Agile Development Model

Project ID: SIH1734
Name: Aatreyee
Registration Number: 23BCE1996
Title: AI-based Downscaling of Satellite Air Quality Data

1. Introduction

The Agile development model is adopted for this project to support iterative development, continuous experimentation, and flexibility in model improvement. The project focuses on downscaling low-resolution satellite-based air quality data into higher spatial resolution maps using Machine Learning techniques. Since ML-based systems require repeated training, validation, and tuning, Agile allows incremental progress with regular evaluation of results.

Instead of developing the complete system in a single phase, the project is divided into multiple sprints. Each sprint delivers a functional and testable component such as data preprocessing, feature integration, model training, or visualization.

---

2. Why Agile is Suitable for This Project

The project involves multiple interconnected components such as satellite data ingestion, auxiliary data integration, machine learning model development, evaluation, and visualization. Model performance and feature selection may evolve based on experimental results and data quality. Agile is suitable because it allows:

- Incremental development of data processing and ML components
- Early testing of downscaling models and evaluation metrics
- Continuous feedback based on model accuracy and outputs
- Easy refinement of features and model parameters
- Reduced risk through early validation of assumptions

---

3. Agile Workflow

The development of the AI-based air quality downscaling system follows an iterative Agile workflow. The project is divided into short development cycles called sprints, with each sprint delivering a working and testable part of the system.

1. Requirement Planning

Requirements are selected from the Software Requirements Specification (SRS), such as satellite data ingestion, preprocessing, feature engineering, model training, or visualization.

2. Design

The selected features are logically designed, including data flow, model architecture, and interaction between different components.

3. Development

The designed features are implemented using Python and relevant ML libraries. This includes coding data pipelines, training ML models, or generating output maps.

4. Testing

Each implemented feature is tested for correctness and performance. Model outputs are evaluated using suitable metrics to verify improvement in spatial resolution.

5. Review and Feedback

The completed sprint is reviewed, and feedback from mentors or evaluators is used to refine the model, improve preprocessing, or adjust features.

6. Next Sprint

Based on feedback and remaining requirements, the next set of features is planned and the Agile cycle continues.

---

4. Sprint Plan

Sprint 1 – Data Collection and Understanding

- Collection of low-resolution satellite air quality data
- Understanding data format and spatial resolution
- Initial data exploration and visualization

Sprint 2 – Data Preprocessing

- Handling missing and inconsistent values
- Normalization and scaling of features
- Alignment of satellite and auxiliary datasets

Sprint 3 – Feature Engineering

- Integration of meteorological and land-use features
- Creation of input features for the ML model
- Feature selection and analysis

Sprint 4 – Model Development

- Selection of baseline ML model for downscaling
- Training the model using prepared datasets
- Initial evaluation of model performance

Sprint 5 – Evaluation and Visualization

- Calculation of evaluation metrics
- Visualization of high-resolution air quality maps
- Comparison between low-resolution and enhanced outputs

Sprint 6 – Optimization and Final Testing

- Model tuning and performance improvement
- End-to-end testing of the complete system
- Documentation and result consolidation

---

5. Advantages of Using Agile

Agile enables the project to progress in a flexible and structured manner. Errors in data preprocessing or model performance can be identified early and corrected without reworking the entire system. Continuous testing and evaluation improve reliability and ensure that the project objectives are met effectively within academic constraints.

6. Conclusion

By following the Agile development model, the AI-based air quality downscaling system can be developed efficiently with continuous validation at each stage. Iterative development, regular feedback, and incremental delivery ensure that the final system is accurate, well-structured, and suitable for demonstrating the application of AI/ML techniques in satellite data analysis.