Milestone 5 — Task 1: Plan Deployment

CRISP-DM Phase 6 — Plan Deployment

Group A

YYYY-MM-DD

Table of Contents

# Executive Summary

This document delivers the Plan Deployment for CRISP-DM Phase 6 (Milestone 5, Task 1). We will deploy the Random Forest regression model trained in Milestone 3 to a minimal, user-friendly interface using R Shiny. The deployment focuses on enabling policy analysts to obtain a prediction of the log-scaled health index and view concise context on model drivers, while maintaining strict consistency with the trained feature schema and documenting a clear pathway from pilot to production.

The model to be deployed is saved at 02\_Project/Milestone\_3/Task\_03/outputs/final\_random\_forest\_model.rds. Per Milestone 4, the model met business success thresholds but flagged caveats (potential data leakage, tuning on test set, baseline comparisons). We address these by: (1) reaffirming preprocessing parity with the trained schema; (2) reserving the test set for final validation; (3) defining acceptance criteria and a go/no-go gate prior to production deployment.

Outcome for Milestone 5: a minimal Shiny demo running locally, with a complete deployment strategy and acceptance criteria aligned to the brief. Monitoring and maintenance details are covered in Task 2 and referenced here where needed.

# Objectives and Users

Primary users and goals (from Milestone 1 and Milestone 4): - Users: Government agencies, NGOs, and Department of Health analysts. - Objective: Provide a quick, interpretable prediction of a log-scaled health index (target: value\_log\_scaled) to support policy exploration and resource prioritization. - Constraints: No personally identifiable information (PII). Operate on engineered features already used for model training. - Success: Low-latency predictions; consistent outputs; simple, explainable inputs aligned to top drivers discussed in Milestone 4.

# Deployment Artefact (Minimal Interface)

Form factor: R Shiny app packaged under 02\_Project/Milestone\_5/shiny\_app/.

Core behavior: - Loads the trained model artifact (final\_random\_forest\_model.rds). - Builds a full feature vector aligned to the training schema, with most fields filled by training-set defaults (means/modes) to simplify the UI. - Exposes a few interpretable controls (e.g., a small set of engineered proxy features) and returns a single predicted value\_log\_scaled value. - Displays a short help panel describing the model scope and assumptions.

User interface specification (no custom styling required): - Inputs: 2–4 simple controls (e.g., numeric slider for a scaled precision proxy; two checkboxes; one categorical dropdown such as sample\_size\_tier). - Actions: Predict button. - Outputs: Predicted log-scaled index (numeric) and a compact explanatory note referencing Milestone 4 drivers (water, sanitation, literacy, healthcare access) to ground interpretation.

Versioning and artifacts: - App folder: 02\_Project/Milestone\_5/shiny\_app/ with a README.md describing how to run locally. - The app references training schema from 02\_Project/Data/04\_Split/train\_data.csv and feature roles from 02\_Project/Data/03\_Scaled/feature\_metadata.csv to ensure parity.

# Environment and Access

Pilot environment (Milestone 5): - Run locally in RStudio. Dependencies: R (>= 4.2), shiny, readr, dplyr, and the modeling package used (e.g., randomForest or ranger). - No network dependencies; uses local artifacts under 02\_Project/.

Path to production (Milestone 6+): - Option A: RStudio Connect (on-prem) with internal authentication. - Option B: shinyapps.io (cloud) for a limited, private pilot.

Access and security: - No PII processed. Restrict access to internal stakeholders for pilot. Add a lightweight usage log (timestamp, user/session id if available, prediction success/failure) to support monitoring.

# Data Flow and Preprocessing

Authoritative inputs and schema: - Model: 02\_Project/Milestone\_3/Task\_03/outputs/final\_random\_forest\_model.rds. - Training schema: 02\_Project/Data/04\_Split/train\_data.csv (defines expected engineered features for inference). - Feature roles: 02\_Project/Data/03\_Scaled/feature\_metadata.csv (indicates feature types to include: numeric\_scaled, binary, label\_encoded, categorical\_engineered, one\_hot\_encoded; and targets to exclude).

Inference construction: - Use the training schema to build a single-row feature frame. Default all fields using training-set means/modes; override a few UI-controlled fields. - Do not introduce raw-source variables (e.g., raw water/sanitation rates) directly into the app if they are not part of the engineered training schema; that prevents schema mismatch. - Maintain preprocessing parity: the model expects already scaled/encoded engineered features. The app must not re-scale or re-encode differently.

Validation and auditability: - Verify that train/test/validation splits remain disjoint and that no test-set information is used during input default computation beyond simple descriptive stats. - Add a simple schema check at app startup (ensure required columns exist in train\_data.csv and match the model’s expectation).

# Tool Research and Selection

Options considered and rationale: - Shiny (Selected): Native to R; directly loads .rds; fast to implement; suitable for small user base; rich community support. - Plumber API + Next.js UI: Flexible and web-native UI, but adds integration complexity and dev overhead; not needed for M5 timelines. - Power BI: Excellent for dashboards; limited for model inference unless using custom R scripts; heavier governance; not ideal for rapid model serving.

Selection justification: - Shiny meets the brief’s “user-friendly interface” requirement with minimal integration effort, leveraging the existing R model artifact and engineered features. It enables a working pilot within Milestone 5 with a clear upgrade path to RStudio Connect or shinyapps.io.

# Acceptance Criteria / KPIs (Deployment Readiness)

Functional criteria: - App runs locally without errors and loads the model artifact. - UI inputs validate ranges/types; prediction returns within < 1 second locally. - Output displays predicted value\_log\_scaled and help text indicating scope and assumptions.

Non-functional criteria: - Parity: Inference feature vector matches the training schema; no missing required columns. - Stability: Recoverable errors are surfaced as friendly messages; app does not crash on invalid input.

Business KPIs (pilot): - Time to first insight: < 5 minutes for a new analyst to obtain a prediction following the quick-start. - Usage: ≥ 5 unique pilot sessions in the first week (if hosted). - Trust indicators: Stakeholder feedback indicates clarity of inputs/outputs (qualitative score ≥ 4/5).

# Alternatives & Phasing (Pilot vs Production)

Phased approach: - Phase 1 (M5): Local Shiny pilot; validate usability and parity; document any gaps. - Phase 2 (M6): Host on shinyapps.io or RStudio Connect; add lightweight authentication; publish user guide and finalize docs. - Phase 3 (Post-M6, optional): Harden for production (CI/CD, environment pinning, periodic validation jobs).

Go/No-Go gate (pre-production): - Meets Acceptance Criteria; passes schema checks; stakeholder sign-off; no critical issues from Milestone 4 audit.

# Knowledge Propagation & Change Management

Artifacts and communication: - Quick-start in 02\_Project/Milestone\_5/shiny\_app/README.md (how to run locally, inputs/outputs explained). - One-page user guide (embedded in the Milestone 5 plan PDF) with screenshots after the demo is running. - 30-minute demo recording or live walkthrough for stakeholders; capture Q&A. - Change log: track updates to the app, model artifact, and schema references.

# Organisational Integration & Security (Lightweight)

Integration: - Link to the app from the internal project page when hosted. - Reference the CRISP-DM artifacts and data dictionary locations for context.

Security and privacy: - Internal-only during pilot; no PII processed; minimal analytics; follow org password policies if auth is added.

# Benefits Measurement (Business KPIs)

Post-deployment measurement plan: - Analyst time saved: qualitative feedback and small time studies (baseline vs using the app). - Decision support quality: stakeholder self-reported clarity and utility of predictions. - Adoption: number of sessions and repeat users (if hosted) over first month.

# Implementation Plan (Milestone 5 Scope)

1. Confirm model artifact and schema (paths, availability).
2. Specify UI controls and default mapping to training schema.
3. Implement minimal Shiny app (load model, build default feature vector, apply overrides, predict, display).
4. Add input validation and error handling; add short help text.
5. Perform pilot test locally; collect quick feedback; capture screenshots for the plan PDF.
6. Document acceptance results; log any gaps for Milestone 6.

# References

* Milestone 5 brief: 02\_Project/Milestone\_5/Milestone\_5.md
* Milestone 4 evaluation: 02\_Project/Milestone\_4/Task\_1.md, 02\_Project/Milestone\_4/Task\_2.md, 02\_Project/Milestone\_4/Task\_3.md, 02\_Project/Milestone\_4/GroupA\_Milestone\_04.Rmd
* Milestone 3 modeling: 02\_Project/Milestone\_3/Task\_03/Task\_03.Rmd, 02\_Project/Milestone\_3/Task\_03/outputs/final\_random\_forest\_model.rds
* Data folders: 02\_Project/Data/03\_Scaled/feature\_metadata.csv, 02\_Project/Data/04\_Split/train\_data.csv, .../val\_data.csv, .../test\_data.csv