**Pollution Prediction LSTM Model:**

**Development Report**

1. **Project Overview**

**Objective**: Develop a time-series prediction model for air quality and pollution trends using Long Short-Term Memory (LSTM) neural networks.

2. **Model Development**

2.1 **Data Sources**

The model integrates data from three primary sources:

* Pollution data
* Air quality data
* Weather forecast data

2.2 **Data Preprocessing**

Key Preprocessing Steps:

* Data merging across multiple datasets
* Timestamp alignment
* Numeric type conversion
* Missing value handling
* Forward fill
* Backward fill
* Normalization using MinMaxScaler

**2.3 Model Architecture**

LSTM Neural Network Configuration:

Architecture: Two-layer LSTM with Dropout

Input Layer:

* 32 LSTM units
* Sequence length: 24 time steps

Hidden Layer:

* 32 LSTM units
* Dropout rate for regularization
* Output Layer: Dense layer matching input feature dimensions

**2.4 Hyperparameter Optimization**

Optimal Hyperparameters:

* LSTM Units: 32
* Dropout Rate: 0.1
* Learning Rate: 0.01
* Epochs: 30
* Batch Size: 16

**3. Experiment Tracking**

**3.1 MLflow Integration**

* **Tracked Metrics:**
  + Test Loss
  + Test Mean Absolute Error (MAE)
  + AQI Mean Squared Error (MSE)
  + AQI Mean Absolute Error
* **Logged Parameters:**
  + - Model hyperparameters
    - Training configuration details

**3.2 Artifact Management**

* Model saved as `optimized\_pollution\_lstm\_model.h5`
* Predictions exported to `optimized\_predicted\_aqi\_trends.csv`
* Experiment details tracked in MLflow

**4. Performance Evaluation**

**4.1 Prediction Capabilities**

The model provides:

* Time-series predictions for multiple pollution indicators
* AQI (Air Quality Index) risk categorization
* Hourly pollution trend forecasts

**4.2 Risk Classification**

AQI Risk Categories:

* Good (AQI < 50)
* Fair (50 ≤ AQI < 100)
* Moderate (100 ≤ AQI < 150)
* Poor (150 ≤ AQI < 200)
* Very Poor (AQI ≥ 200)

**5. Technical Implementation Details**

5.1 Technology Stack

* Python 3.x
* TensorFlow/Keras
* Pandas
* NumPy
* Scikit-learn
* MLflow

**5.2 Key Libraries**

* Deep Learning: TensorFlow
* Machine Learning: Scikit-learn
* Data Manipulation: Pandas, NumPy
* Experiment Tracking: MLflow

**6. Recommendations for Future Improvements**

1. Incorporate more diverse data sources

2. Experiment with hybrid model architectures

3. Implement real-time prediction capabilities

4. Develop more granular AQI risk classification

5. Integrate advanced feature engineering techniques

**7. Conclusion**

The developed LSTM model demonstrates robust capabilities in predicting pollution trends with comprehensive tracking and evaluation mechanisms. The MLflow integration ensures transparency and reproducibility of the experiment.

Project Status: Successfully developed and optimized

---

Note: Detailed experiment logs and model artifacts are available through the MLflow tracking interface.