Milestone 4: API Deployment – PM2.5 Forecasting

# Objective

To build and deploy a FastAPI-based RESTful service that exposes the trained linear regression model for forecasting PM2.5 concentration levels.

# API Architecture

Framework: FastAPI

Routing: Modular routers defined under `api/v1/endpoints`

Model: Pretrained Linear Regression loaded with `joblib`

Request/Response Models: Pydantic-based schemas located in `api/v1/schemas`

# Endpoint Specification

- URL: `/v1/predict/predict`

- Method: POST

- Input: JSON body with lag and seasonal features:

* • pm25\_lag\_1, pm25\_lag\_2, pm25\_lag\_3
* • year, month, sin\_month, cos\_month

- Output: Predicted PM2.5 value (float)

# Configuration

Model path and other settings are stored in a central `core/config.py` module for maintainability.

Routing is registered via `api/v1/\_\_init\_\_.py` and included once in `main.py` using `include\_router`.

# Example Request

POST /v1/predict/predict  
Content-Type: application/json  
  
{  
 "pm25\_lag\_1": 8.5,  
 "pm25\_lag\_2": 8.2,  
 "pm25\_lag\_3": 9.0,  
 "year": 2024,  
 "month": 6,  
 "sin\_month": 0.5,  
 "cos\_month": 0.87  
}

# Result

Predicted PM2.5 value is returned as a float in the response JSON.

# Conclusion

This deployment milestone marks the successful exposure of the trained model via a robust and maintainable FastAPI backend. With modular routing, centralized configuration, and Pydantic validation, the system is now ready for real-world integration and automated testing.