HPV Prevalence Data – Exploration & Missing Data Analysis

SAINATH R

NIT Raipur

May 30, 2025

Project Background & Replication

- Previous intern: Daljeet used XGBoost on HPV prevalence data.
- Target Variables: `ncc_combined`, `high_cin_combined`, `ncc_16_prevalence`, etc.
- I successfully replicated Daljeet's results for all target variables.
- Daljeet dropped many prevalence, pregnancy, and case-related features.

Daljeet's Setup (ncc_combined)

- Model: XGBoost Regressor
- Dropped Features: All prevalence & cases related
- Parameters:
  - `eta=0.05`, `max_depth=3`, `gamma=0.3`
  - `min_child_weight=4`, `subsample=1.0`
  - `colsample_bytree=0.7`, `lambda=2.0`, `alpha=1.0`
- Test $R^2 = 0.5281$

My Replication: ncc_combined

- Dropped only target-related columns.
- Model: XGBoost
- Test R^2: 0.7416

High CIN Combined: Comparison

Metric	Daljeet	Me
Train MSE	33.3991	0.0024
Train R^2	0.4095	1.0000
Test MSE	172.7375	5.6040
Test R^2	-0.1423	0.9412

My Model Parameters (High CIN Combined)

- `colsample_bytree=1.0`, `learning_rate=0.1`
- `max_depth=4`, `min_child_weight=3`
- `n_estimators=200`, `reg_alpha=0`, `reg_lambda=1`
- `subsample=0.7`

NCC-16 Prevalence: Comparison

Metric	Daljeet	Me
Train R^2	0.8687	0.9932
Test R^2	0.8033	0.7416

My Parameters (ncc_16_prevalence)

- colsample_bytree=1.0, learning_rate=0.1
- max_depth=4, min_child_weight=3
- n_estimators=100, reg_alpha=1, reg_lambda=1
- subsample=0.9

Daljeet Parameters (ncc_16_prevalence)

- gamma = 0.9773, min_child_weight = 4
- subsample = 0.9999, colsample_bytree = 0.5012
- colsample_bylevel = 0.9995
- lambda = 9.9672, alpha = 0.0050

Additional Contributions

- Scraped data for:
  - Region-wise HPV prevalence in India.
  - Top cities with high cervical cancer incidence.
- This enriched the dataset and gave potential for future regional model training.

Learning Difficulty Calculation (Resampling Strategy)

This function estimates how hard it has been to learn a sample *i* over time.

Inputs:

- y_{true}: Actual label
- pred_hist: List of predictions for the sample across training iterations

Logic:

- For each iteration t:
    - Compute previous error: $|\text{pred}_{t-1} - y_{\text{true}}|$
    - Compute current error: $|\text{pred}_{t} - y_{\text{true}}|$
    - If error reduced ⇒ **learning**; else ⇒ **unlearning**
- Track total learning and unlearning over time

Difficulty Score:

$$\text{Difficulty} = \frac{c + \text{Total Unlearning}}{c + \text{Total Learning}}$$

- c is a small constant to avoid division by zero

Results: NCC Combined

- **Train Performance:**
    - RMSE $= 3.0413$
    - $R^2 = 0.8049$
- **Test Performance:**
    - RMSE $= 4.4$
    - $R^2 = 0.7379$

Results: NCC 16 Prevalence

- **Train Performance:**
    - RMSE $= 6.4223$
    - $\mathrm{R}^2 = 0.6445$
- **Test (Gold Standard) Performance:**
    - RMSE $= 7.3066$
    - $\mathrm{R}^2 = 0.2527 \pm 0.6200$

Results: High CIN 16 Prevalence

- **Train Performance:**
  - RMSE $= 10.1983$
  - $R^2 = 0.4931$
- **Test (Gold Standard) Performance:**
  - RMSE $= 9.6869$
  - $R^2 = 0.2073$

parity plot: NCC Combined

parity plot: NCC 16 Prevalence

parity plot: High CIN 16 Prevalence