**Project Report: TechBharat — RTGS Agent**

**Introduction**

**TechBharat** is an end-to-end data pipeline system designed to transform raw, messy government datasets into actionable insights for policymakers. It provides tools for data cleaning, transformation, analysis, and visualization, ensuring decision-makers can access information in intuitive formats like CLI dashboards, summaries, and APIs.

[Policymaker CLI] --> [RTGS Agent]

1. Load Health Dataset

2. Clean & Standardize

- Fix missing values

- Standardize district/facility names

3. Transform

- Aggregate counts (beds, staff per district)

- Generate ratios / coverage metrics

4. Analyze

- Highlight districts below thresholds

- Identify patterns / imbalances

5. Output Insights

- ASCII tables

- Summary logs

- Optional CSV / PNG

**Problem Statement**

Government and sector datasets (e.g., health, tourism, consumption, temperature) are often raw, inconsistent, and not user-friendly. Policymakers cannot easily extract answers for governance decisions, resource allocation, or long-term planning.

**Solution Overview**

TechBharat automates the full pipeline:

1. **Ingestion & Cleaning** – Fix missing values, standardize district/facility names, and remove duplicates.
2. **Transformation** – Aggregate metrics (e.g., staff per district, total visitors), compute ratios, and derive insights.
3. **Analysis** – Compare data against thresholds, detect imbalances, and identify problem areas.
4. **Visualization & Reporting** – Provide outputs as ASCII tables, summaries, dashboards, or CSV/PNG files.

**Workflow Example (Health Dataset)**

* **Dataset**: Health Facilities of Telangana (Telangana Open Data Portal).
* **Steps**:
  + Loaded and cleaned 2,717 duplicate rows.
  + Aggregated counts (e.g., beds, staff per district).
  + Calculated ratios like kit coverage and high-risk pregnancy ratio.
  + Highlighted districts below thresholds or requiring policy intervention.
  + Generated outputs (ASCII tables, PNG dashboards, and executive summaries).

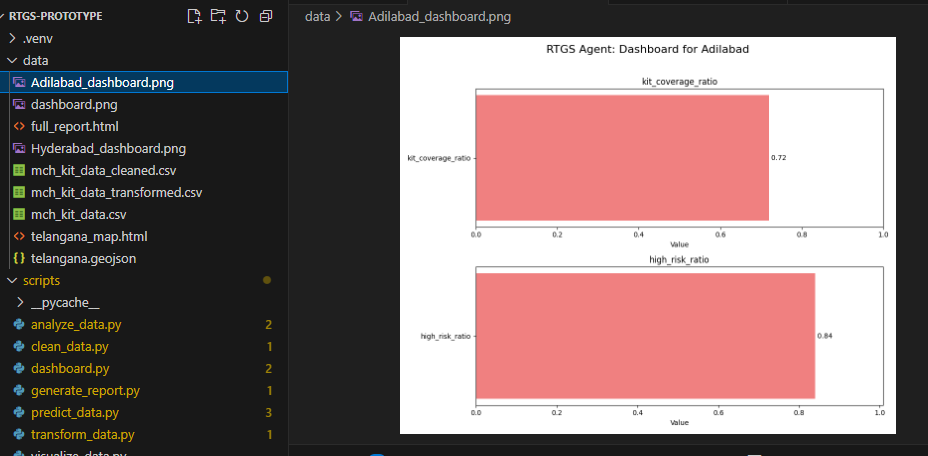
**Insights Generated**:

* *Low Kit Distribution*: Medchal-Malkajgiri, Wanaparthy, Hyderabad flagged below 80%.
* *High-Risk Pregnancies*: Mahabubabad, Mulugu, Nagarkurnool exceeded 10%.

**CLI Functionality**

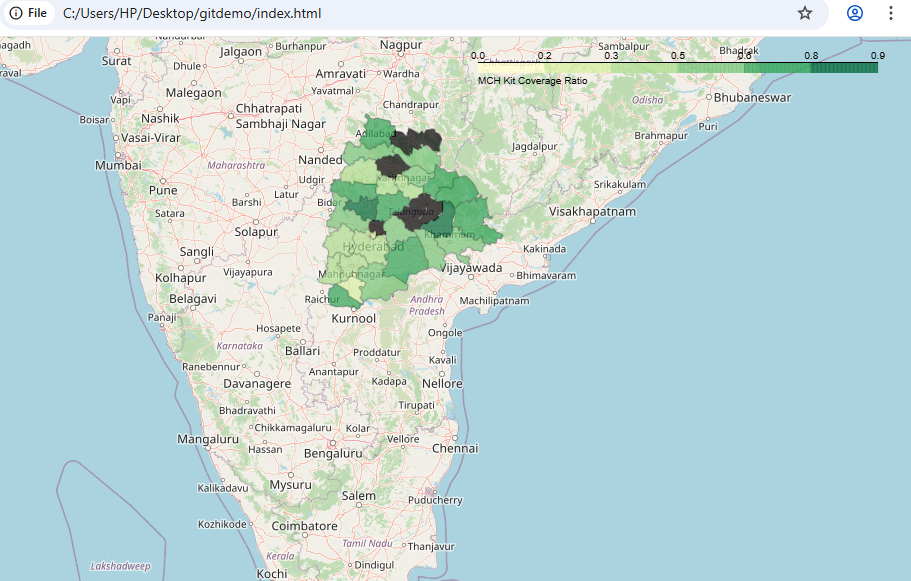
Policymakers interact via an **interactive CLI (RTGS-CLI)**:

* get\_insights Hyderabad → returns insights for Hyderabad.
* set\_threshold kits 0.7 → dynamically change thresholds.
* run\_analysis → re-analyze with new settings.
* dashboard\_for Adilabad kit\_coverage\_ratio,high\_risk\_ratio → generates district-specific dashboards as PNGs.



* predict 2024-03-01 → forecasts demand for kits/resources.
* **geospatial visualization**

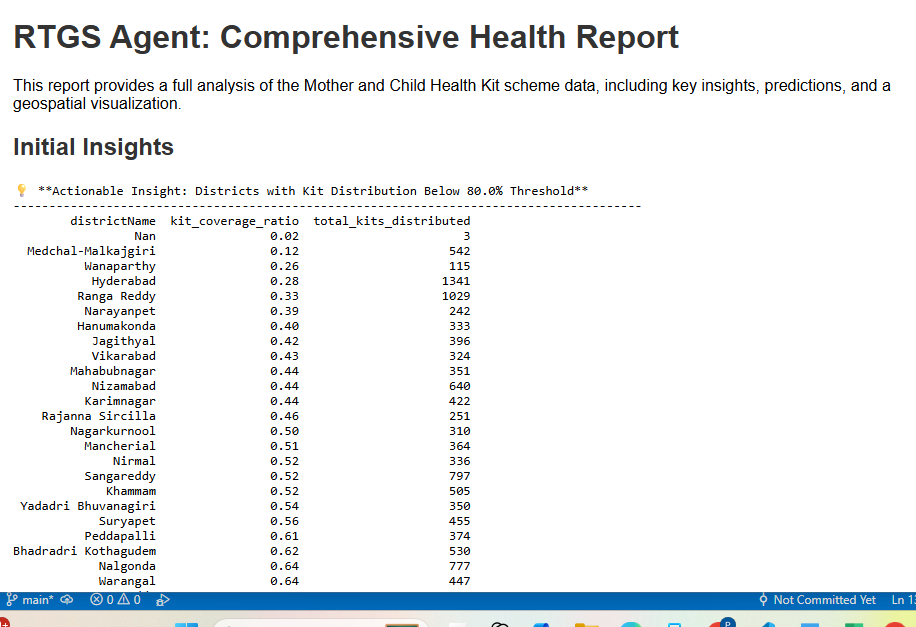
This step refines the geospatial visualization by changing the map's color scheme to improve clarity and intuition for policymakers. By updating the fill\_color property in the folium code, we can use palettes like YlOrBr or BrBG to highlight a progressive range of values or show data's deviation from an average. This makes the map a more effective tool for visual data analysis.



**Darker shades of green** indicate a **higher** kit coverage ratio. These are the districts performing well, with more kits distributed relative to registered women.

**Lighter shades of green** indicate a **lower** kit coverage ratio. These are the districts with insufficient kit distribution that require a policymaker's immediate attention.

Report Building- **Automated Report Generation**



**Advanced Features**

1. **Predictive Analysis**
   * Forecast demand for resources using time-series modeling.
   * Example: Predicted MCH kits for March 2024 → 15,294.
2. **Geospatial Visualization**
   * Maps districts using color-coded ratios (e.g., kit coverage).
   * Darker shades = higher coverage; lighter shades = gaps.
3. **Automated Report Generation**
   * Produces comprehensive HTML/PDF reports.
   * Includes insights, dashboards, predictions, and summaries.
4. **Root Cause Analysis**
   * Uses correlation analysis to detect systemic issues.
   * Example: Kit coverage negatively correlated with number of registered women (-0.47).
5. **Configurable & Data-Agnostic**
   * Configured via config.yaml → supports multiple datasets (health, tourism, temperature, consumption).
   * No code changes needed to switch datasets.
6. **API & NLP Agent**
   * Exposes FastAPI endpoints (/pipeline/run, /nlp\_query, /insights).
   * Supports natural language queries like: *“How many tourists visited Warangal in 2023?”*.

**Datasets Supported**

* **Health Data**: Kit distribution, ANC completion, high-risk pregnancies.
* **Tourism Data**: Visitors per district, seasonal trends.
* **Temperature Data**: Average and maximum temperatures by district.
* **Consumption Data**: Billed services, usage patterns.

**Technical Stack**

* **Language**: Python
* **Libraries**: pandas, spaCy, matplotlib, seaborn, folium
* **Frameworks**: FastAPI for APIs
* **Data**: Telangana Open Data Portal (sample datasets)
* **Config**: config.yaml for dataset flexibility

**Outcomes**

* Clean, transformed, and standardized datasets.
* Executive summaries for quick review.
* Dynamic CLI with dashboards and predictions.
* Secure, extensible API for NLP queries and dashboards.
* System proven to be scalable and adaptable for multiple datasets.

**Closing Summary**

TechBharat is more than a script — it’s a robust **prototype RTGS Agent** that:

* Cleans and standardizes raw datasets.
* Transforms and analyzes data against policy thresholds.
* Outputs actionable insights through CLI dashboards, APIs, and summaries.
* Supports predictive analysis, geospatial visualizations, and automated reporting.