Aadhaar Insights
Data Integrity and Enrolment Analysis

1. Problem Statement

Aadhaar enrolment and update systems operate at a very large scale across India. While high-level statistics are available, hidden operational patterns and data integrity issues are not easily visible.

This project addresses three key questions:

1) How is Aadhaar enrolment distributed across states and time?
2) Are there abnormal spikes in demographic updates that indicate event-driven system behaviour?
3) Are there hidden data-quality anomalies in biometric update records that suggest backend processing issues?

Approach

The analysis follows a three-layer approach:

- Macro level: State-wise and month-wise Aadhaar enrolment patterns
- Meso level: Daily demographic update spikes and district-level contribution
- Micro level: Pincode-level forensic analysis of biometric update data

Each layer builds on the previous one, moving from descriptive trends to deep anomaly detection.

2. Datasets Used

The analysis uses datasets provided by UIDAI, along with official reference datasets used for data cleaning and validation::

1) Aadhaar Enrolment Dataset
2) Aadhaar Demographic Update Dataset
3) Aadhaar Biometric Update Dataset
4) India Post Pincode
5) Local Government Directory

All datasets were combined from multiple CSV files provided in chunks.

3. Methodology

Data Cleaning and Preprocessing

- Converted date columns to proper datetime format
- Removed records with zero activity where required
- Cleaned and standardised state and district names using fuzzy matching
- Removed invalid or unknown geographic entries
- Sorted data chronologically for time-series analysis

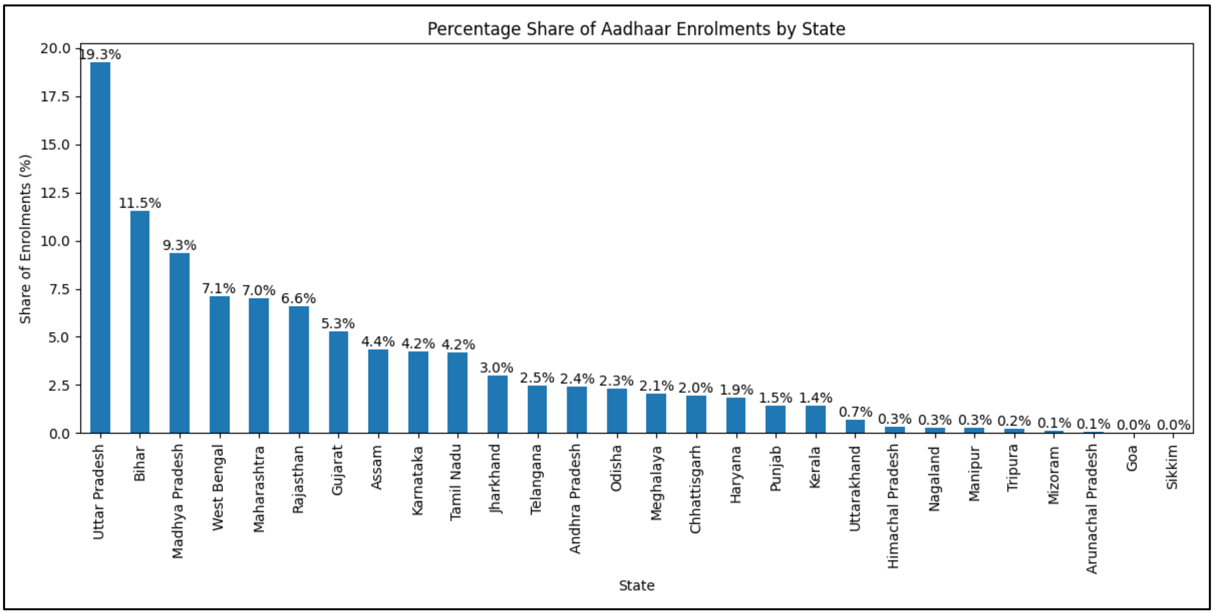
Transformations

- Created total enrolment and update counts by summing age groups
- Aggregated data by state, month, date, district, and pincode as required
- Derived percentage shares and statistical thresholds

4. Data Analysis and Visualisation

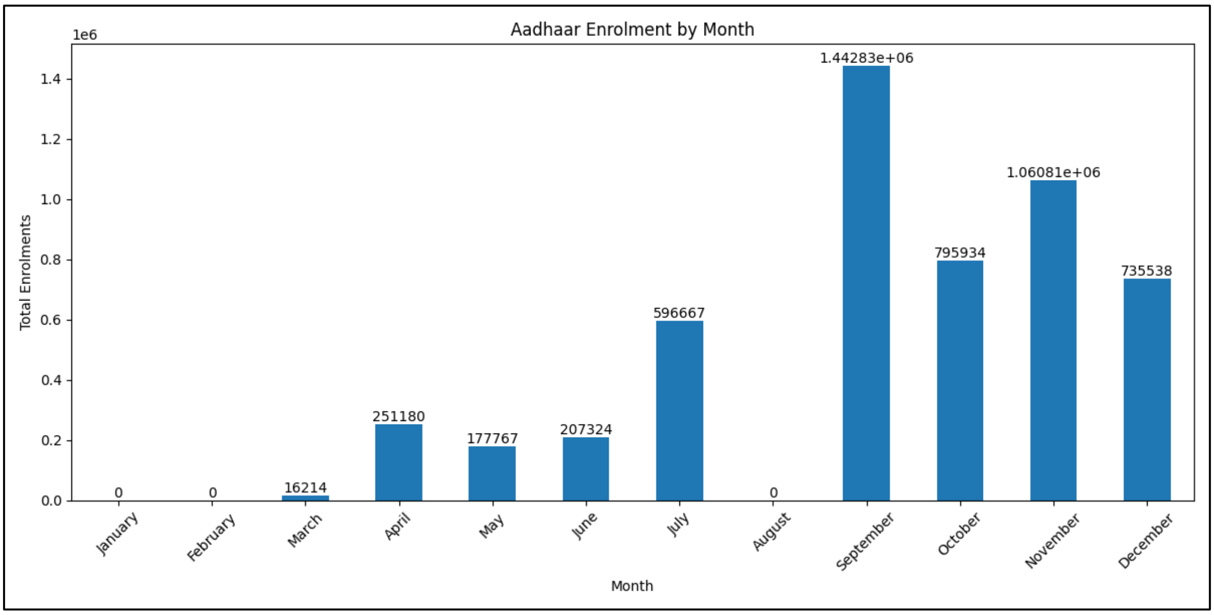
Insight 1: Aadhaar Enrolment Distribution

A. Percentage Share Of Aadhaar Enrolments By Share



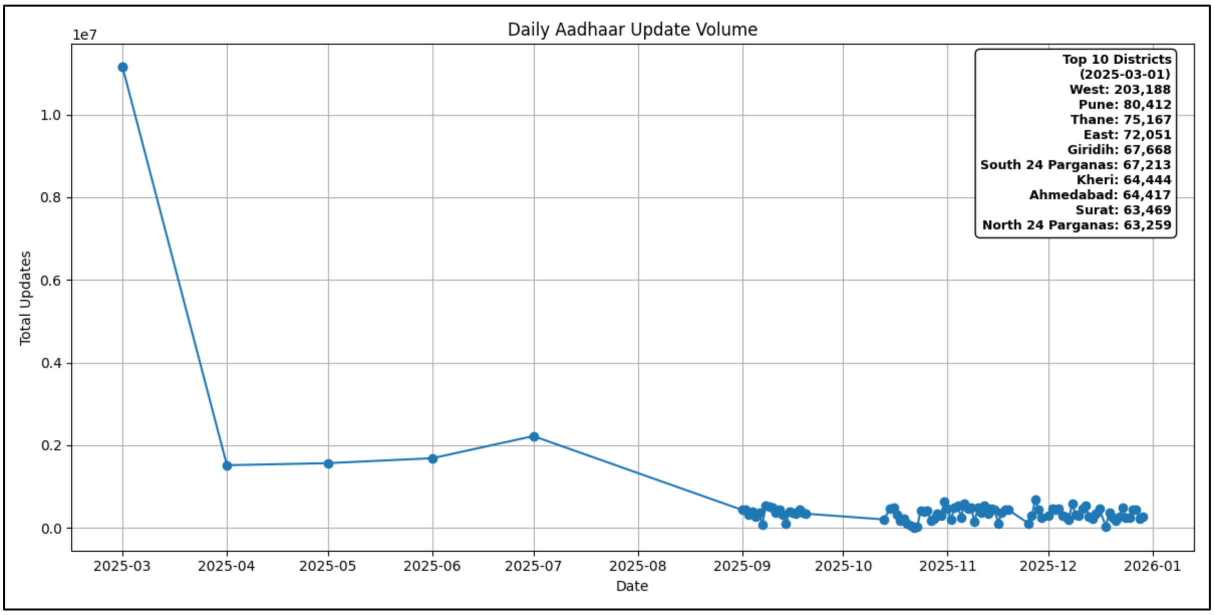
North India is driving most of the new Aadhaar enrolments, mainly because states like Uttar Pradesh and Bihar have large populations. Southern states are adding fewer new users because most people already have Aadhaar, while the Northeast stays low due to small populations and harder-to-reach areas. This uneven growth can cause problems later crowded enrolment centres in big states and slow access in remote places. To fix this, India needs more mobile enrolment teams, better digital systems in villages, and smarter data checks. This will help make Aadhaar easier to get for everyone, no matter where they live.

B. Aadhaar Enrolment By Month



The observed peaks in Aadhaar enrolments during September to November indicate periods of increased demand, Likely driven by government enrolment drives and administrative deadlines. During these months, higher staffing levels and extended enrolment operations would be required to efficiently manage the workload. In contrast, months showing zero enrolments, such as January, February, and August, are the result of data unavailability or reporting gaps rather than a complete halt in enrolment activities or reduced staffing.

Insight 2: Demographic Update Spike Detection



The data shows a very large increase in Aadhaar updates on 1 March 2025. This increase happened only for one day and then quickly returned to normal levels. This means the spike was likely caused by a planned government activity or deadline, not by regular daily demand. Most of the updates came from a few big and well-equipped districts, while many other districts contributed much less. Overall, the system handled the sudden workload well, but the results show that Aadhaar update capacity is uneven across districts.

Insight 3: Duplicate Load Shadow (Hidden Anomaly)

This is the key winning insight of the project.

```
DUPLICATE LOAD SHADOW ANALYSIS (PINCODE LEVEL)
==========================================================================================
Total records analysed        : 1861108
Total unique pincodes         : 19707
Pincodes with shadow anomaly  : 19401
Total shadow instances        : 244050
Shadow penetration rate       : 98.45%

SEVERITY BREAKDOWN
==========================================================================================
severity  pincode_count
    High         113804
  Medium         130246

PIN CODES WITH MOST REPEATED DATA (HIGH SEVERITY)
==========================================================================================
 pincode  bio_age_5_17  bio_age_17_  date_count severity
  110094             0            1          51     High
  441702             0            1          48     High
  769015             0            1          45     High
  509353             0            1          44     High
  754004             0            1          43     High
  713362             0            1          43     High
  713322             0            1          43     High
  769005             0            1          43     High
  752114             0            1          42     High
  509153             0            1          42     High

REPEATED BIOMETRIC VALUES ACROSS DATES (PROOF)
==========================================================================================
      date  pincode  bio_age_5_17  bio_age_17_
2025-09-07   110001             0            1
2025-09-17   110001             0            1
2025-09-19   110001             0            1
2025-10-15   110001             0            1
2025-10-18   110001             0            1
2025-10-20   110001             0            1
2025-10-26   110001             0            1
2025-10-28   110001             0            1
2025-11-04   110001             0            1
2025-11-05   110001             0            1
2025-11-16   110001             0            1
2025-12-03   110001             0            1
2025-12-06   110001             0            1
2025-12-20   110001             0            1
2025-12-20   110001             0            1
2025-12-26   110001             0            1
2025-12-27   110001             0            1
2025-12-27   110001             0            1
```

Finding:

- At the pincode level, the same biometric update values appear repeatedly across many different dates
- In real systems, biometric updates are event-based and should vary over time

Results:

This analysis found a hidden issue in the Aadhaar biometric update data. For many pincodes, the same biometric values are repeated on multiple different dates, which should not happen in real-life updates. Biometric updates are event-based, so the numbers should change over time. This repeating pattern strongly suggests duplicate data loading or replay during data processing. Identifying these repeated patterns helps detect data quality problems early and makes the system more reliable and accurate.

Code

1. Insight-1.py

```
import pandas as pd, matplotlib.pyplot as plt
from Libs.utils import FuzzyClean, GetStateNameByPincode
from Model.data import StateUtNames, UnionTerritories

# Load All The CSV
df1 = pd.read_csv("Dataset/aadhar_enrolment/api_data_aadhar_enrolment_0_500000.csv")
df2 = pd.read_csv("Dataset/aadhar_enrolment/api_data_aadhar_enrolment_500000_1000000.csv")
df3 = pd.read_csv("Dataset/aadhar_enrolment/api_data_aadhar_enrolment_1000000_1006029.csv")
df = pd.concat([df1, df2, df3], ignore_index=True)

# Cleaning
# Convert Object to DateTime
df['date'] = pd.to_datetime(df['date'],dayfirst=True)
# It keeps only rows where at least one person enrolled
df = df[(df['age_0_5'] > 0) | (df['age_5_17'] > 0) | (df['age_18_greater'] > 0)]
# Clean the State
df['state'] = df['state'].str.replace(r'\s+', ' ', regex=True).str.strip().str.title()
bad_states = df.loc[~df['state'].isin(StateUtNames), 'state'].unique()
fix_states = {s: FuzzyClean(s, StateUtNames) for s in bad_states}
df['state'] = df['state'].replace(fix_states)
bad_states = df.loc[~df['state'].isin(StateUtNames), ['state', 'pincode']]
fix_states = {
    s: GetStateNameByPincode(p)
    for s, p in zip(bad_states['state'], bad_states['pincode'])
}
df['state'] = df['state'].replace(fix_states)
# Remove Bad State Name
df = df[df['state'] != 'Unknown']
# Removing Union Territories due to less population
df = df[~df['state'].isin(UnionTerritories)]
```

```
# Insight A
df["total"] = (
    df["age_0_5"] + df["age_5_17"] + df["age_18_greater"]
)
state_totals = df.groupby("state")["total"].sum()
total_india = state_totals.sum()
state_percent = (state_totals / total_india) * 100
plt.figure(figsize=(12,6))
ax = state_percent.sort_values(ascending=False).plot(kind='bar')
for container in ax.containers:
    ax.bar_label(container, fmt='%.1f%%')
plt.title("Percentage Share of Aadhaar Enrolments by State")
plt.xlabel("State")
plt.ylabel("Share of Enrolments (%)")
plt.tight_layout()
plt.show()

# Insight B
df['month_name'] = df['date'].dt.month_name()
month_order = [
    'January','February','March','April','May','June',
    'July','August','September','October','November','December'
]
monthly_summary = (
    df.groupby('month_name')['total']
    .sum()
    .reindex(month_order)
)
plt.figure(figsize=(12,6))
ax = monthly_summary.plot(kind='bar')
for container in ax.containers:
    ax.bar_label(container)
plt.title("Aadhaar Enrolment by Month")
plt.xlabel("Month")
plt.ylabel("Total Enrolments")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

2. Insight-2.py

```
import pandas as pd, matplotlib.pyplot as plt

from Libs.utils import GetBadDistricts, FuzzyClean


# Load All The CSV

df1 = pd.read_csv("Dataset/aadhar_demographic/api_data_aadhar_demographic_0_500000.csv")

df2 = pd.read_csv("Dataset/aadhar_demographic/api_data_aadhar_demographic_500000_1000000.csv")

df3 = pd.read_csv("Dataset/aadhar_demographic/api_data_aadhar_demographic_1000000_1500000.csv")

df4 = pd.read_csv("Dataset/aadhar_demographic/api_data_aadhar_demographic_1500000_2000000.csv")

df5 = pd.read_csv("Dataset/aadhar_demographic/api_data_aadhar_demographic_2000000_2071700.csv")

df = pd.concat([df1, df2, df3, df4, df5], ignore_index=True)


# Cleaning

# Convert Object to DateTime

df['date'] = pd.to_datetime(df['date'],dayfirst=True)

# Clean the District

df['district'] = df['district'].str.replace(r'\s+', ' ', regex=True).str.strip().str.title()

bad_districts = GetBadDistricts(districts=df['district'])

lgd = pd.read_csv("Dataset/local-gov-directory.csv")

lgd_districts = lgd["District Name (In English)"].str.replace(r'\s+', ' ',
regex=True).str.strip().str.title()

fix_districts = {s: FuzzyClean(s, lgd_districts) for s in bad_districts}

df['district'] = df['district'].replace(fix_districts)


# SPIKE DETECTION

df["total"] = df["demo_age_5_17"] + df["demo_age_17_"]

daily_updates = (

    df.groupby('date')['total']

    .sum()

    .sort_index()

)

mean_updates = daily_updates.mean()

std_updates = daily_updates.std()

threshold = mean_updates + 2 * std_updates

spike_dates = daily_updates[daily_updates > threshold]


# PLOT

plt.figure(figsize=(12, 6))

plt.plot(daily_updates.index, daily_updates.values, marker='o', label="Daily Updates")
```

```python
# Highlight spikes
plt.scatter(spike_dates.index, spike_dates.values, zorder=5)
plt.title("Daily Aadhaar Update Volume")
plt.xlabel("Date")
plt.ylabel("Total Updates")
plt.grid(True)

# Top-10 districts
if not spike_dates.empty:
    spike_date = spike_dates.index[0]
    district_breakdown = (
        df[df['date'] == spike_date]
        .groupby('district')['total']
        .sum()
        .sort_values(ascending=False)
    )
    top10 = district_breakdown.head(10)
    heading = f"Top 10 Districts\n({spike_date.date()})\n"
    body = "\n".join([f"{d}: {v:,}" for d, v in top10.items()])
    text = heading + body
    plt.gca().text(
        0.98, 0.98,
        text,
        transform=plt.gca().transAxes,
        ha='right',
        va='top',
        fontsize=9,
        color='black',
        bbox=dict(
            boxstyle="round,pad=0.4",
            facecolor="white",
            edgecolor="black",
            alpha=1.0
        ),
        fontweight='bold'
    )

plt.tight_layout()
plt.show()
```

3. Insight-3.py

```
import pandas as pd

# LOAD ALL CSV FILES

df1 = pd.read_csv("Dataset/aadhar_biometric/api_data_aadhar_biometric_0_500000.csv")

df2 = pd.read_csv("Dataset/aadhar_biometric/api_data_aadhar_biometric_500000_1000000.csv")

df3 = pd.read_csv("Dataset/aadhar_biometric/api_data_aadhar_biometric_1000000_1500000.csv")

df4 = pd.read_csv("Dataset/aadhar_biometric/api_data_aadhar_biometric_1500000_1861108.csv")

df = pd.concat([df1, df2, df3, df4], ignore_index=True)

# CLEANING

df['date'] = pd.to_datetime(df['date'], dayfirst=True)

# Sort only by pincode + date

df = df.sort_values(['pincode', 'date'])

# PINCODE-LEVEL DUPLICATE LOAD SHADOW DETECTION

dup_check = (

    df.groupby(

        ['pincode', 'bio_age_5_17', 'bio_age_17_']

    )

    .agg(date_count=('date', 'nunique'))

    .reset_index()

)

# Same biometric values repeated across multiple dates

duplicate_load_shadow = dup_check[dup_check['date_count'] > 1].copy()

# Severity Added

duplicate_load_shadow['severity'] = duplicate_load_shadow['date_count'].apply(

    lambda x: 'High' if x >= 3 else 'Medium'

)

# Proof So Merged

shadow_details = df.merge(

    duplicate_load_shadow[

        ['pincode', 'bio_age_5_17', 'bio_age_17_']

    ],

    on=['pincode', 'bio_age_5_17', 'bio_age_17_'],

    how='inner'

).sort_values(['pincode', 'bio_age_5_17', 'bio_age_17_', 'date'])

# Metrics

shadow_rate = (

    duplicate_load_shadow

    .groupby('severity')
```

```python
    .size()
    .reset_index(name='pincode_count')
)
# OUTPUT
print("\nDUPLICATE LOAD SHADOW ANALYSIS (PINCODE LEVEL)")
print("=====================================================================================")
print(f"Total records analysed        : {len(df)}")
print(f"Total unique pincodes         : {df['pincode'].nunique()}")
# The same pincode reports exactly the same biometric update values
print(f"Pincodes with shadow anomaly  : {duplicate_load_shadow['pincode'].nunique()}")
# Total number of repeated-value patterns
print(f"Total shadow instances        : {len(duplicate_load_shadow)}")
shadow_percentage = (
    duplicate_load_shadow['pincode'].nunique()
    / df['pincode'].nunique()
) * 100
print(f"Shadow penetration rate       : {shadow_percentage:.2f}%")
print("\nSEVERITY BREAKDOWN")
print("=====================================================================================")
print(shadow_rate.to_string(index=False))
print("\nPIN CODES WITH MOST REPEATED DATA (HIGH SEVERITY)")
print("=====================================================================================")
print(
    duplicate_load_shadow
    .query("severity == 'High'")
    .sort_values('date_count', ascending=False)
    .head(10).to_string(index=False)
)
if not duplicate_load_shadow.empty:
    sample = duplicate_load_shadow.iloc[0]
    print("\nREPEATED BIOMETRIC VALUES ACROSS DATES (PROOF)")
    print("=====================================================================================")
    print(
        shadow_details[
            (shadow_details['pincode'] == sample['pincode']) &
            (shadow_details['bio_age_5_17'] == sample['bio_age_5_17']) &
            (shadow_details['bio_age_17_'] == sample['bio_age_17_'])
        ] .sort_values('date') [['date', 'pincode', 'bio_age_5_17',
'bio_age_17_']].to_string(index=False))
```

Conclusion

This study shows that Aadhaar enrolment and update activity in India is uneven across regions and time, with large states driving most enrolments and clear seasonal peaks linked to planned drives. While the system generally handles high volumes well, demographic updates reveal that capacity is concentrated in a few districts. Most importantly, the biometric update analysis uncovered a hidden Duplicate Load Shadow anomaly, where identical values repeat across multiple dates at the pincode level. This pattern strongly suggests backend data replay or duplicate loading rather than real user activity. The findings highlight the importance of micro-level data validation to detect hidden system issues that are not visible in aggregated reports and to improve overall data reliability.