|  |  |
| --- | --- |
| Phase 2 | Air Quality Analysis in TamilNadu |

**Innvoation:**

Incorporating machine learning algorithms to improve the accuracy of air quality predictive models in Tamil Nadu (TN) is an excellent way to address air quality concerns and make more precise forecasts. Here's a step-by-step approach on how we are going to leverage machine learning for this purpose:

1. **Data Collection:**
   * Gather historical air quality data from monitoring stations across Tamil Nadu. Include variables such as PM2.5 levels, PM10 levels, NO2, SO2, CO, O3, temperature, humidity, wind speed, and wind direction.
   * Collect data on local weather patterns, industrial activities, traffic congestion, and other relevant factors that can influence air quality.
2. **Data Preprocessing:**
   * Clean and preprocess the data by handling missing values, outliers, and formatting issues.
   * Aggregate data by location and time (e.g., hourly or daily averages) to create a structured dataset for model training.
3. **Feature Engineering:**
   * Engineer features that can capture temporal patterns, seasonality, and external factors affecting air quality, such as public holidays, festivals, and industrial shutdowns.
   * Creating lag features to capture historical trends.
4. **Select Machine Learning Algorithms:**
   * Choose appropriate machine learning algorithms for air quality prediction. Time series forecasting models like ARIMA, SARIMA, or machine learning algorithms like Random Forest, XGBoost, or Long Short-Term Memory (LSTM) recurrent neural networks are commonly used for such tasks.
5. **Model Training:**
   * Split the dataset into training, validation, and test sets.
   * Train the selected machine learning models using historical air quality and environmental data.
   * Optimize hyperparameters and fine-tune the models to achieve the best performance.
6. **Evaluation Metrics:**
   * Evaluation metrics for air quality prediction, such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or R-squared (R2) to assess model accuracy.
7. **Cross-Validation:**
   * Implement cross-validation techniques to ensure the model's robustness and prevent overfitting.
8. **Real-Time Data Integration:**
   * Set up a data pipeline to collect real-time air quality and environmental data from monitoring stations and weather sources.
   * Continuously update the model with new data to keep it accurate and relevant.
9. **Model Deployment:**
   * Deploy the trained model in a production environment where it can generate real-time air quality predictions.
   * Create a user-friendly interface or API for stakeholders and the public to access air quality forecasts.