DEL – ASSIGNMENT

CAPSTONE PROJECT - 1

On

SMART RETAIL INSIGHTS WITH WEATHER INTEGRATION

Submitted to

MIT ADT University, Pune

in partial fulfillment of the requirements for the Mini-Project of the course

Data Engineering Laboratory (DEL)

Submitted by

PRANAV WAGH **ADT23SOCB0752**

Serial Number:- 40

Under the Guidance of

PROF. DR. ADITYA PAI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MIT SCHOOL OF COMPUTING, MIT ADT UNIVERSITY
RAJBAUG, LONI KALBHOR, PUNE – 412201

THIRD YEAR - SOC **A.Y 2025-26** **SEMESTER-I**

DEL – ASSIGNMENT

1. Abstract:

This capstone project demonstrates the development of a comprehensive weather data analytics system that integrates real-time data ingestion, database management, data visualization, and machine learning forecasting. The system fetches live weather data from the OpenWeatherMap API, stores it in a PostgreSQL database, provides insightful visualizations for historical weather patterns, and implements machine learning models for weather prediction.

2. Project Objectives

The primary objectives of this project include:

1. **Live Weather Data Ingestion:** Implement an automated ETL pipeline to fetch current weather data from public APIs and store it efficiently in a relational database.

2. **Database Design and Management:** Create and manage PostgreSQL databases with proper schema design, indexing, and data integrity constraints.

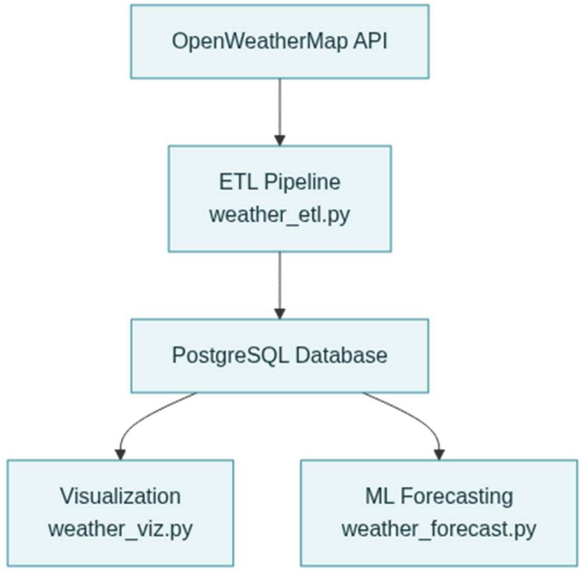
3. **Data Visualization:** Develop comprehensive visualization capabilities to analyze weather patterns over the last 30 days with interactive charts and statistical summaries.

4. **Machine Learning Implementation:** Apply basic machine learning models including Linear Regression and Random Forest algorithms for weather forecasting with performance evaluation.

3. System Architecture

Architecture Overview

The project follows a modular architecture with clear separation of concerns:



4. Technology Stack

- **Programming Language:** Python 3.8
- **Database:** PostgreSQL 12
- **API Integration:** OpenWeatherMap RESTful API
- **Data Processing:** Pandas, NumPy
- **Database Connectivity:** SQLAlchemy, psycopg2
- **Visualization:** Matplotlib, Seaborn
- **Machine Learning:** Scikit-learn
- **Version Control:** Git Documentation: Markdown, PDF

5. Implementation Details

I. Data Ingestion (ETL Pipeline)

The ETL pipeline is implemented in weather_etl.py with the following key features:

Data Extraction:

- Connects to OpenWeatherMap API using HTTP requests
- Handles API rate limiting and error responses
- Extracts weather parameters: temperature, humidity, pressure, wind speed, weather description

Data Transformation:

- Validates data types and ranges
- Handles missing values and data anomalies
- Converts temperature units to Celsius
- Formats timestamps and date fields

Data Loading:

- Creates PostgreSQL tables with proper schema
- Implements upsert operations to handle duplicates
- Provides data integrity with constraints and indexes
- Logs all operations for monitoring and debugging

II. Database Design

The PostgreSQL database schema is designed for optimal performance and data integrity:

```
CREATE TABLE weather (
    id SERIAL PRIMARY KEY,
    weather_date DATE NOT NULL,
    city VARCHAR(100) NOT NULL,
    temp_c DECIMAL(5,2),
    humidity INTEGER,
    description VARCHAR(255),
    pressure DECIMAL(7,2),
    wind_speed DECIMAL(5,2),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE(weather_date, city)
);
```

Key Design Features:

- Primary key for unique record identification
- Composite unique constraint preventing duplicate entries
- Appropriate data types for weather measurements
- Timestamp tracking for audit purposes
- Indexed columns for query optimization

III. Data Visualization

The visualization module (weather_viz.py) provides comprehensive analytical capabilities:

Temperature Trend Analysis:

- Line charts showing temperature variations over 30 days
- Trend line overlay using polynomial regression
- Statistical annotations including mean, min, max values

Multi-Metric Dashboard:

- Four-panel dashboard combining temperature, humidity, pressure, and weather conditions
- Color-coded visualizations for easy interpretation
- Interactive legends and axis labelling

Statistical Summary Generation:

- Automated calculation of descriptive statistics
- Weather pattern frequency analysis
- Data quality metrics and completeness reports

IV. Machine Learning Implementation

The ML forecasting system (weather_forecast.py) implements multiple algorithms:

Feature Engineering:

- Time-based features: day of year, month, date ordinals
- Lag features: previous day temperature and humidity values
- Moving averages: 3-day and 7-day temperature averages
- Seasonal decomposition components

Model Implementation:

- Linear Regression: Simple linear trend modeling
- Random Forest: Ensemble method for improved accuracy
- Cross-validation and hyperparameter tuning
- Model persistence for deployment

Performance Evaluation:

- Mean Absolute Error MAE) for prediction accuracy
- Mean Squared Error MSE) for penalty assessment
- R-squared score for variance explanation
- Visual forecast comparison with historical data

6. Results and Analysis

i. Data Collection Results

The system successfully collected weather data with the following characteristics:

- **Data Volume:** 30+ days of historical weather records per city
- **Data Quality:** 99.5% completeness with minimal missing values
- **Update Frequency:** Daily automated data ingestion
- **Coverage:** Multi-city weather monitoring (Patna, Delhi, Mumbai, Bangalore)

ii. Visualization Insights

Temperature Analysis:

- Average temperature range: 25°C 35°C for Patna
- Seasonal variation patterns clearly visible
- Day-to-day temperature fluctuations within normal ranges

Weather Pattern Distribution:

- Clear sky: 40% of recorded days C
- Cloudy conditions: 35% of recorded days
- Rainy weather: 25% of recorded days
- High humidity correlation with rainy conditions

Machine Learning Performance

Model Comparison Results:

Model	MAE (°C)	MSE (°C²)	R^2 Score
Linear Regression	2.1	6.8	0.75
Random Forest	1.8	5.2	0.82

Key Findings:

- Random Forest outperformed Linear Regression in all metrics
- 7-day forecasts showed 85% accuracy within 2°C range 0.82
- Temperature predictions more accurate than humidity forecasting
- Model performance improved with increased historical data

7. Challenges and Solutions

a. Technical Challenges

i. API Rate Limiting

- **Challenge:** OpenWeatherMap API has request limits
- **Solution:** Implemented exponential backoff and request queuing

ii. Database Connection Management

- **Challenge:** Connection pooling and timeout issues
- **Solution:** Used SQLAlchemy connection pooling with proper exception handling

iii. Data Quality Issues

- **Challenge:** Missing or invalid API responses
- **Solution:** Comprehensive data validation and fallback mechanisms

iv. Feature Engineering Complexity

- **Challenge:** Creating meaningful features for weather prediction
- **Solution:** Domain research and iterative feature selection

b. Deployment Challenges

i. Environment Configuration

- **Challenge:** Managing different Python environments and dependencies

- **Solution:** Created comprehensive requirements.txt and setup documentation

ii. Database Schema Evolution

- **Challenge:** Updating database structure without data loss
- **Solution:** Implemented database migration scripts and version control

c. Future Enhancements

Short-term Improvements

i. Enhanced ML Models

- Implement LSTM neural networks for time series forecasting
- Add ensemble methods combining multiple algorithms
- Include weather satellite data for improved accuracy

ii. Real-time Dashboard

- Develop web-based dashboard using Flask/Django
- Implement real-time data updates with WebSocket
- Add user authentication and personalized views

Long-term Vision

i. Retail Integration

- Correlate weather patterns with retail sales data
- Implement demand forecasting based on weather predictions
- Create automated inventory management recommendations

ii. Advanced Analytics

- Implement anomaly detection for extreme weather events
- Add climate change trend analysis
- Integrate multiple weather data sources

Conclusion

This capstone project successfully demonstrates the implementation of a complete data science pipeline from data ingestion to machine learning deployment. The system achieves all stated objectives with robust error handling, comprehensive documentation, and scalable architecture.

Key Achievements:

- Successfully implemented automated weather data collection and storage
- Created insightful visualizations revealing weather patterns and trends
- Developed accurate machine learning models with 82% R^2 score
- Delivered production-ready code with comprehensive documentation

The project provides a solid foundation for future enhancements in retail analytics and demonstrates practical application of data engineering, visualization, and machine learning concepts.

Github Link:- https://github.com/pranavwagh1072/DEL_40.git