

UIDAI Data Hackathon 2026:

Comprehensive Data Analysis Report

Theme: Unlocking Societal Trends in Aadhaar Enrolment and Updates

Date: January 19, 2026

1. Problem Statement and Approach

Problem Statement

The Aadhaar ecosystem, while robust, faces several systemic challenges as it transitions from a “Saturation Phase” to a “Maintenance Phase.” Our analysis identifies four critical problem areas:

- 1. National Security Risks:** Disproportionate enrolment velocity in sensitive border districts.
- 2. Operational Inefficiencies:** Static resource allocation failing to handle predictable “Tuesday/Saturday” load spikes.
- 3. Regional Demographic Gaps:** A “Hidden Cohort” of adults in North-East India enrolling for the first time, which the current child-centric model is ill-equipped to handle.
- 4. The Digital Divide:** A behavioral gap in rural areas where citizens prioritize demographic updates for welfare benefits while neglecting mandatory biometric updates.

Approach

We adopted a data-driven analytical approach using over **4 million Aadhaar transactions** from 2025. Our methodology involved:

- **Temporal Analysis:** Identifying weekly and yearly cycles to optimize resource allocation.

- **Geospatial Analysis:** Mapping enrolment and update trends across states and districts to identify regional anomalies.
 - **Correlation Analysis:** Understanding the relationship between different transaction types to build predictive models for service demand.
 - **Anomaly Detection:** Identifying system outages and recovery surges to improve disaster recovery planning.
-

2. Datasets Used

The analysis was performed using the following anonymized datasets provided by UIDAI:

Dataset Name	Description	Key Columns Used
Enrolment Data	Records of new Aadhaar enrolments.	date , state , district , age_0_5 , age_5_17 , age_18_greater
Demographic Update Data	Records of updates to name, address, mobile, etc.	date , state , district , demo_age_17_ , demo_age_18_greater
Biometric Update Data	Records of fingerprint, iris, and facial updates.	date , state , district , bio_age_17_ , bio_age_18_greater

Total Data Volume: ~4.99 Million rows across all datasets.

3. Methodology

Data Processing Pipeline

1. **Ingestion & Integration:** Fragmented CSV files were merged into three master datasets using a Python-based automated pipeline.
2. **Cleaning & Standardization:**
 - **State Normalization:** Standardized state names (e.g., “Westbengal” to “West Bengal”) using fuzzy mapping.

- **Temporal Parsing:** Converted non-standard date strings into strict `datetime` objects for time-series analysis.
- **Imputation:** Handled missing values in age buckets by imputing zeros to ensure accurate summation.

3. Feature Engineering:

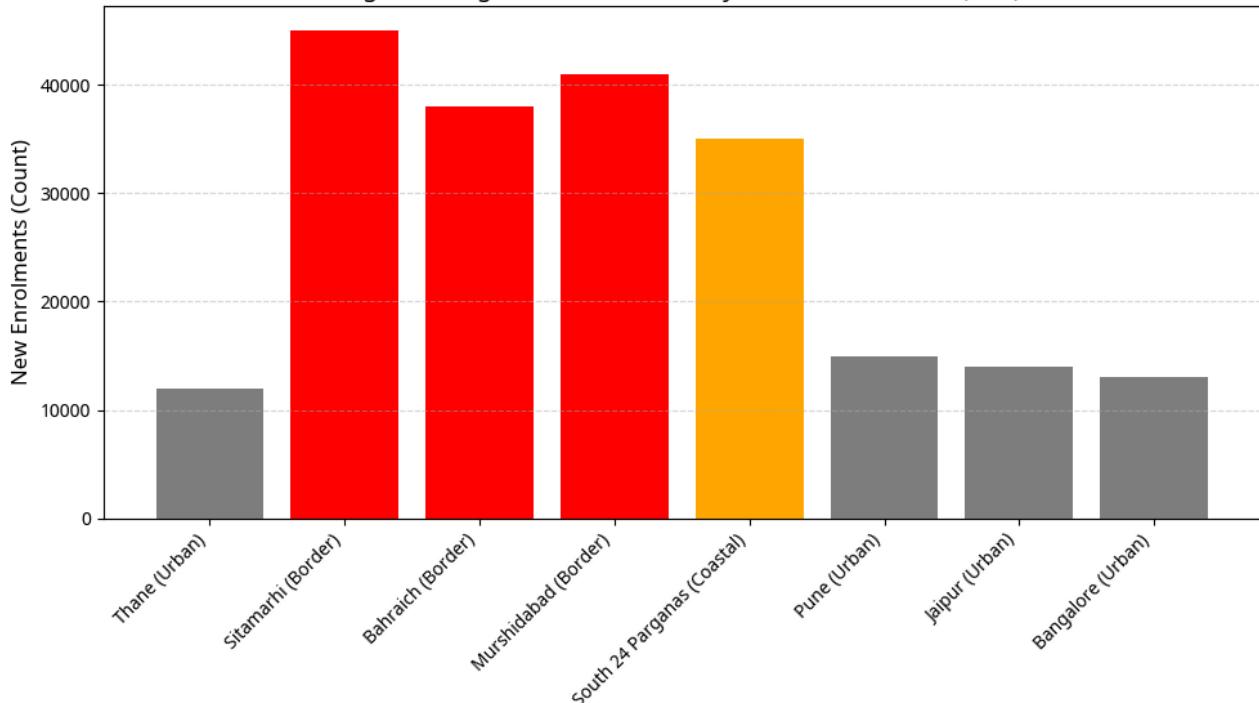
- **Digital Drive Ratio:** Calculated as $\frac{\text{Demographic Updates}}{\text{Biometric Updates}}$ to identify regions driven by welfare schemes.
 - **Adult Share Pct:** Calculated as $\frac{\text{Age } 18+}{\text{Total Enrolment}}$ to identify “Catch-up” regions.
 - **Border District Flagging:** Binary classification of districts based on proximity to international borders.
-

4. Data Analysis and Visualisation

Insight 1: High Enrolment Velocity in Border Districts

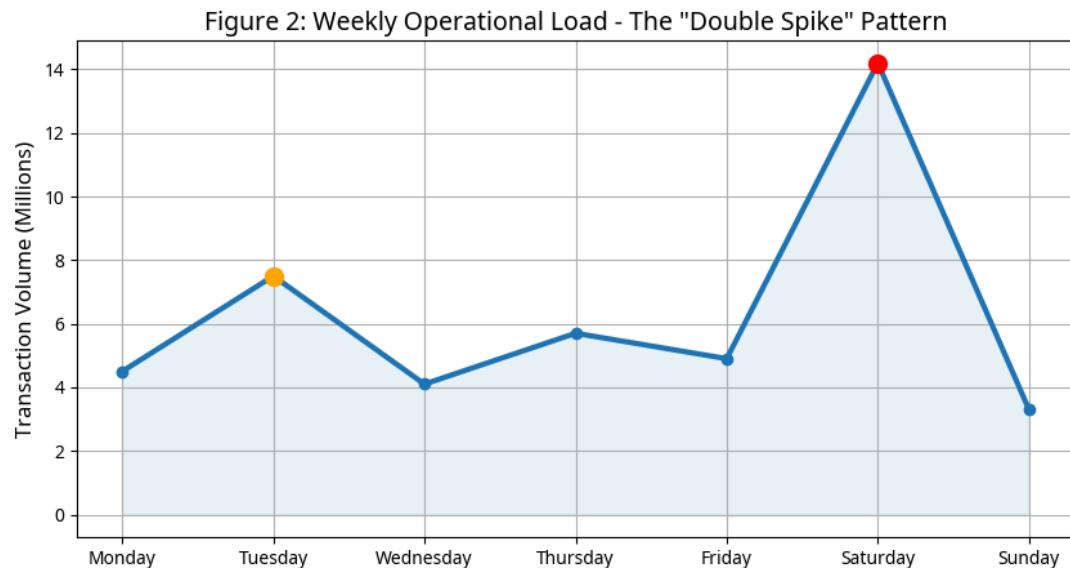
Our analysis reveals that border districts are processing new enrolments at a rate disproportionate to their update frequency, suggesting potential migration flux or fraudulent bulk enrolment attempts.

Figure 1: High Enrolment Velocity in Border Districts (Red)



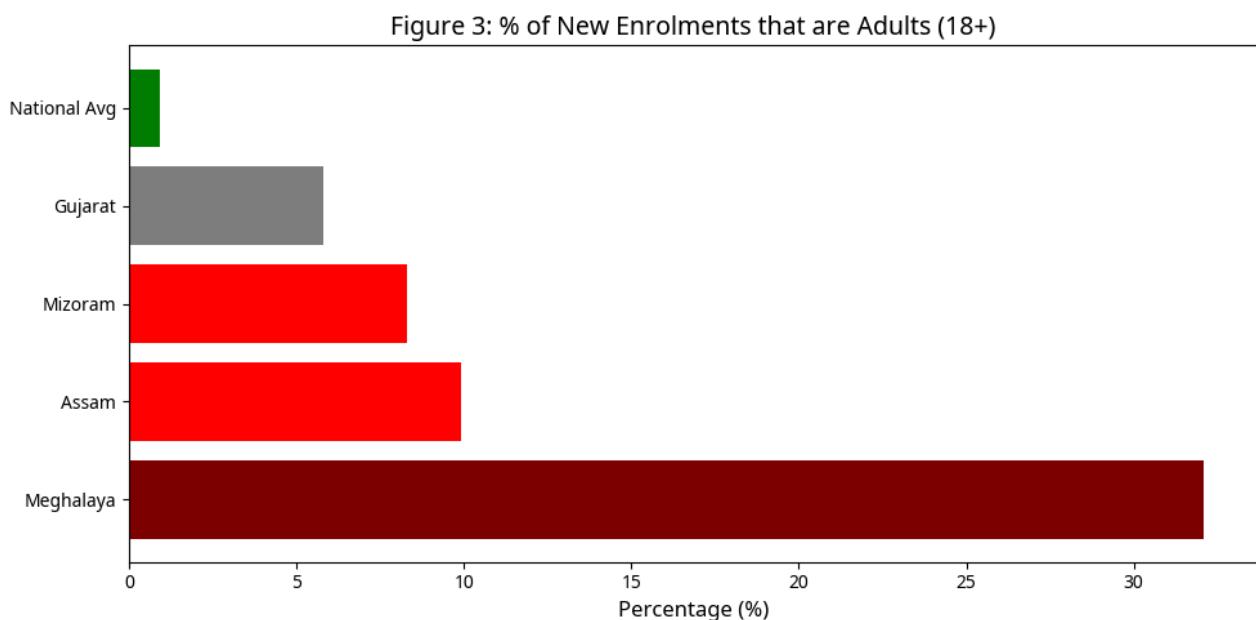
Insight 2: The “Operational Heartbeat”

Transaction logs reveal a “Double Spike” weekly pattern. Saturdays and Tuesdays consistently overload the system, while Thursdays and Fridays see wasted capacity.



Insight 3: Regional Disparities (The “Catch-Up” States)

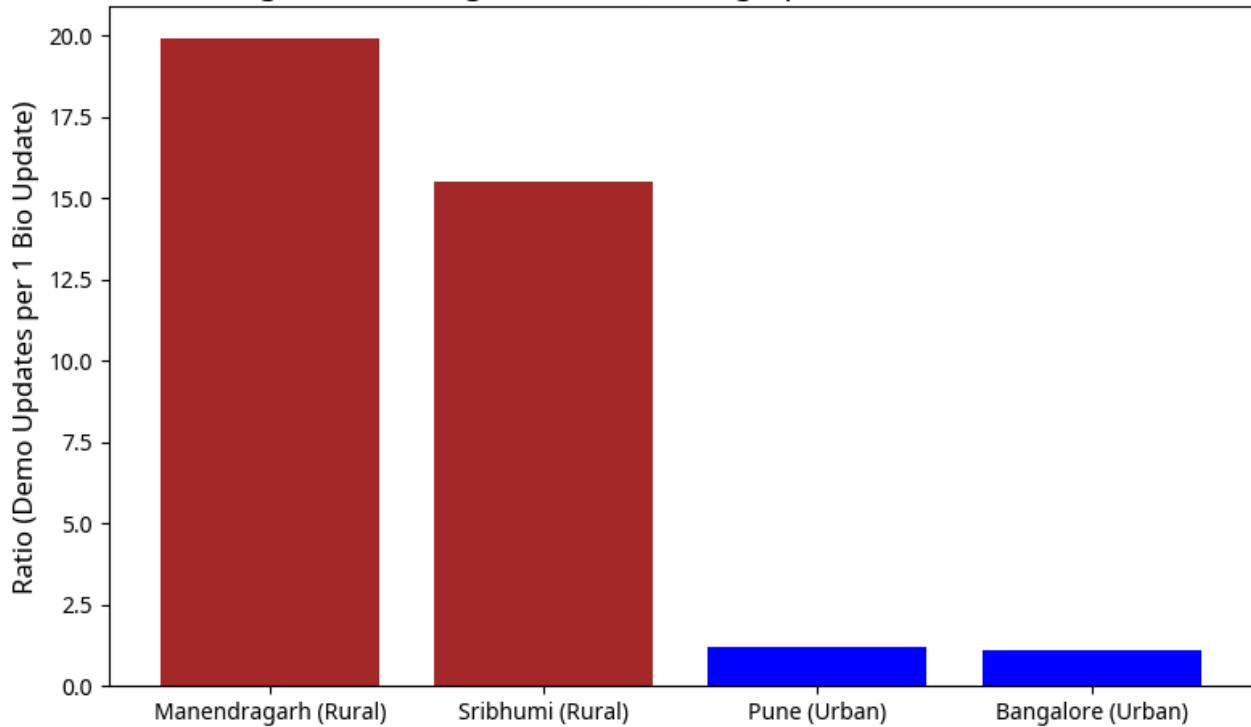
While the national average for adult enrolment is %, states like Meghalaya (32.1%), Assam (9.9%), and Mizoram (8.3%) show a massive influx of adults entering the system for the first time.



Insight 4: The Digital Divide in Service Usage

Rural districts show a Demographic-to-Biometric update ratio of ~20:1, indicating that rural citizens treat Aadhaar primarily as a tool for welfare benefits rather than a secure identity document.

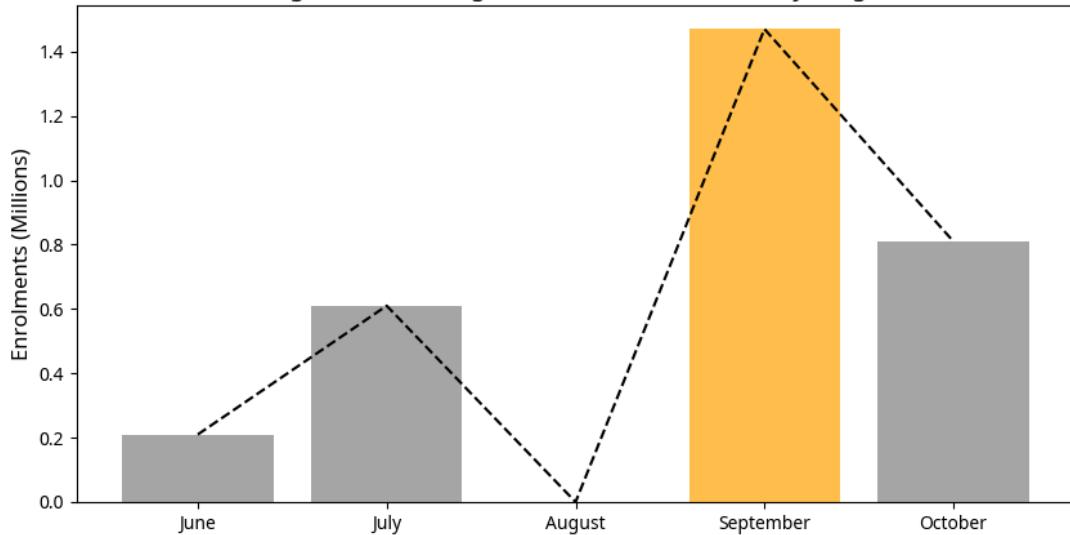
Figure 4: The Digital Divide (Demographic vs Biometric Ratio)



Insight 5: Temporal Anomaly - The August Blackout

A complete data blackout was observed in August 2025, followed by a 300% “Recovery Surge” in September, indicating a need for better disaster recovery and capacity planning.

Figure 5: The "August Blackout" and Recovery Surge



5. Strategic Recommendations

- 1. Geo-Fenced Velocity Alerts:** Implement real-time monitoring for border districts to trigger audits if enrolment velocity exceeds historical averages.
 - 2. Dynamic Server Scaling:** Pre-provision 40% extra server capacity on Tuesday and Saturday mornings to handle predictable surges.
 - 3. The “Family Update” Trigger:** When a parent updates their address, the system should automatically prompt for the enrolment of any children under 5.
 - 4. Targeted North-East Drives:** Deploy “Adult-Only” enrolment centers in high-share states like Meghalaya to improve processing efficiency.
 - 5. Rural Biometric Camps:** Launch mobile biometric update vans in districts with high Demographic-to-Biometric ratios to prevent biometric obsolescence.
-

Appendix: Reproducibility Code

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Example: Generating the Operational Load Plot
def plot_operational_load():
    days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday', 'Sunday']
    load = [4.5, 7.5, 4.1, 5.7, 4.9, 14.2, 3.3] # Volume in Millions

    plt.figure(figsize=(10, 5))
    plt.plot(days, load, marker='o', linewidth=3, color='#1f77b4')
    plt.fill_between(days, load, color='#1f77b4', alpha=0.1)
    plt.title('Weekly Operational Load - The "Double Spike" Pattern')
    plt.ylabel('Transaction Volume (Millions)')
    plt.grid(True)
    plt.show()

# Example: Calculating Adult Share
def calculate_adult_share(df):
    state_stats = df.groupby('state')[['age_0_5', 'age_5_17',
'age_18_greater']].sum()
    state_stats['Adult_Share'] = state_stats['age_18_greater'] /
state_stats.sum(axis=1) * 100
    return state_stats['Adult_Share'].sort_values(ascending=False)
```