HDPSA Milestone 6

Bin 381 Group A

10/22/2025

Members:

Llewellyn Jakobus Fourie – 601314

Juan Oosthuizen – 600161

Erin David Cullen – 600531

Qhamaninande Ndlume – 601436

Table of Contents

Executive Summary

This final report represents the culmination of the HDPSA project, which applied the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology to explore, model, and deploy a data-driven system for analysing health and demographic trends in South Africa. The project's primary goal was to enhance the efficiency, accuracy, and interpretability of national health data through predictive analytics and interactive reporting.

The project progressed through six milestones, from problem understanding and data collection to model deployment and maintenance planning. This final report documents the outcomes of the Deployment Phase (Milestone 6), where the insights, datasets, and models from earlier milestones were integrated into a unified, reproducible analytical ecosystem.

The resulting deployment includes a predictive Random Forest regression model, hosted as a Flask API, and a Next.js web dashboard embedding an interactive Power BI report. Together, these components enable automated data validation, real-time reporting, and improved stakeholder engagement. Furthermore, the model's monitoring and maintenance strategy ensures long-term reliability, fairness, and ethical compliance.

Core outcomes:

- **Business value:** Faster validation of national health surveys, improved forecasting for planning, and visual communication tools that enable policy makers to make evidence-based decisions.
- **Model performance:** Random Forest regression achieved $R^2 = 0.997$, RMSE = 0.0554, and MAE = 0.0382, demonstrating excellent predictive stability.
- **Deployment environment:** Integration of predictive and reporting services via Flask and Next.js ensures accessibility, scalability, and reproducibility.
- **Operational sustainability:** Maintenance plans define retraining thresholds, drift detection mechanisms, and governance cycles to ensure continuous model relevance.

Business Understanding (Recap)

The HDPSA project was designed in response to challenges faced by health ministries and research institutions when managing diverse and incomplete health datasets. Stakeholders included national data quality teams, DHS survey planners, researchers, and policy analysts. Each of these groups required the ability to validate data, detect anomalies, and extract actionable insights from large, aggregated health indicators.

Primary Objectives: The key objectives were to validate health-survey data automatically, identify and fill temporal gaps in historical records, and communicate these patterns effectively through an interactive dashboard. These objectives were directly aligned with the business needs of decision-makers in the public health domain.

Success Criteria: The success of the system was evaluated through quantitative metrics (R^2, RMSE, and MAE) and qualitative feedback from stakeholders. Reduced manual data validation time and improved interpretability of trends were used as indicators of successful business value creation.

Scope and Limitations: The project focused exclusively on aggregated survey indicators rather than individual-level data, ensuring privacy compliance and maintaining an appropriate level of analytical abstraction. Predictions were limited to identifying trends within existing distributions, avoiding any causal or personal inferences.

Key References:

- *Task_1_Deployment_Strategy.md*: Defined stakeholder roles, success measures, and tool selection.
- *Milestone_1.md / Milestone_2.md*: Established the foundational business understanding and data scope.

Data Understanding and Sources

The HDPSA analysis was based on DHS-style national datasets containing health indicators across multiple domains, including water access, sanitation, immunization, child and maternal health, and literacy. Each dataset followed a consistent schema, ensuring compatibility and enabling efficient transformation for predictive modelling.

Data Sources and Documentation:

- *Data_Dictionary.md*: Describes each field's meaning, units, and valid ranges.
- *Data_Pipeline.md*: Outlines the multi-stage processing framework and dependencies between stages.
- **Folder Structure:** Data moved through a defined pipeline (01_Raw → 02_Cleaned → 03_Scaled → 04_Split → 05_ModelReady → Flat Data), supporting lineage and reproducibility.

Key Features of the Data:

- Consistent column naming conventions and controlled vocabularies facilitated automation and repeatable transformations.
- Each record contained fields such as Indicator, Value, Precision, SurveyYear, and confidence intervals (CILow, CIHigh), providing both central estimates and uncertainty measures.
- Denominator fields supported quality assessment by indicating sample size and weighting.

By combining multiple survey years across domains, the dataset provided a comprehensive view of South Africa's public health evolution over time.

Exploratory Data Analysis (Summary)

During the exploratory phase, the project team examined the completeness, variability, and relationships across all curated datasets. Of the initial thirteen raw data domains, seven were selected for modelling due to their completeness, consistency, and relevance to health outcomes. The selection included datasets such as *water*, *immunization*, *child mortality*, *HIV behaviour*, *access to healthcare*, and *toilet facilities*.

Each cleaned dataset maintained uniform structure (11 columns) but differed in the number of indicators and observations, reflecting domain-specific complexities. For instance, *water* contained 100 rows and 62 indicators, while *child-mortality-rates* had 40 rows and 15 indicators. Missing data were non-existent in curated files, eliminating the need for imputation.

The analysis also revealed that indicators combined both count and percentage formats. To avoid scale distortions, indicators were normalised, and feature engineering was applied to distinguish between measurement units. Confidence intervals and precision scores were later incorporated into modelling to account for uncertainty in the source data.

Data Preparation

Data preparation followed a well-documented multi-stage process. Each stage was executed sequentially to ensure traceability, reliability, and consistency across the modelling pipeline.

1. **01_Raw:** Immutable storage of original datasets ensured data provenance and reproducibility.
2. **02_Cleaned:** Missing and inconsistent records were resolved; field types were standardised; duplicates were removed.
3. **03_Scaled:** Feature scaling and categorical encoding were applied to ensure consistent numerical ranges and prevent model bias.
4. **04_Split:** Datasets were divided into training, validation, and testing subsets, ensuring that model evaluation was conducted on unseen data.

The structured pipeline enabled modular development and simplified retraining or reprocessing. This process was crucial in transforming heterogeneous health survey data into consistent, high-quality inputs for machine learning.

Why Raw Data Cannot Be Used Directly

The *01_Raw* dataset served as the archival foundation but was unsuitable for direct analysis. Its contents were aggregated summaries rather than individual observations, leading to several analytical limitations:

- **Aggregation Effects:** Records summarised large population groups, preventing micro-level inference.
- **Unit Inconsistency:** Indicators expressed both rates and counts, distorting model inputs.
- **Semantic Diversity:** Each domain used unique categorical encodings, causing sparse data matrices when merged.
- **Irregular Time Series:** Uneven survey intervals disrupted trend analyses.
- **Uncertainty:** Variability in confidence intervals was not uniformly represented.

Early experiments confirmed that unsupervised methods like clustering and PCA grouped indicators by unit scale rather than semantic similarity, while simple regression models overfit due to sparsity. These findings validated the need for a structured cleaning and transformation pipeline before modelling.

Modelling

A **Random Forest Regressor**, implemented in Python's *scikit-learn*, was selected for its robustness, non-linearity handling, and interpretability through feature importance. The model predicted the target variable value_log_scaled, representing scaled indicator values after log transformation.

Implementation: The modelling pipeline was encapsulated in *ml_service.py*, which handled training, validation, and inference. The *app.py* script provided REST API endpoints for retraining, evaluation, and prediction. Trained artifacts were stored in a versioned directory, enabling rollback and auditing.

Performance Metrics:
The model achieved $R^2 = 0.997$, RMSE = 0.0554, and MAE = 0.0382, confirming excellent generalisation within the dataset's historical bounds. However, the model was designed to operate strictly within historical data distributions and should not be used for extrapolation or individual predictions.

Evaluation and Validation

The evaluation phase ensured that the model's predictive performance was reliable and fair across different population segments. Using the prepared data splits, the model's accuracy was verified against baseline thresholds.

Metrics: RMSE, MAE, and R^2 were computed to assess error magnitude and explained variance.
Fairness Checks: Subgroup parity tests compared error rates across provinces and urban/rural categories.
Confidence Awareness: Confidence intervals from the original data were used to interpret deviations in predictions, distinguishing natural uncertainty from model error.

Results indicated stability across validation cycles, with no significant fairness deviations. This consistency verified that the model could be safely deployed for operational use.

Deployment

Deployment activities combined both predictive and reporting components. The **Next.js dashboard** serves as the user interface, embedding a **Power BI report** for interactive exploration of results. The dashboard provides visual summaries for key domains such as sanitation, immunization, and water access, enabling stakeholders to interpret results intuitively.

The **Flask API** operates as the analytical backend, providing endpoints for health checks, model training, evaluation, and batch predictions. Its modular design ensures extensibility and integration with monitoring scripts. Data flows between the backend and frontend are governed by a shared data schema to maintain consistency.

System Architecture:

- Frontend: Next.js dashboard integrating Power BI.
- Backend: Flask ML API handling model operations.
- Storage: Local and cloud repositories storing model artifacts and logs.

This layered architecture ensures that updates to the analytical engine or UI can occur independently, supporting agile iteration and maintainability.

Monitoring and Maintenance

To sustain accuracy and fairness, the project adopted a structured **Monitoring and Maintenance Plan**.

Monitoring: Defined performance thresholds (e.g., RMSE > 0.07 or R^2 < 0.95) trigger automated alerts. Data drift is measured using PSI and KS tests. A Power BI-based operations dashboard visualises performance trends and model stability.

Maintenance: Retraining triggers, rollback procedures, and governance roles were established. Each quarter, an ethics and performance review is conducted to verify compliance with fairness standards. Version control systems ensure reproducibility and rollback capabilities.

By combining these practices, the HDPSA system achieves operational transparency and accountability across its lifecycle.

Ethical Considerations

Ethical governance was embedded throughout the project lifecycle. All processing occurred on anonymised, aggregated data, ensuring compliance with POPIA. Role-based access control was implemented to protect model artifacts and logs.

Fairness and Inclusivity: Performance parity across geographic and demographic subgroups was maintained within ±25% of overall MAE and RMSE. Continuous fairness monitoring ensures that retraining occurs if disparities persist beyond two evaluation cycles.

Transparency: Model documentation, feature importance rankings, and explainability reports accompany each release. Dashboards display performance and drift summaries for public accountability.

Escalation Procedures: The Data Scientist prepares fairness reports, the Engineer validates reproducibility, and the BI Manager authorises redeployment. Breaches of performance or fairness thresholds automatically initiate review and retraining workflows.

Visual Storytelling and Interactivity

Interactive storytelling was a core design principle of the HDPSA reporting interface. The Power BI report embedded within the Next.js dashboard allows users to navigate through health indicators via slicers, filters, and tooltips. Charts illustrate temporal trends and disparities across provinces, translating complex data into intuitive visuals.

The layout ensures accessibility and interpretability, even for non-technical audiences. Descriptive text and disclaimers clarify data provenance and limitations, reducing the risk of misinterpretation. These design decisions collectively enhance user engagement and facilitate informed decision-making.

Project Review and Lessons Learned

Achievements:

The project successfully established a full CRISP-DM pipeline from raw data to operational deployment. The modular data pipeline streamlined reproducibility, while the Random Forest model provided highly accurate predictions. The dashboard's interactive nature improved data communication among stakeholders.

Areas for Improvement:

Automated continuous integration (CI) should be implemented for the Flask API to improve reliability. Fairness assessments can be expanded with calibration curves and temporal drift monitoring. Integrating model metrics directly into the Power BI dashboard would provide a unified operational view.

Future Directions:

Adding conformal prediction intervals would quantify uncertainty more effectively. Gradient boosting or explainable sparse models could improve interpretability. Finally, a model registry system could streamline version tracking and artifact management.

Personal Reflection:

This project provided valuable experience in managing large datasets, implementing end-to-end machine learning systems, and collaborating in a multidisciplinary team. It highlighted the importance of ethical foresight, reproducibility, and structured governance in data science projects.

Deliverables Checklist

- **Final Report:** *Milestone_6_Final_Report.md* – structured according to CRISP-DM standards.
- **Presentation:** 10-slide summary highlighting data, model, and deployment outcomes.
- **Code and Data:** R Markdown scripts, Python API modules, and final processed datasets.
- **Dashboard:** *page.tsx* with Power BI embed and disclaimers.
- **Predictive Service:** *app.py* and *ml_service.py* for model training and inference.
- **Documentation:** *Data_Dictionary.md* and *Data_Pipeline.md* detailing structure and flow.

References

- CRISP-DM Methodology (Phases 1–6).

- *GroupA_Milestone_04.Rmd* and *Milestone_5_Report.Rmd*.
- Operational Plans: *Task_1_Deployment_Strategy.md*, *Task_3_Maintenance_Plan.md*, *Task_4_Monitoring_Plan.md*.
- Data Documentation: *Data_Dictionary.md*, *Data_Pipeline.md*.