RV UNIVERSITY

School of Computer Science and Engineering

Bengaluru – 560059

FUNDAMENTALS OF DATA ENGINEERING

COURSE CODE: CS3238

V Semester B.Sc. (HONS.)
Project Report

Name	**Niyanthri R Sridhar** **Mohammed Ikram**
USN	**1RVU22BSC065** **1RVU22BSC054**
Academic Year	**2024 - 2025**
Project Title	**Weather Data Collection Data Pipeline**

Weather Data Collection Data Pipeline Project Report

---

1. Abstract

This project involves the creation of a scalable data pipeline to collect, process, and store weather data from a public API, aiming to provide real-time weather insights and historical data for analysis. Leveraging Python, Databricks, and Delta Lake, the pipeline fetches data at regular intervals, cleans and structures it, and stores it in a format optimized for analysis and visualization. By implementing data visualization and real-time alerting mechanisms, the pipeline serves as a powerful tool for monitoring weather patterns and generating alerts for adverse weather conditions. This project has broad applications across industries such as agriculture, transportation, and urban planning, where timely weather information is crucial.

2. Introduction

Weather data plays an essential role in decision-making across a wide range of sectors, including agriculture, transportation, and urban planning. For instance, in agriculture, real-time data helps farmers plan optimal planting and harvesting schedules based on forecasted weather, while transportation sectors use weather alerts to improve route planning and safety protocols. Real-time weather data enables users to make timely, informed decisions, while historical data provides a foundation for trend analysis and predictive modeling. This project focuses on building a data pipeline to automate the collection, transformation, and storage of weather data, ensuring it is readily accessible for analysis and visualization. Through this pipeline, we aim to create a reliable resource that can assist city managers, environmental analysts, and disaster response teams in making data-driven decisions based on accurate and timely weather information.

3. Project Profile

a. Objectives

The main objectives of this project are:

- To create an automated pipeline that fetches and stores real-time weather data from a public API.
- To transform, clean, and validate the data for easy retrieval and analysis, ensuring high data quality and reliability.
- To design visualizations and analytical tools for tracking weather patterns.
- To set up real-time alerts for specific weather conditions, aiding rapid response to extreme events.

b. Dataset

The dataset for this project comes from an online weather API (such as OpenWeatherMap), known for its reliability and frequent data refresh rate. Data is collected at regular intervals and includes various weather parameters, such as:

- City name
- Temperature
- Weather description (e.g., clear, cloudy, rainy)
- Timestamp This data is stored in Delta Lake tables on Databricks, supporting efficient querying and schema evolution for incremental data updates.

c. Methodology

The pipeline is built using Python within the Databricks environment and involves the following steps:

1. **Data Collection**: The pipeline uses the requests library to fetch weather data from the API at regular intervals, managed through the schedule library. The choice of schedule was based on its simplicity and flexibility for setting up recurring data fetches with minimal configuration.
2. **Data Transformation**: Collected data is initially structured into a Pandas DataFrame and later transformed into a Spark DataFrame for efficient storage and querying. Data cleaning processes include removing unnecessary fields, filling in missing values where applicable, and enforcing a uniform schema to ensure consistency.
3. **Error Handling and Monitoring**: To handle API errors or network issues, the pipeline includes retry mechanisms, with alerts triggered if data collection is interrupted. Logs are generated for each step, allowing for easy debugging and monitoring of pipeline health.
4. **Data Storage**: The transformed data is written into a Delta Lake table on Databricks, which enables structured querying and supports efficient analysis.
5. **Data Visualization and Analysis**: A series of SQL queries and Databricks dashboards are used to visualize weather patterns, such as displaying the hottest and coldest cities in real time. The dashboards are designed to be intuitive and customizable, facilitating rapid analysis and decision-making.

4. Observations and Analysis

Significant weather patterns were identified from the collected data:

- **Temperature Variations**: Some cities displayed substantial daily temperature fluctuations, indicating potential regions with high weather volatility, which can impact agriculture and energy sectors.
- **Alerts for Extreme Conditions**: Real-time alerts were generated for temperature extremes, highlighting regions that may experience heatwaves or severe storms, supporting proactive emergency planning.
- **Heatmap**: A heatmap visualization was created to depict temperature variations across different cities, clearly highlighting regions experiencing extreme weather

conditions. This provides a quick overview for stakeholders like disaster management teams.

- **Top/Bottom Temperature**: SQL queries were run to identify the cities with the highest and lowest recorded temperatures, aiding in tracking temperature extremes across regions.
- **Real-Time Counters**: Real-time counters track the number of updates per city, providing a detailed view of the latest weather parameters for quick analysis.

5. Results

The data pipeline was successfully deployed and automated to collect weather data at regular intervals. Key findings include:

- Real-time weather alerts were triggered for certain threshold values, such as extreme temperatures.
- The pipeline reliably stored all weather data, enabling analysis of trends across different time periods.
- Visualization on Databricks allowed for easy interpretation of data, highlighting cities with specific weather trends (e.g., hottest, coldest, most humid).

The successful deployment and functionality of this pipeline underscore its potential as a reliable resource for real-time weather monitoring.

6. Conclusion

The Weather Data Collection Data Pipeline project effectively demonstrates the benefits of automating data ingestion, storage, and monitoring to deliver real-time weather insights. By leveraging Databricks, Delta Lake, and Python, this pipeline is highly scalable and adaptable to various weather APIs and data structures. Not only does the pipeline automate weather data collection, but it also provides robust tools for meaningful analysis and visualization, enabling data-driven decision-making across sectors like agriculture, urban planning, and disaster management. Future improvements could involve integrating other environmental data sources, enhancing alert mechanisms with anomaly detection, and implementing predictive modeling for more advanced weather analytics. Such additions would broaden the pipeline's applicability, further positioning it as a valuable resource for proactive weather management and planning.