IntelliHack 5.0

Team: CodeLabs

Task 01- Report 02

Initial round

Table of Content

MLOps System Design for Weather Forecasting

1. Introduction

This report describes the MLOps system designed to predict the probability of rain for the next 21 days using historical weather data. The system is engineered to handle real-time data ingestion from IoT sensors and other data sources, process the data using robust ETL pipelines, and deliver predictions via a Model Inference API. The code provided below is part of the model inference component that computes future rain probabilities using a pre-trained model.

2. System Architecture Overview

- **Data Ingestion Layer:**
  Captures data from various sources—including IoT sensors, Excel files, and weather APIs—in real time using streaming technologies (e.g., Kafka or MQTT) with backup batch processing to ensure continuous data flow even during sensor outages.

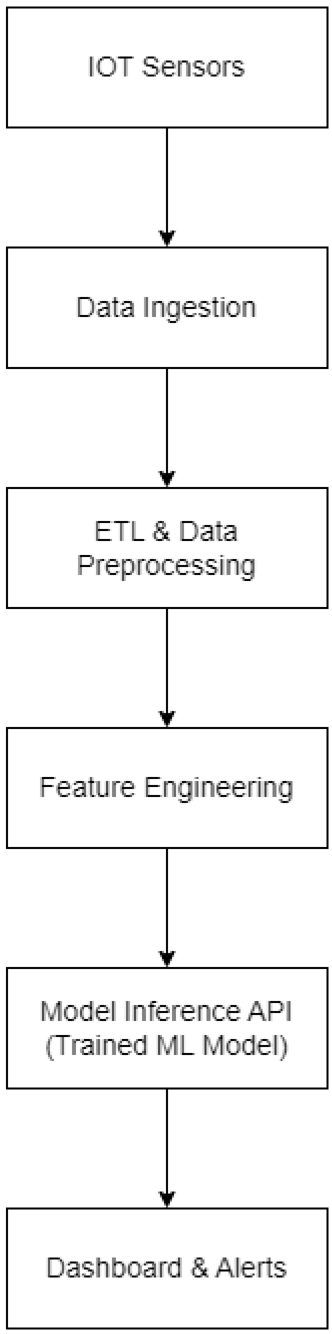
- **Data Preprocessing & Storage:**
  Cleans and transforms raw data through an ETL pipeline that handles missing values, formatting issues, and duplicates. The processed data, along with the raw input, is stored in a database or data lake for both immediate analysis and historical reference.

- **Feature Engineering Module:**
  Applies necessary transformations to create features (e.g., lagged variables, rolling averages, and interaction terms) that mirror the training process. This ensures consistency in data representation between training and real-time inference.

- **Model Inference Engine:**
  The model inference engine is responsible for loading the pre-trained model ***(rain_prediction_model.pkl)*** and generating future predictions. This is used to predict rain probability for the next 21 days. It leverages iterative forecasting by updating lagged and rolling features based on previous day outputs. This simulation-based approach reflects the dynamic nature of weather data and helps in providing daily updated probabilities.

- **Dashboard & Alerting System:**
  Provides an intuitive interface for visualizing historical trends and future predictions. It also includes an alert mechanism to notify users (such as farmers or project managers) when significant rain probabilities are detected.

- **Monitoring & Feedback:**
  Continuously checks the quality of incoming sensor data and tracks model performance. It facilitates automatic alerts for data anomalies or prediction drifts and triggers retraining processes when necessary.



3. Integration & Handling Sensor Malfunctions

Within the overall system, the above prediction module is integrated as follows;

- **Error Handling:** The model inference service is designed with error-handling routines to flag inconsistent or missing data from IoT sensors. If data ingestion fails, fallback mechanisms (e.g., batch mode predictions from stored data) are triggered.
- **Real-Time Updates:** The Model Inference API updates predictions on a daily basis. In a live system, this code would be encapsulated in a microservice that responds to real-time API calls.
- **Scalability:** Modular design allows for independent scaling of the data ingestion, processing, and prediction services to handle increased loads or additional sensor data.

4. Conclusion

This MLOps system design ensures that the weather forecasting model is integrated into a scalable, fault-tolerant pipeline. By combining robust data ingestion, dynamic feature engineering, and real-time model inference, the system delivers actionable 21-day rain probability predictions essential for smart agriculture operations.