E Mukoko — Predictive Models, Inputs & Graphs Specification

Document ID: EMK-ML-2026-001
Version: 1.0
Date: February 18, 2026
Status: Technical Specification

Table of Contents

---

1. Overview

The E Mukoko platform uses AI/ML models across three domains:

Domain	Models	Purpose
Hive Health	Pest Prediction, Health Scoring, Swarm Prediction, Maintenance Scheduling	Protect colonies and reduce losses
Production	Yield Forecasting, Quality Prediction, Weight Anomaly Detection	Maximise output and ensure integrity
Market	Demand Forecasting	Optimise pricing and supply planning

All models consume data from the IoT Smart Hive sensors (temperature, humidity, weight, vibration). Models are served via a REST API at `/api/ai/*` endpoints.

---

2. Model 1: Pest & Disease Prediction (Sensor-Based)

SRS Reference: FR-3.03, FR-2.06

Purpose

Predict pest infestations (Varroa mites, wax moths, hive beetles) and diseases (American Foulbrood, European Foulbrood, Nosema, chalkbrood) using IoT sensor data patterns — **no cameras or images required**. The model correlates environmental conditions, hive weight changes, vibration anomalies, and temperature/humidity deviations with known pest and disease onset patterns.

Model Type

- **Architecture:** Gradient Boosted Trees (XGBoost / LightGBM) for tabular feature classification; optional LSTM layer for temporal pattern detection
- **Framework:** Scikit-learn / XGBoost / LightGBM (server-side); ONNX Runtime for edge inference
- **Approach:** Supervised learning trained on historical sensor readings labelled with confirmed pest/disease events. Feature engineering extracts statistical summaries (rolling means, variances, rate-of-change) from raw time-series sensor data.

Inputs

Input	Type	Source	Window
Internal temperature	Float[] (time series, °C)	IoT sensor	Last 14 days (sampled every 30 min = 672 points)
External temperature	Float[] (time series, °C)	IoT sensor	Last 14 days
Internal humidity	Float[] (time series, %)	IoT sensor	Last 14 days
External humidity	Float[] (time series, %)	IoT sensor	Last 14 days
Hive weight	Float[] (time series, kg)	IoT sensor	Last 14 days
Vibration intensity	Float[] (time series)	IoT sensor	Last 14 days
Vibration frequency spectrum	Float[] (frequency bins)	IoT sensor	Last 7 days (aggregated)

Weight rate-of-change	Float (kg/day)	Computed	Last 7-day slope
Temperature deviation from brood norm (35°C)	Float (°C)	Computed	Last 7-day average
Humidity deviation from optimal (50-60%)	Float (%)	Computed	Last 7-day average
Season / month	Float (0–1, normalised)	System clock	Current
Geographic region	Enum	Database	Static per hive
Hive age (days since install)	Integer	Database	Current
Previous pest/disease history	Boolean[]	Database	Per pest/disease type
Hive ID	String	System context	Per prediction
Timestamp	ISO 8601	System clock	Per prediction

Engineered Features (extracted from raw inputs)

Feature	Derivation
`temp_rolling_mean_24h`	24-hour rolling average of internal temp
`temp_rolling_std_24h`	24-hour rolling standard deviation of internal temp
`temp_rate_of_change`	Linear slope of temperature over last 48h
`humidity_deviation`	Mean deviation from 50-60% optimal range
`weight_daily_delta`	Day-over-day weight change
`weight_7d_trend`	7-day linear regression slope of weight
`vibration_energy`	RMS of vibration signal over 24h window
`vibration_frequency_peak`	Dominant frequency from FFT analysis
`temp_humidity_ratio`	Internal temp / internal humidity
`night_day_temp_range`	Difference between max daytime and min nighttime temps

Outputs

Output	Type	Description
Prediction class	Enum	`varroa_mite`, `wax_moth`, `hive_beetle`, `foulbrood`, `nosema`, `chalkbrood`, `healthy`
Risk probability	Float (0–1)	Predicted probability of pest/disease occurrence within the next 7 days
Severity level	Enum	`low`, `medium`, `high`, `critical` — derived from probability + trend direction
Contributing factors	Array of {feature, importance}	Top sensor features driving the prediction (SHAP values)
Recommendation	String	AI-generated action plan based on prediction
Predicted onset window	String	Estimated timeframe (e.g., "3–5 days") if risk is elevated

Alert Thresholds

Condition	Severity	Action
Risk probability ≥ 0.85	Critical	Immediate push notification + SMS
Risk probability ≥ 0.70	High	In-app alert within 1 hour
Risk probability ≥ 0.50	Medium	Dashboard flag, next inspection reminder
Risk probability < 0.50	Low	Logged for review, no alert

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Pest Risk Timeline	Area chart	Date	Risk probability (0–1) colour-coded by pest type	Track predicted pest/disease risk over time per hive
Pest Type Distribution	Donut / Pie chart	—	Count by predicted pest type	Show which pests are most frequently predicted
Risk Factor Breakdown	Horizontal bar chart	SHAP importance value	Sensor feature name	Explain which sensor readings are driving the current prediction
Monthly Alert Frequency	Stacked bar chart	Month	Alert count (stacked by type)	Seasonal pest trends
Sensor Correlation Heatmap	Heatmap	Sensor features	Sensor features	Show correlations between sensor readings and pest risk
Prediction Accuracy Over Time	Line chart	Month	Accuracy % (confirmed / predicted)	Model drift monitoring

3. Model 2: Swarm Prediction

SRS Reference: FR-3.04

Purpose

Predict swarming events 48–72 hours in advance using vibration patterns, weight changes, and temperature anomalies, giving farmers time to intervene.

Model Type

- **Architecture:** LSTM (Long Short-Term Memory) or Temporal Convolutional Network (TCN)
- **Framework:** TensorFlow / PyTorch
- **Training data:** Historical sensor readings labelled with known swarm events

Inputs

Input	Type	Source	Window
Vibration readings	Float[] (time series)	IoT sensor	Last 7 days (sampled every 30 min = 336 data points)
Internal temperature	Float[] (time series)	IoT sensor	Last 7 days
External temperature	Float[] (time series)	IoT sensor	Last 7 days
Humidity readings	Float[] (time series)	IoT sensor	Last 7 days
Weight readings	Float[] (time series)	IoT sensor	Last 7 days
Time of year	Float (0–1, normalised month)	System clock	Current
Hive age (days since install)	Integer	Database	Current
Colony history	Boolean	Database	Has this hive swarmed before?

Outputs

Output	Type	Description
Swarm probability	Float (0–1)	Likelihood of swarming within 72 hours
Estimated timeframe	String	"24–48 hours" or "48–72 hours"
Key contributing factor	String	Which sensor(s) contributed most to prediction
Confidence	Float (0–1)	Model certainty

Decision Thresholds

Probability	Action
≥ 0.80	**Critical alert** — Inspect immediately, check for queen cells
0.60–0.79	**Warning** — Schedule inspection within 24 hours
0.40–0.59	**Watch** — Monitor vibration closely over next 48 hours
< 0.40	Normal — No action required

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Vibration Pattern Analysis	Line chart (multi-line)	Time (7 days)	Vibration amplitude	Show pre-swarm vibration spikes vs. normal pattern
Swarm Probability Gauge	Radial gauge	—	0–100%	Real-time swarm risk indicator on hive detail page
Swarm Risk Over Time	Area chart	Date (30 days)	Probability (0–1)	Track how swarm risk builds over weeks
Contributing Factors Radar	Radar / Spider chart	Sensor axes (temp, humidity, vibration, weight, time)	Normalised contribution (0–1)	Explainability — which inputs drive the prediction

---

4. Model 3: Harvest Yield Forecasting

SRS Reference: FR-2.03, FR-3.10

Purpose

Predict expected honey yield (kg) for each hive over the next 30, 60, and 90 days based on weight trends, seasonal patterns, and historical production.

Model Type

- **Architecture:** Gradient Boosted Trees (XGBoost / LightGBM) or Prophet (time-series decomposition)
- **Framework:** scikit-learn / Facebook Prophet
- **Training data:** Historical weight curves mapped to actual harvest weights

Inputs

Input	Type	Source	Description
Weight time series	Float[]	IoT sensor	Last 90 days of daily weight readings
Weight rate of change	Float	Computed	kg/day average over last 14 days
Season/month	Integer (1–12)	System clock	Seasonal nectar flow indicator
Hive location	Categorical	Database	Region affects flora availability
Historical yields	Float[]	Database	Past harvests (kg) for this hive
Temperature rolling avg	Float	IoT sensor	14-day rolling average temp
Humidity rolling avg	Float	IoT sensor	14-day rolling average humidity
Colony age	Integer (days)	Database	Older colonies may produce differently
Current hive weight	Float (kg)	IoT sensor	Latest reading

Outputs

Output	Type	Description
Predicted yield (30d)	Float (kg)	Expected harvestable honey in 30 days
Predicted yield (60d)	Float (kg)	Expected harvestable honey in 60 days
Predicted yield (90d)	Float (kg)	Expected harvestable honey in 90 days
Confidence interval	[Float, Float]	80% prediction interval (lower, upper bound)
Optimal harvest date	Date	Suggested date for maximum yield
Revenue estimate	Float ($)	Predicted yield × current average market price

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Weight Trend + Forecast	Line chart (solid + dashed)	Date	Weight (kg)	Historical weight with projected curve and confidence band
Yield Forecast Cards	Stat cards	—	30d / 60d / 90d yield (kg)	Quick summary on dashboard overview
Seasonal Yield Comparison	Grouped bar chart	Season (Spring/Summer/Autumn)	Yield (kg) per hive	Compare productivity across seasons
Fleet Yield Projection	Stacked area chart	Date (90 days)	Total kg across all hives	Farm-wide production forecast
Revenue Forecast	Line chart with $ overlay	Date	Estimated revenue ($)	Monetary projection for the farmer

---

5. Model 4: Hive Health Scoring

SRS Reference: FR-2.03

Purpose

Generate a single composite health score (0–100) for each hive by combining all sensor data, alert history, and AI diagnostics into one actionable metric.

Model Type

- **Architecture:** Weighted scoring algorithm with ML-learned weights, or Random Forest classifier mapping sensor combinations to health categories
- **Framework:** scikit-learn
- **Approach:** Supervised — labelled with expert beekeeper health assessments

Inputs

Input	Type	Weight (approx)	Source
Internal temperature	Float (°C)	20%	IoT sensor
Temperature deviation from optimal (34–36°C)	Float	—	Computed

Humidity	Float (%)	15%	IoT sensor
Humidity deviation from optimal (50–65%)	Float	—	Computed
Weight trend (7-day)	Float (kg/day)	15%	Computed from IoT
Vibration level	Float	15%	IoT sensor
Active alert count	Integer	15%	Alert system
Days since last inspection	Integer	10%	Database
Pest prediction history (last 30d)	Integer (count)	10%	AI Model 1

Outputs

Output	Type	Description
Health score	Integer (0–100)	Overall hive health
Health category	Enum	`healthy` (80–100), `warning` (50–79), `critical` (0–49)
Top risk factors	String[]	Up to 3 factors dragging the score down
Trend	Enum	`improving`, `stable`, `declining` — based on 7-day score change

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Health Score Gauge	Radial gauge (green/yellow/red)	—	0–100	At-a-glance health on hive card
Health Score History	Line chart	Date (30/90 days)	Score (0–100)	Track health trends over time
Fleet Health Overview	Horizontal bar chart	Health score	Each hive (sorted)	Compare all hives at a glance
Health Factor Breakdown	Stacked bar or treemap	—	Contribution per factor	Show what's driving the score
Health Heatmap	Calendar heatmap	Day	Colour (green→red)	Daily health at a glance over months

---

6. Model 5: Maintenance Scheduling

SRS Reference: FR-3.04

Purpose

Predict when each hive needs maintenance (inspection, cleaning, component replacement, feeding) before issues arise.

Model Type

- **Architecture:** Survival analysis (Cox proportional hazards) or classification model (Random Forest)
- **Framework:** lifelines (Python survival analysis) or scikit-learn
- **Training data:** Historical maintenance logs with failure/event timestamps

Inputs

Input	Type	Source	Description
Days since last inspection	Integer	Database	Time gap since last manual check
Sensor anomaly count (7d)	Integer	Computed	Number of out-of-range readings
Current health score	Float (0–100)	Model 4	Composite health
Vibration trend	Float	IoT sensor	7-day vibration moving average
Season	Categorical	System clock	Maintenance needs vary by season
Hive age	Integer (days)	Database	Older equipment needs more attention
Last maintenance type	Categorical	Database	What was done last time
Active alerts	Integer	Alert system	Unresolved issues count

Outputs

Output	Type	Description
Days until next maintenance	Integer	Predicted days before maintenance is needed
Maintenance type	Enum	`inspection`, `cleaning`, `feeding`, `repair`, `treatment`
Urgency	Enum	`routine`, `soon`, `urgent`
Recommended actions	String[]	Specific tasks to perform

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Maintenance Calendar	Gantt / Timeline	Date	Hive (rows)	Schedule view of upcoming maintenance per hive
Maintenance Type Distribution	Donut chart	—	Count by type	What kinds of maintenance are most common
Days Between Maintenance	Box plot	Hive	Days	Distribution of maintenance intervals
Overdue Maintenance Alert List	Table / Card list	—	—	Hives where predicted date has passed

---

7. Model 6: Market Demand Forecasting

SRS Reference: FR-2.14, FR-3.10

Purpose

Forecast honey demand (units/month) and optimal pricing by region to help farmers time their sales and admins manage supply distribution.

Model Type

- **Architecture:** ARIMA / SARIMA or Facebook Prophet for time-series; Gradient Boosted Trees for price elasticity
- **Framework:** statsmodels / Prophet / scikit-learn
- **Training data:** Historical marketplace transaction volume, pricing, and regional data

Inputs

Input	Type	Source	Description
Monthly demand (units)	Integer[]	Marketplace DB	Last 12–24 months of order volumes
Monthly avg price ($)	Float[]	Marketplace DB	Historical average selling price per kg
Region	Categorical	Marketplace DB	Geographic area (Harare, Bulawayo, Masvingo, etc.)
Season / Month	Integer	System clock	Seasonal patterns (holiday demand spikes)
Supply volume (kg)	Float[]	Harvest records	Available honey supply per period
Number of active farmers	Integer	User DB	Supply-side capacity
Consumer signups (monthly)	Integer[]	User DB	Growing customer base indicator
Competing product count	Integer	External / Manual	Market competition factor

Outputs

Output	Type	Description
Demand forecast (next 3 months)	Float[]	Predicted units of honey demanded per month
Optimal price range	[Float, Float]	Suggested min–max price per kg to maximise revenue
Supply gap	Float	Predicted demand minus predicted supply
Top demand region	String	Region with highest projected demand
Seasonal advice	String	"Increase production for Oct–Dec holiday peak"

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Demand & Supply Trend	Dual-line chart	Month	Units demanded / supplied	Visualise supply–demand gap over time
Price Trend + Forecast	Line chart (solid + dashed)	Month	$/kg	Show historical price with future projection
Regional Demand Bar Chart	Horizontal bar chart	Demand (units)	Region	Compare demand across regions
Demand Heatmap by Region	Choropleth map	Geography	Colour intensity = demand	Geographic demand visualisation
Revenue Opportunity Matrix	Bubble chart	Demand (x)	Price (y), bubble size = gap	Identify best region/time to sell
Seasonal Demand Pattern	Polar / Radar chart	Month (12 spokes)	Normalised demand	Annual demand cycle at a glance

---

8. Model 7: Honey Quality Prediction

SRS Reference: FR-3.11

Purpose

Predict the quality grade (A, B, or C) of a honey harvest before lab testing, based on sensor conditions during the production period.

Model Type

- **Architecture:** Multi-class classifier — Random Forest or Gradient Boosted Trees
- **Framework:** scikit-learn / XGBoost
- **Training data:** Historical harvests with sensor data mapped to final quality grades

Inputs

Input	Type	Source	Description
Average internal temp (production period)	Float (°C)	IoT sensor	Mean temp during honey production
Temperature stability (std dev)	Float	Computed	How stable the temperature was
Average humidity	Float (%)	IoT sensor	Mean humidity during production
Humidity spikes count	Integer	Computed	Number of readings > 70%
Weight gain rate	Float (kg/day)	Computed	Honey accumulation speed
Harvest weight (sensor)	Float (kg)	IoT sensor	Total honey weight at harvest
Season	Categorical	System clock	Spring, Summer, Autumn
Flora type (region proxy)	Categorical	Database	Hive location as flora indicator
Days since last treatment	Integer	Database	Chemical residue risk
Colony health score at harvest	Float	Model 4	Health context

Outputs

Output	Type	Description
Predicted grade	Enum	`A`, `B`, or `C`
Grade probabilities	[Float, Float, Float]	Probability for each grade
Key quality drivers	String[]	Top 3 factors influencing the grade
Improvement suggestions	String[]	Actions to improve grade next season

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Quality Grade Distribution	Donut / Pie chart	—	Count per grade	Overview of grade distribution across harvests
Quality Prediction Confidence	Stacked bar	Harvest	Grade probability (A/B/C)	Show how confident the model is per harvest
Sensor Conditions vs Grade	Scatter plot	Avg temp	Avg humidity, colour = grade	Visualise how conditions correlate with quality
Grade Trend Over Time	Stacked area chart	Season	Count per grade	Are grades improving or declining?
Factor Importance	Horizontal bar chart	Feature importance score	Input feature	Which inputs matter most for quality

---

9. Model 8: Weight Reconciliation Anomaly Detection

SRS Reference: FR-3.02

Purpose

Flag suspicious discrepancies between IoT-recorded harvest weight (sensor) and physically delivered weight. Detects potential fraud, sensor errors, or handling losses.

Model Type

- **Architecture:** Isolation Forest or Autoencoder for anomaly detection
- **Framework:** scikit-learn / PyTorch
- **Approach:** Unsupervised — learns normal weight loss patterns during transport and flags outliers

Inputs

Input	Type	Source	Description
Weight (sensor recorded)	Float (kg)	IoT sensor	Weight at harvest time from Smart Hive
Weight (delivered)	Float (kg)	Quality control	Physical scale weight at E Mukoko
Weight difference (%)	Float	Computed	`abs(sensor - delivered) / sensor × 100`

Distance from hive to depot	Float (km)	Database	Transport distance (more distance = more expected loss)
Transport duration	Float (hours)	Database/Farmer input	Time between harvest and delivery
Ambient temperature	Float (°C)	Weather API	Temperature during transport (affects honey viscosity)
Farmer history	Float	Database	Average historical discrepancy % for this farmer
Harvest season	Categorical	System clock	Some seasons have stickier honey

Outputs

Output	Type	Description
Anomaly flag	Boolean	`true` if discrepancy is suspicious
Anomaly score	Float (0–1)	How anomalous this harvest is (1 = very)
Expected loss range	[Float, Float]	Normal loss % range for this context
Likely cause	Enum	`normal_loss`, `sensor_error`, `handling_issue`, `investigation_needed`
Action	String	Suggested next step (approve / re-weigh / review)

Graphs Needed

Graph	Type	X-Axis	Y-Axis	Purpose
Sensor vs Delivered Scatter	Scatter plot	Sensor weight (kg)	Delivered weight (kg)	Identify outliers from the diagonal
Discrepancy Distribution	Histogram	% discrepancy	Frequency	Show the normal distribution and outlier range
Anomaly Score Timeline	Line chart with threshold	Date	Anomaly score (0–1)	Track anomalies over time
Farmer Reconciliation History	Bar chart	Harvest	Discrepancy % (coloured by flag)	Per-farmer integrity tracking

---

10. Graphs & Visualisation Summary

Complete list of all graphs needed across the platform, organised by dashboard section.

10.1 Farmer Dashboard — Hive Detail

#	Graph	Chart Library	Model Source
1	Internal & External Temperature (dual line)	Recharts `LineChart`	IoT raw data
2	Humidity over time (area)	Recharts `AreaChart`	IoT raw data
3	Colony Weight over time (area)	Recharts `AreaChart`	IoT raw data
4	Vibration over time (line)	Recharts `LineChart`	IoT raw data
5	Weight Trend + Yield Forecast (line + dashed)	Recharts `ComposedChart`	Model 3
6	Swarm Probability Gauge	Custom SVG radial	Model 2
7	Health Score Gauge	Custom SVG radial	Model 4
8	Health Score History (line)	Recharts `LineChart`	Model 4
9	Contributing Factors Radar	Recharts `RadarChart`	Model 2 / 4

10.2 Farmer Dashboard — Overview

#	Graph	Chart Library	Model Source
10	Fleet Health Overview (horizontal bar)	Recharts `BarChart`	Model 4
11	Health Heatmap (calendar)	Custom grid component	Model 4
12	Fleet Yield Projection (stacked area)	Recharts `AreaChart`	Model 3
13	Revenue Forecast (line)	Recharts `LineChart`	Model 3 + 6

10.3 AI Diagnostics Page

#	Graph	Chart Library	Model Source
14	Pest Risk Timeline (area chart)	Recharts `AreaChart`	Model 1
15	Pest Type Distribution (donut)	Recharts `PieChart`	Model 1
16	Monthly Alert Frequency (stacked bar)	Recharts `BarChart`	Model 1
17	AI Accuracy Over Time (line)	Recharts `LineChart`	Model 1
18	Swarm Risk Over Time (area)	Recharts `AreaChart`	Model 2

10.4 Market Demand Dashboard

#	Graph	Chart Library	Model Source
19	Demand & Supply Trend (dual line)	Recharts `LineChart`	Model 6
20	Price Trend + Forecast (line + dashed)	Recharts `LineChart`	Model 6
21	Regional Demand Bar Chart (horizontal)	Recharts `BarChart`	Model 6
22	Revenue Opportunity Bubble	Recharts `ScatterChart`	Model 6
23	Seasonal Demand Pattern (polar)	Recharts `RadarChart`	Model 6

10.5 Admin Analytics Dashboard

#	Graph	Chart Library	Model Source
24	Quality Grade Distribution (donut)	Recharts `PieChart`	Model 7
25	Grade Trend Over Time (stacked area)	Recharts `AreaChart`	Model 7
26	Sensor vs Delivered Scatter	Recharts `ScatterChart`	Model 8
27	Discrepancy Distribution (histogram)	Recharts `BarChart`	Model 8
28	Farmer Reconciliation History (bar)	Recharts `BarChart`	Model 8
29	Maintenance Calendar (Gantt)	Custom component	Model 5
30	Demand Heatmap by Region (map)	Leaflet / Mapbox	Model 6

10.6 Farmer Profile Page

#	Graph	Chart Library	Model Source
31	Seasonal Yield Comparison (grouped bar)	Recharts `BarChart`	Model 3
32	Quality Prediction Confidence (stacked bar)	Recharts `BarChart`	Model 7
33	Earnings Over Time (line)	Recharts `LineChart`	Transaction data

Total unique graphs: 33

---

11. Data Pipeline Architecture

```
Smart Hive Sensors
        │
        ▼
  MQTT / HTTP Ingest
        │
        ▼
  ┌──────────────────────────────┐
  │   Time-Series Database       │
  │   (TimescaleDB / InfluxDB)   │
  │   - Raw sensor readings      │
  │   - 30-min resolution        │
  └──────────────────────────────┘
           │
           ▼
     Feature
     Engineering
           │
           ▼
  ┌──────────────────┐
  │  Models 1-8      │
  │  (All sensor-    │
  │   based)         │
  └──────────────────┘
          │
          ▼
  ┌──────────────────────────┐
  │    Prediction API        │
  │    /api/ai/*             │
  └──────────────────────────┘
           │
     ┌─────┴──────┐
     ▼            ▼
  Dashboard    Alert
  Graphs       Engine
                 │
                 ▼
          Push / SMS / Email
```

Feature Engineering Requirements

Feature	Computation	Frequency
Rolling averages (7d, 14d, 30d)	Mean of sensor readings over window	Every new reading
Rate of change	`(current - previous) / time_delta`	Every new reading
Standard deviation (stability)	Std dev of readings over window	Daily
Anomaly counts	Count of readings outside optimal range	Daily
Seasonal encoding	Sine/cosine transformation of month	Static per reading
Days since event	`current_date - last_event_date`	Daily

---

12. Technology Stack

Component	Technology	Notes
ML Framework	ONNX Runtime, scikit-learn, XGBoost, LightGBM	All models are sensor/tabular-based
ML Training	scikit-learn / XGBoost / LightGBM (tabular), PyTorch (LSTM)	Model development
Time-Series Forecasting	Facebook Prophet, statsmodels	Yield and demand forecasting
Anomaly Detection	scikit-learn (Isolation Forest)	Weight reconciliation
Survival Analysis	lifelines (Python)	Maintenance scheduling
Model Serving	FastAPI + ONNX Runtime	Low-latency REST API
Feature Store	Redis (real-time) + PostgreSQL (batch)	Precomputed features
Charting	Recharts (React)	All frontend graphs
Map Visualisation	Leaflet / Mapbox GL	Regional demand heatmap
Model Monitoring	MLflow or Weights & Biases	Track experiments, detect drift
Training Infrastructure	Google Colab / AWS SageMaker	GPU for LSTM training if needed

---

End of Document — E Mukoko Innovations | Predictive Models Specification