool that provides smallholder farmers with personalized crop recommendations and yield predictions based on their location, soil conditions, and weather patterns. By integrating global datasets and building scalable models using PySpark, XGBoost, and TabNet, we created a reliable system capable of supporting data-informed farming decisions.

AgriAid transforms complex environmental data into simple, actionable insights through an intuitive interface, enabling farmers to improve planning, reduce risk, and boost yields. With strong model performance and a scalable architecture, our solution is well-positioned to expand globally and deliver real-world impact for farmers in diverse regions.

Appendix: Reference to Code and Notebooks

For full transparency and reproducibility of our work, the following Jupyter notebooks contain the complete implementation of each phase of the project. These are located in the Group11-Big-Data-Project directory and are organized by task-specific folders such as Model, Dataset, and others related to data processing, training, and evaluation.

- **Data Sets:**The main project directory for Group 11 includes a folder named Dataset, which contains four subfolders**,** YieldData-SPAM, SoilGrids-ti, ERA5-GlobalWeatherData and ProcessedDataSets. The first three folders represent the raw datasets used in the project, covering yield, soil, and weather data respectively. The fourth folder, ProcessedDataSets, contains the cleaned and merged version of the raw data used for modeling and analysis**.**
- **Data Processing**
  All data extraction, cleaning, transformation, and integration steps are documented in: Data_Processing.ipynb Data processing.ipynb
- **Model Training & Evaluation**
  All model development, training, hyperparameter tuning, and evaluation metrics are detailed in: Training_Global.ipynb Training-Global.pynb
- **Model Artifacts:** The folder contains all the output files generated from training our machine learning models. This folder includes: Trained TabNet and XGBoost model files (.joblib, .zip, .json)
- **User Interface Development**
  The final AgriAid recommendation interface, including the coordinate-based input system and output rendering, is implemented in UI.ipynb UI.ipynb
- **MVP-Yield Prediction** for the minimum viable product tested at the first phase: data cleaning, transformation, FNN modeling, OpenAI API accessing process are documented in: MVP_yield_prediction.ipynb
- **MVP-Weather & Rain Prediction** for minimum viable product for rain predictions MVP_Weather&Rain.ipynb