

# Technical Narrative: Holistic Farmer Trust Scoring (HFTS)

## 1 Introduction

The Holistic Farmer Trust Scoring (HFTS) system is a machine learning-powered platform designed to assess the creditworthiness of smallholder farmers in Kenya. By leveraging alternative data sources, the system provides a more inclusive and accurate risk assessment than traditional credit scoring models, which often fail to serve the unbanked and underbanked agricultural sector.

## 2 System Architecture

The HFTS system is built on a modern, scalable architecture that consists of a FastAPI backend and a Next.js frontend.

### 2.1 Backend

The backend is a Python-based application powered by FastAPI, a high-performance web framework. The core of the backend is the `scoring` module, which contains the LightGBM model and the logic for feature engineering and risk classification.

- **API:** A RESTful API provides endpoints for scoring farmers, generating reports, and system health checks.
- **Model:** A pre-trained LightGBM model is loaded at startup for real-time predictions.
- **Data Models:** Pydantic models are used for data validation and serialization.
- **Report Generation:** The `report_generator` module creates professional PDF credit assessment reports.

### 2.2 Frontend

The frontend is a single-page application (SPA) built with Next.js, a popular React framework. It provides a user-friendly interface for inputting farmer data and viewing the resulting trust score.

- **Components:** The UI is built with reusable React components, including a form for data entry and a results page to display the score and risk analysis.
- **State Management:** React hooks are used for managing component state.
- **Styling:** Tailwind CSS is used for styling the application.

### 3 Machine Learning Pipeline

The machine learning pipeline is a critical component of the HFTS system. It consists of the following stages:

- **Data Generation:** A synthetic dataset of 10,000 farmer profiles is generated to train and evaluate the model.
- **Feature Engineering:** Raw data is transformed into meaningful features using the three-tiered weighting system.
- **Model Training:** The LightGBM model is trained on the engineered features to predict the probability of default.
- **Model Evaluation:** The model's performance is evaluated using a variety of metrics, including AUC-ROC, precision, and recall.

### 4 Deployment

The HFTS system is designed for easy deployment using Docker.

- **Backend:** The backend can be containerized and deployed as a standalone service.
- **Frontend:** The frontend can be built and served as a static site or deployed as a containerized application.

### 5 Testing

The system includes a suite of tests to ensure its quality and reliability.

- **Backend:** Pytest is used for unit testing the backend components.
- **API:** The API can be tested using tools like `curl` or Postman.