Register Number	23BCE1996	Subject Code	BCSE309P
Name	Aatreyee	**Subject Name**	Software Lab
Programme	B.Tech	**Slot**	L15+L16
Course	CSE	**Semester**	6
Date	25/02/2026	**Assignment No.**	5

Project Title: getMAP – Spatial Downscaling of Tropospheric NO_2 Satellite Air Quality Maps

1. Problem Statement

Plenty of coarse-resolution satellite-based air quality data is available; however, its spatial granularity is insufficient for localized air quality assessment and monitoring. The problem statement aims to use machine learning techniques to enhance coarse-resolution satellite data—such as tropospheric NO_2 measurements—into finer spatial resolution maps that better capture local variability. While individual tools exist for satellite data processing and machine learning modelling, no comprehensive, validated, end-to-end solution exists for NO_2 spatial downscaling. *getMAP* addresses this gap.

2. Objectives

- To develop an end-to-end AI/ML pipeline that performs spatial downscaling of coarse-resolution satellite-derived tropospheric NO_2 data into fine-resolution air quality maps.
- To enhance spatial resolution by up to 8× while preserving physically meaningful spatial patterns present in the original satellite observations.
- To apply and compare multiple machine learning models (Random Forest, XGBoost, and Gradient Boosting) for spatial regression and identify the most effective approach for NO_2 downscaling.
- To validate the downscaled satellite predictions against independent ground-based NO_2 measurements from CPCB monitoring stations using standard statistical metrics.

- To provide an interactive, user-friendly visualization and analysis interface for comparing original and downscaled maps, inspecting feature importance, and exporting results for reporting and research use.

3. Methodology / Approach

1. Data Acquisition and Preprocessing

- Coarse-resolution satellite NO_2 data (GeoTIFF format) from sources such as TROPOMI/Sentinel-5P or OMI/Aura is uploaded to the system.
- Missing or cloudy pixels (NaN values) are handled using either spatial interpolation or mean filling prior to model training.
- Optional ground-station NO_2 measurements from CPCB (CSV format) are uploaded for validation and stored in a local SQLite database.

2. Feature Engineering

- Each valid satellite pixel is transformed into feature vector

3. Model Training

- The processed dataset is split into 80% training and 20% testing pixels to ensure validation on unseen spatial data.
- One of the selected ML models (Random Forest, XGBoost, or Gradient Boosting) is trained after feature normalization using a standard scaler.
- Feature importance is extracted post-training to analyze model behavior and spatial dependencies.

4. Spatial Downscaling and Prediction

- The trained ML model refines an upsampled interpolated grid by predicting NO_2 values at each fine-resolution pixel based on engineered features.

5. Validation and Evaluation

- Model predictions are evaluated using statistical metrics including MSE, RMSE, MAE, R^2 score, and bias.
- If ground station data is provided, predicted values are compared against CPCB measurements for external validation.

6. Visualization and Export

- Interactive Plotly heatmaps display side-by-side comparisons of original and downscaled NO_2 maps using a common color scale.
- Final downscaled grids and evaluation metrics can be exported as CSV files for documentation and reporting.

4. Work Completed So Far

Work Completed So Far

- Implemented the core end-to-end pipeline for spatial downscaling of satellite-based NO_2 data, including data upload, preprocessing, feature engineering, model training, prediction, and visualization.
- Successfully integrated three machine learning models (Random Forest, XGBoost, and Gradient Boosting) with configurable parameters and model switching via the user interface.
- Developed spatial feature engineering (coordinates, local neighborhood statistics, and gradients) and validated its effectiveness through feature importance analysis.
- Built interactive visualizations for original and downscaled maps, including side-by-side comparison and metric dashboards.
- Implemented basic validation using statistical metrics (MSE, RMSE, MAE, R^2, Bias) and support for uploading CPCB ground-station data.

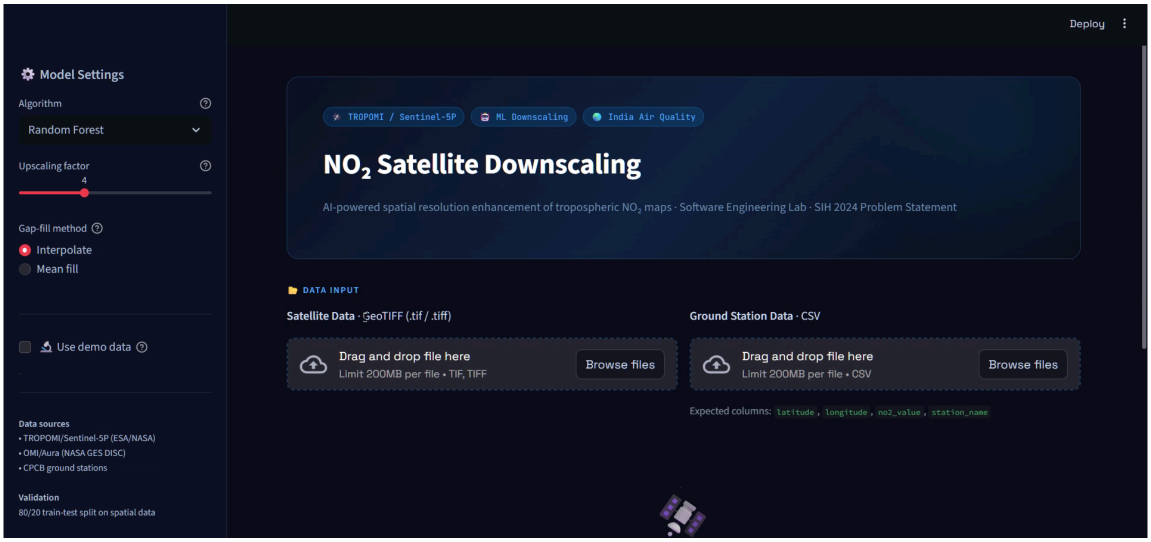
Work Remaining / In Progress

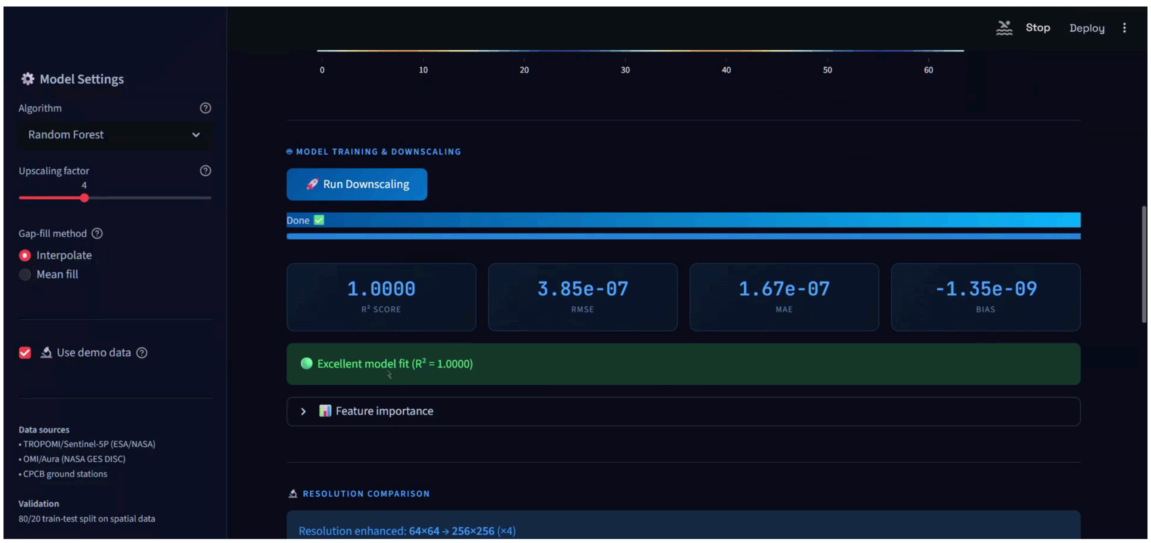
- Extensive large-scale validation across multiple regions and time periods is pending.
- Further hyperparameter tuning and robustness testing under varying cloud cover and data sparsity conditions remains to be completed.
- Performance optimization and memory efficiency improvements for large GeoTIFF inputs are yet to be implemented.
- Final documentation, testing, and deployment-level hardening are in progress.

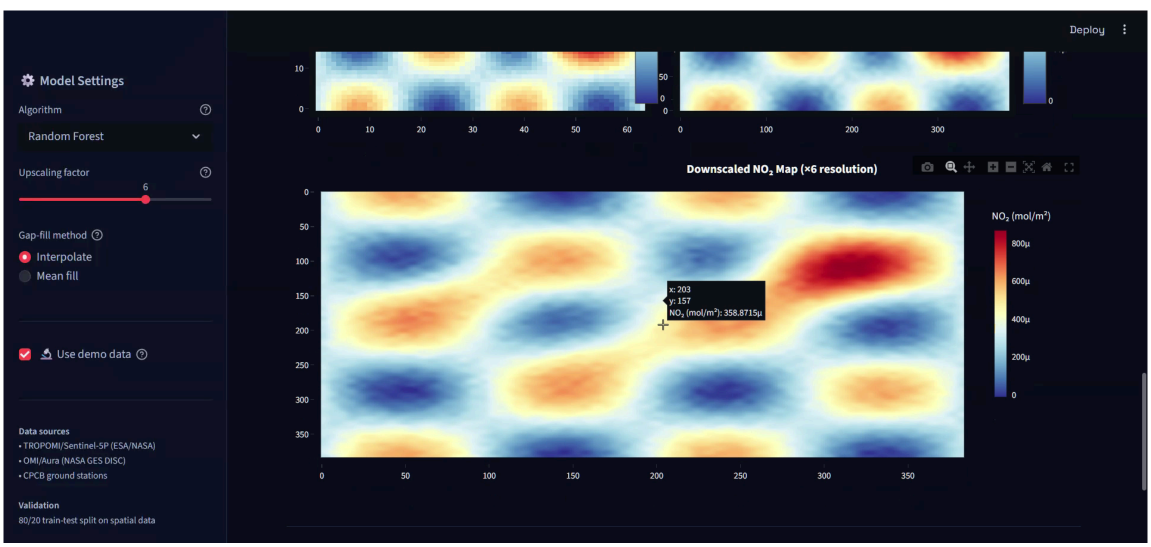
5. Proof of Implementation (GitHub Link)

GitHub Link : https://github.com/aatreyee-23bce1996/getMAP

Screenshots :







6. Demo Video (max of 10 mins)

Demo Video Link : https://www.loom.com/share/39457f2da0c24181b290a89c3efd5b9f