**Project Concept 2**

**Predicting Air Quality and Emission Hotspots Using Machine Learning**

* **Problem Statement**

This project aims to predict air quality levels and identify emission hotspots using machine learning models. It integrates datasets on air pollutants (PM2.5, PM10, NOx, SOx), weather parameters (temperature, wind speed, humidity), and traffic/industrial activity.

Public datasets from sources like EPA, CPCB, WHO, OpenWeather will be used. Outcomes include accurate air quality forecasts, hotspot mapping, Results will support urban planning and public health interventions through an interactive dashboard and real-time analytics

* **Problem Definition**

- Define the scope of the project:

* Predict air quality levels and identify emission hotspots.
* Identify the pollutants of interest (PM2.5, PM10, NOx, SOx).

- Objectives:

* Develop accurate models for forecasting and visualize results in an interactive dashboard.
* **Data Collection**

- Sources:

* + EPA, CPCB, WHO: Air quality data.
  + OpenWeather API: Weather parameters (temperature, humidity, wind speed, etc.).
  + Local government or traffic data portals: Traffic and industrial activity.

- Data Types:

* + Time-series data (e.g., daily pollutant concentrations, weather data).
  + Geospatial data (e.g., locations of traffic and industrial activity).
* **Data Preprocessing**

**-** Data Cleaning:

- Handle missing values and outliers.

- Ensure consistency in units across datasets.

- Data Transformation:

- Normalize pollutant concentrations and weather parameters.

- Create time-lagged features for time-series forecasting.

- Data Integration:

- Merge air quality, weather, and traffic/industrial datasets based on time and location.

- Visualize pollutant trends and correlations with weather parameters.

- Identify patterns in hotspot locations.

* **Model Selection and Development**

- Prediction Models:

- Start with regression models like Linear Regression and Random Forest.

- Convert continuous AQI values into categories (e.g., "Good," "Moderate," etc.) based on predefined thresholds.

- Implement time-series models like ARIMA, LSTM, or Prophet for temporal predictions.

- Hotspot Detection Models:

- Use clustering algorithms like K-Means or DBSCAN for spatial analysis.

- Evaluation Metrics:

- AQI prediction: MAE, RMSE, R².

- AQI Category Prediction: Accuracy Score, Confusion Matrix.

* **Model Training and Testing**

- Split the data into training, validation, and testing sets.

- Use cross-validation for robust evaluation.

- Fine-tune hyperparameters to optimize model performance.

* **Design & Deployment**

- Employ APIs for real-time data ingestion (e.g., OpenWeather API).

- Use tools like Tableau, Power BI for creating the interactive dashboard.

- Deploy the dashboard on a web server using Flask/Django (backend) and React/Python GUI (frontend).

* **Monitoring and Maintenance**

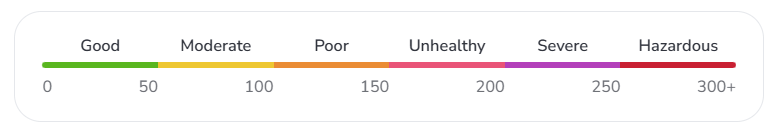
- Set up alerts for anomalies in air quality forecasts.

- Regularly update models with new data to maintain accuracy.

- Collect user feedback to improve the dashboard’s usability.

* **Project Outcomes**

- Air Quality Predictions:

 - Provide daily forecasts for pollutant levels and AQI Category Prediction.

- Hotspot Mapping:

- Visualize high-pollution areas on the dashboard.

- Policy Recommendations:

- Share insights for urban planning and public health interventions.

* **Team Collaboration Plan**

| Key Deliverables | Names |

|-----------------------------------|------------------------------------------------|

| Data Collection

| Data Preprocessing

| Model Development

| Dashboard Development

| Deployment and Testing