**Air Quality Forecasting with LSTM**

**Introduction**

This document outlines the process and methodology used for air quality forecasting in Delhi using machine learning techniques, primarily focusing on the Long Short-Term Memory (LSTM) model. The task involves downloading air quality data from a CPCB station in Delhi for two years, setting up Ollama locally or on Colab, and showcasing the performance of the LSTM model for forecasting air quality for the next 24 hours. Additionally, insights regarding the LSTM model's performance and efficient ways to visualize the results using Matplotlib are provided. While attempts were made to explore Language Model (LLM) approaches, they did not succeed as anticipated**.**

**Data Source**

* Air quality data from a CPCB station in Delhi for two years.
* Link: [CPCB Automatic Monitoring Data](https://cpcb.nic.in/automatic-monitoring-data/)

**Workflow Overview**

1. Data Acquisition:
   * Download air quality data from the specified CPCB station for the required period.
2. Setup Ollama:
   * Install and configure Ollama locally or on Colab for model training and evaluation.
3. Model Training and Evaluation:
   * Implement the LSTM model for time-series forecasting.
   * Prepare the data for training and evaluation, including data preprocessing and feature engineering.
   * Train the LSTM model using the prepared data.
   * Evaluate the performance of the LSTM model using metrics such as accuracy, RMSE, and MAE.
4. Visualization of Results:
   * Utilize Matplotlib for visualizing the forecasted air quality data.
   * Showcase insights and comparisons between predicted and actual air quality values.

**Code Execution**

* Import necessary libraries including pandas, numpy, matplotlib, and scikit-learn.
* Load the air quality data into a pandas DataFrame.
* Prepare the data for training and evaluation, including data preprocessing and feature engineering.
* Train the LSTM model for air quality forecasting.
* Evaluate the performance of the LSTM model based on metrics such as accuracy, RMSE, and MAE.
* Visualize the forecasted air quality data using Matplotlib.

**Insights and Analysis**

* Evaluate the performance of the LSTM model based on key metrics.
* Analyze the strengths and weaknesses of the LSTM model for air quality forecasting.
* Provide insights into potential improvements or future directions for enhancing the forecasting accuracy and robustness.

**Conclusion**

In conclusion, while attempts were made to explore Language Model (LLM) approaches for air quality forecasting, they did not yield the anticipated results. However, the LSTM model serves as a powerful tool for air quality forecasting, providing valuable insights into the dynamics of air pollution in Delhi. Through effective data preprocessing, model training, and evaluation, the LSTM model demonstrates its capability to predict air quality levels with reasonable accuracy. By visualizing the forecasted air quality data using Matplotlib, stakeholders can gain deeper insights into the trends and patterns of air pollution, enabling informed decision-making and policy formulation for air quality management.