মেট্রোপলিটন ইউনিভার্সিটি
Metropolitan University

Report on

AtmoSense: Air Quality Prediction System

Course Title: Artificial Intelligence Lab
Course Code:CSE 422

Submitted To:
Ishrar Nazah Chowdhury
Lecturer
Department of Computer Science & Engineering
Metropolitan University,Sylhet.

Submitted By:
Name: Mahmuda Sultana
ID:231-115-018

Department of Computer Science and Engineering
Metropolitan University,Sylhet.

Date of Submission:11/12/2025

AtmoSense: Air Quality Prediction System

Project Report:

1. Motivation:

Air pollution stands as one of the most critical challenges to global public health and environmental sustainability. The World Health Organization (WHO) reports that **9 out of 10 people breathe polluted air**, contributing to approximately **7 million premature deaths annually**.

While traditional monitoring stations provide accurate data, they are hindered by:

- High costs
- Sparse deployment
- Reporting latencies

AtmoSense bridges this gap by leveraging **machine learning** to democratize air quality monitoring. By utilizing affordable sensor data to predict **Carbon Monoxide (CO) levels**—a key proxy for overall pollution—this system aims to:

- Overcome the limitations of sparse sensor networks through predictive modeling
- Provide actionable health insights by classifying air quality into interpretable categories
- Enable proactive decision-making for vulnerable populations through early warnings

2. Objectives:

⇨Primary Objectives

- **Predictive Modeling:** Develop robust regression models (Linear Regression via Closed-Form and Gradient Descent) to forecast continuous CO concentrations.

- **Health Classification:** Implement classification algorithms (K-Nearest Neighbors, Logistic Regression) to categorize air quality indices (Good, Moderate, Bad).
- **End-to-End Pipeline:** Architect a seamless data pipeline encompassing ingestion, cleaning, transformation, and inference.
- **Explainability:** Ensure model transparency to allow stakeholders to understand the drivers of pollution levels.

⇨Secondary Objectives

- **Data Integrity:** Engineer a rigorous preprocessing strategy to handle sensor drift and missing values in real-world time-series data.
- **Exploratory Analysis:** Uncover latent temporal and meteorological patterns driving pollution spikes.
- **Scalability:** Design a modular code structure allowing for the future integration of additional pollutants (e.g., PM2.5, NOx).

3. Methodology:

➪A. Data Acquisition

- **Source:** UCI Machine Learning Repository - *AirQualityUCI*
- **Context:** Multisensor device deployed in an Italian city (March 2004 – April 2005)
- **Hardware:** Metal oxide chemical sensors
- **Features:** 15 distinct features including:
    - **Pollutants:** CO, Non-methanic Hydrocarbons, Benzene, NOx, NO2
    - **Atmospherics:** Temperature (T), Relative Humidity (RH), Absolute Humidity (AH)

➪B. Data Preprocessing Pipeline

Reliable prediction requires high-quality input. The raw data underwent a **multi-stage refinement process**:

1. Cleaning & Imputation:

- **Thresholding:** Removed features with >40% missing data
- **KNN Imputation:** Applied K-Nearest Neighbors (k=5) to fill gaps based on local data topology

2. Outlier Management:

- **Method:** Interquartile Range (IQR)
- **Capping:** Extreme outliers capped at 1.5 × IQR to prevent model skew

3. Feature Engineering:

- **Temporal Decomposition:** Extracted Year, Month, Day, Hour, DayOfWeek
- **Cyclical Encoding:** Represented time features using sine/cosine functions (e.g., Hour_sin, Hour_cos)
- **Contextual Flags:** Created binary features: IsWeekend, IsRushHour, IsNight

4. Data Splitting Strategy:

- **Regression:** 80/20 random split
- **Classification:** 80/20 stratified split

➪C. Model Training & Architecture

1. Regression (Continuous Target: CO Level)

- **Linear Regression (Scikit-Learn):** Baseline using Ordinary Least Squares (OLS)
- **Gradient Descent (Custom Implementation):**
  - Learning Rate (α): 0.01
  - Iterations: 1000
  - Manual partial derivative updates to minimize Cost Function J(θ)

2. Classification (Categorical Target: Air Quality Index)

- **K-Nearest Neighbors (KNN):**
  - k=5 via 5-fold cross-validation
  - Distance-weighted voting
- **Logistic Regression:**
  - One-vs-Rest (OvR) strategy for multi-class classification
  - L2 regularization to prevent overfitting

4. Results and Discussion:

➪A. Regression Performance

Both implementation approaches yielded high fidelity in tracking pollution trends.

Model	MSE	R2 Score	Analysis
Linear Regression (SKLearn)	**0.142**	**0.874**	Best performance; highly efficient.
Gradient Descent (Custom)	0.145	0.871	Near-identical performance validates the custom algorithm logic.

Key Insight: An R^2 score of >0.87 implies the model captures **87% of the variance in CO levels**, proving linear approximation is effective for this sensor array

➪B. Classification Performance

The system was tested on its ability to bucket air quality into *Good*, *Moderate*, and *Bad*.

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score
KNN (k=5)	89.2%	0.891	0.892	0.891
Logistic Regression	**90.4%**	**0.903**	**0.904**	**0.903**

Interpretation: Logistic Regression slightly outperformed KNN (90.4% vs 89.2%), indicating that decision boundaries between air

quality classes are largely linear in the high-dimensional feature space.

5. Conclusion and Future Scope:

AtmoSense demonstrates that high-accuracy air quality monitoring can be achieved without expensive, industrial-grade infrastructure. By combining robust data preprocessing with interpretable machine learning models, the system attained over **90% classification accuracy and an R^2 score exceeding 0.87 in regression** tasks.

Key Achievements:

1. **Dual-Pipeline Architecture:** Successfully integrated regression and classification workflows.
2. **Resilience:** The system is robust against sensor noise due to advanced imputation and outlier handling.
3. **Explainability:** The use of linear and distance-based models allows for clear interpretation of *why* an alert is triggered.

Future Roadmap:

- **Deep Learning Integration:** Implement LSTM (Long Short-Term Memory) networks to better capture long-term sequential dependencies.
- **IoT Deployment:** Port the trained model to a Raspberry Pi/Arduino environment for real-time edge inference.
- **Spatial Expansion:** Incorporate GPS and meteorological API data to create a city-wide pollution heatmap.