TITLE: REAL-TIME WEATHER DATA PROCESSING PIPELINE

Presented by-
Byreddy Sireesha
Patrika Chatterjee
Sudhir Kumar Singh
Seshanth G
Maridu Sruthi

CONTENTS

- PROJECT OVERVIEW
- OBJECTIVE
- TECHNICAL REQUIREMENTS
- PROJECT ARCHITECTURE
- HIGH LEVEL DESIGN
- LOW LEVEL DESIGN
- ERROR HANDLING & DATA QUALITY
- AUDITING, ALERTS
- WORKFLOW ORCHESTRATION
- TESTING
- BUSINESS OUTCOMES

PROJECT OVERVIEW AND OBJECTIVE

PROJECT OVERVIEW

- Real-time weather data ingestion & processing using AWS Kinesis + Databricks (PySpark)
- Delta Lake (Bronze → Silver → Gold) with Unity Catalog for governance & analytics
- Enhanced with alerts, anomaly detection & Git-based CI/CD

.

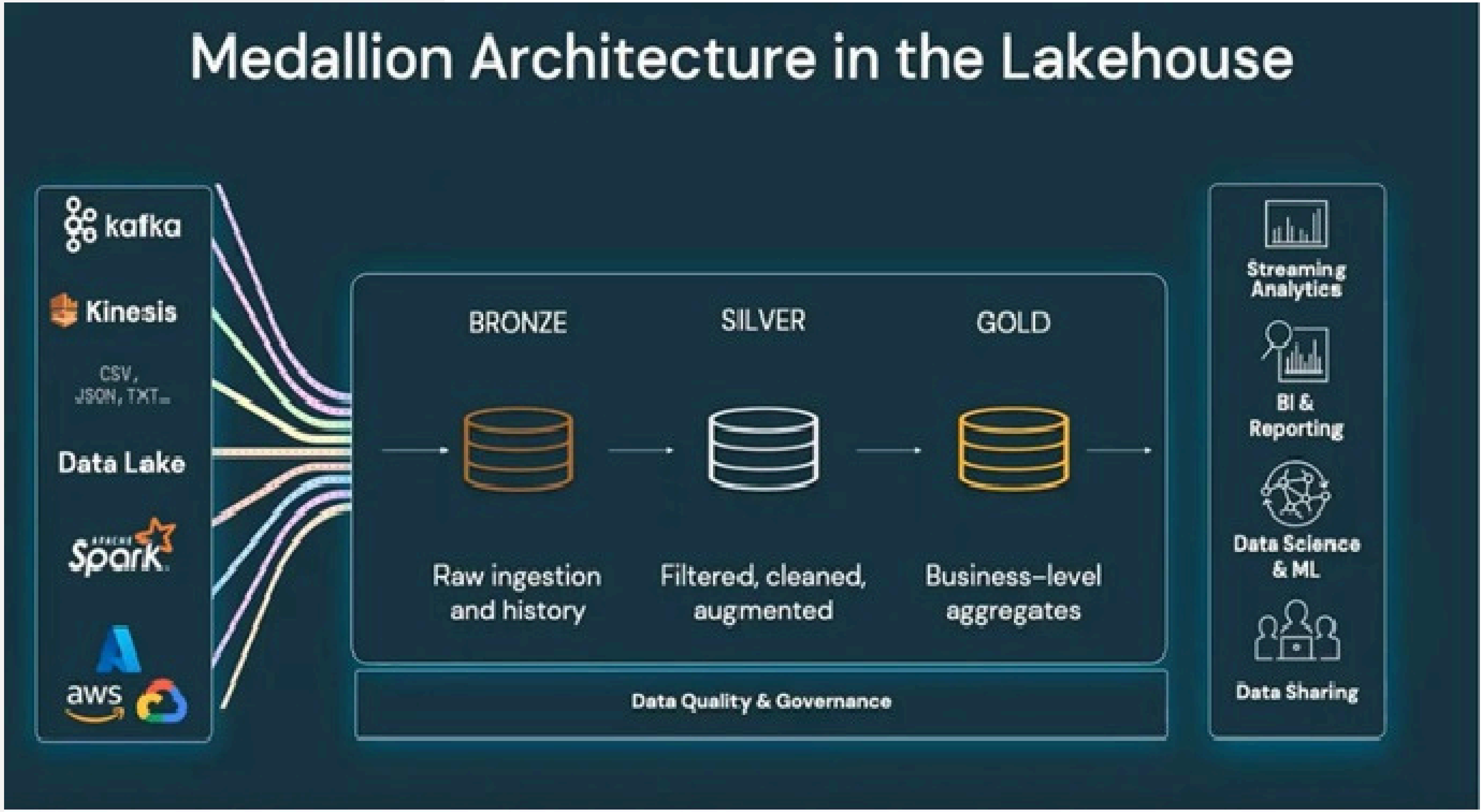
OBJECTIVES

- Real-time forecasting with accurate & validated data
- Actionable insights: stats, anomalies, extreme weather detection
- Reliable pipeline with governance, monitoring & orchestration

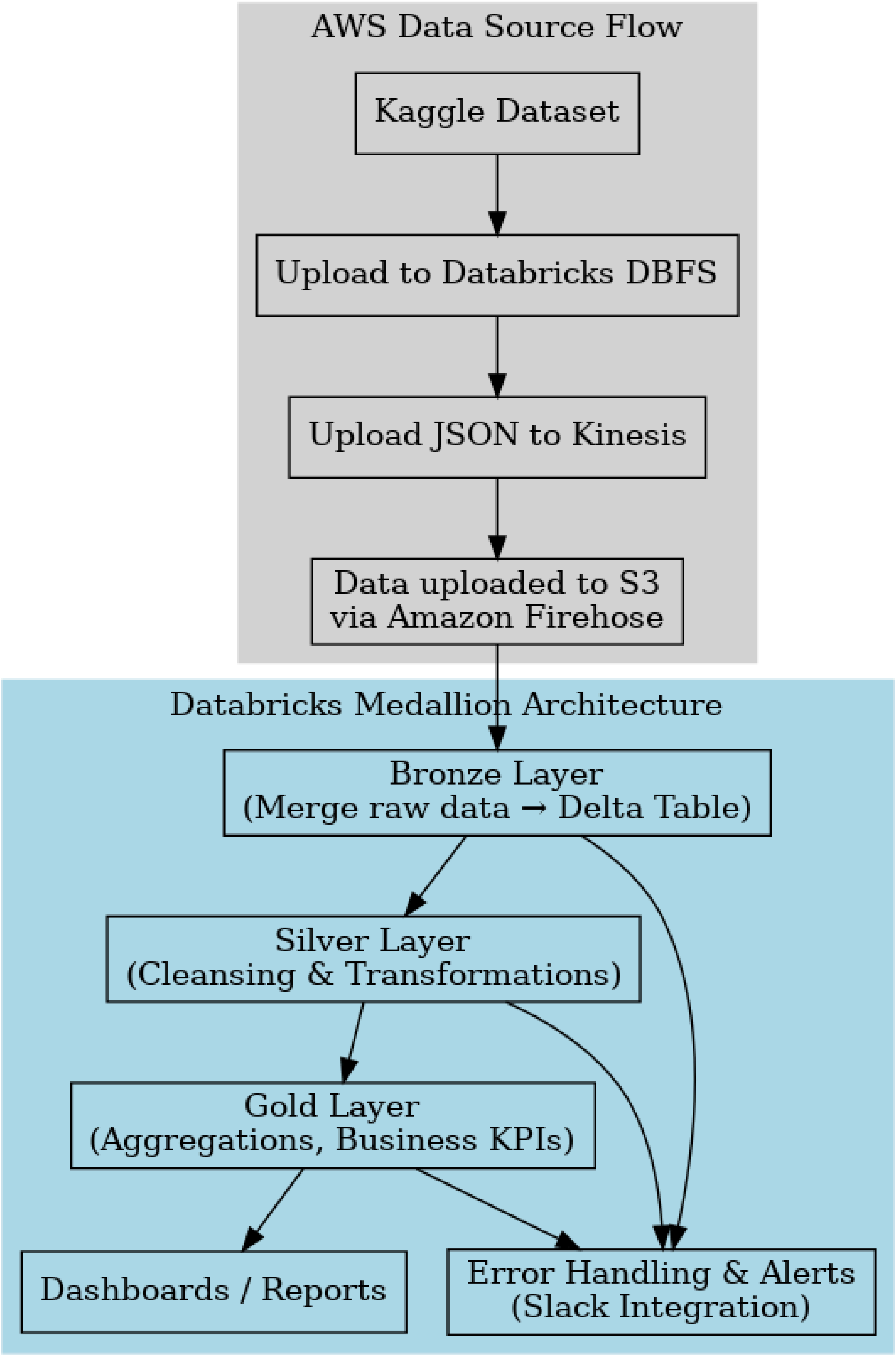
TECHNICAL REQUIREMENTS & SPECIFICATIONS

- **Ingestion:** Real-time JSON weather data via Amazon Kinesis
- **Processing & Storage:** Databricks + PySpark; Delta Lake (Bronze → Silver → Gold)
- **Error Handling & Monitoring:** Databricks logs, retries & checkpoints
- **Data Quality & Governance:** Schema validation, null/outlier checks, Unity Catalog
- **Alerts & Collaboration**: Slack notifications; GitHub
- **Orchestration & Performance:** Databricks Workflows, autoscaling clusters, batch & streaming support

PROJECT ARCHITECTURE

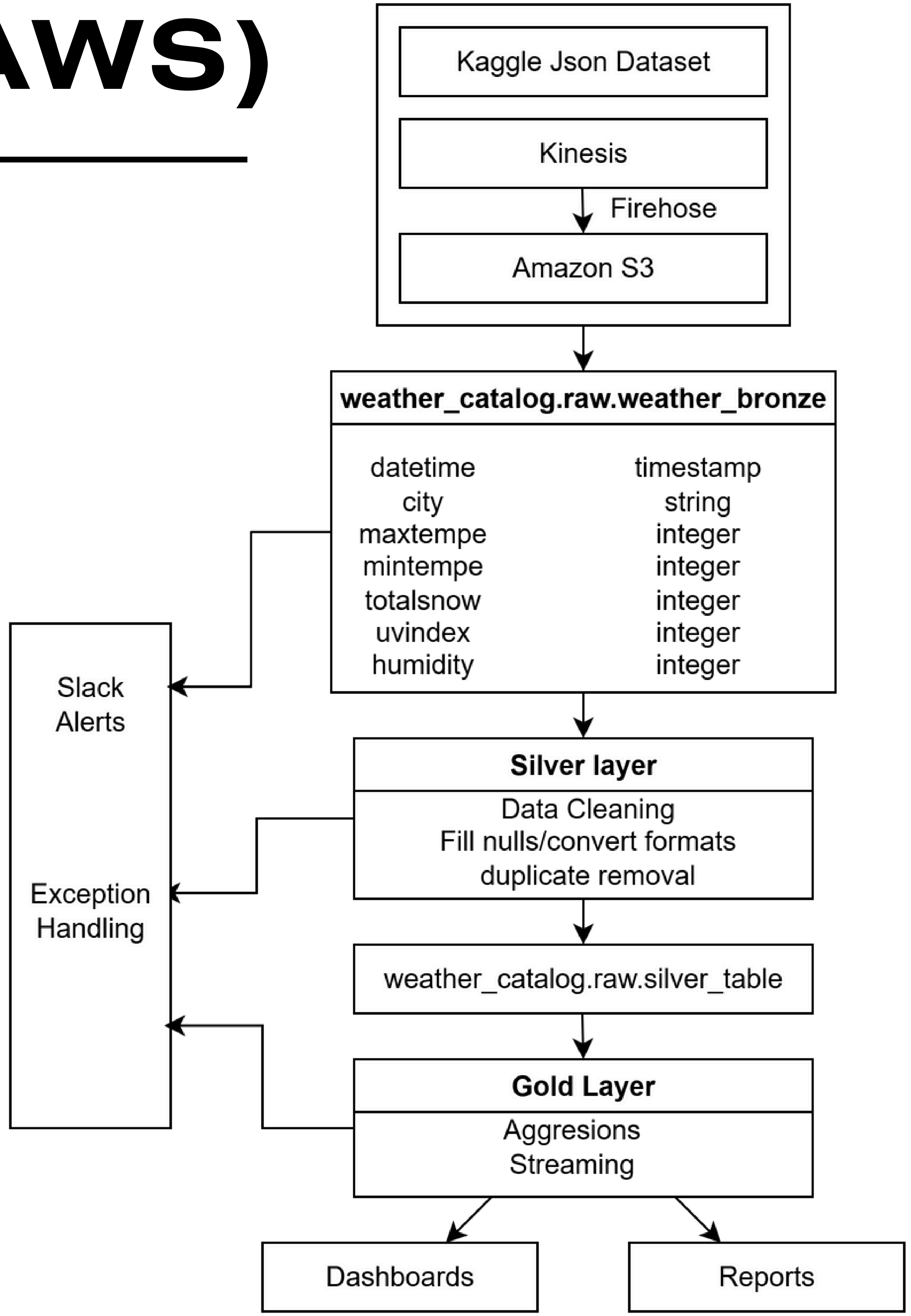

Medallion Architecture in the Lakehouse
kafka
Kinesis
CSV,
JSON, TXT...
Data Lake
Spark
aws
BRONZE
SILVER
GOLD
Raw ingestion and history
Filtered, cleaned, augmented
Business-level aggregates
Data Quality & Governance
Streaming Analytics
BI & Reporting
Data Science & ML
Data Sharing

HIGH LEVEL DESIGN



LOW-LEVEL DESIGN (DATABRICKS + AWS)

- **Raw Table Creation** – created weather_catalog.raw.weather_bronze with schema (datetime, city, maxtemp, mintemp, humidity, uvindex, totalsnow).
- **Data Cleaning Rules** – We applied null handling, type casting, and duplicate removal to generate weather_catalog.raw.silver_table.
- **Error Handling & Alerts** – We implemented try–except blocks and integrated Slack alerts for pipeline monitoring.
- **Analytics Tables** – We developed weekly, monthly, yearly, and city-wise analytics tables in the Gold layer for reporting and dashboards.



ERROR HANDLING APPROACHES

- **Streaming Error Logging:** Capture errors during JSON parsing & ingestion
- **Slack Alerts**: Notify team of anomalies, job failures, or delays
- **Checkpointing & Retry:** Enable structured streaming checkpoints to ensure fault tolerance
- **Dead Letter Queue (DLQ)**: Invalid/corrupt records stored separately for review

DATA QUALITY CHECKS

Schema Validation: Ensure all required fields exist (timestamp,city, temperature)

Null/Invalid Checks: Filter out missing/negative values (e.g., negative humidity)

Range Checks: Flag abnormal values (temperature < -10°C, wind speed > 200 km/h)

Deduplication: Handle duplicate records

Business Rules: Compute rolling averages, detect sudden spikes

AUDITING

- **Audit Table:** weather_catalog.logging.ingestion_silver
- **Tracks:** timestamp, city, temperature, pressure, humidity
- **Run Status Email Notification:** Capture pipeline run IDs, execution time, job status (success/failure); sends mails to team members
- **Integration:** Git integrated with Databricks for version control and collaboration
- **Data Lineage:** Unity Catalog tracks transformations across Bronze → Silver → Gold

ALERTS & NOTIFICATIONS

Types of Alerts Sent via Slack:-

- **Pipeline Success:** Ingestion completed for Bronze layer
- **Data Quality Warning:** 10% records dropped due to missing humidity
- **Failure/Error:** Kinesis stream disconnected, pipeline stopped

WORKFLOW ORCHESTRATION

- **Databricks Workflows:**

Orchestrates ingestion → transformation → aggregation → alerts

Schedules streaming jobs

- **Git integration with Databricks**:

Automates deployment of Databricks notebooks.

- **Monitoring:**

Databricks job logs

TESTING STRATEGY

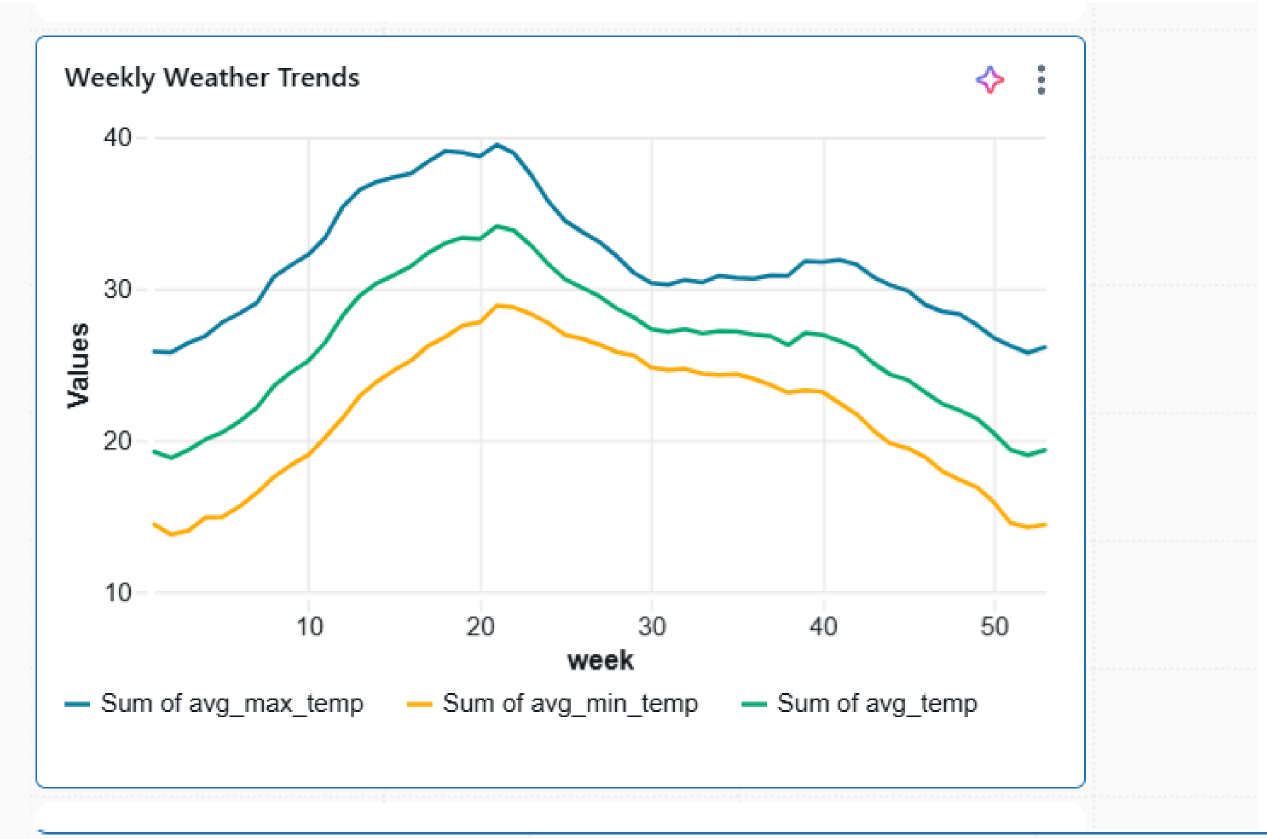
Types of Testing Performed

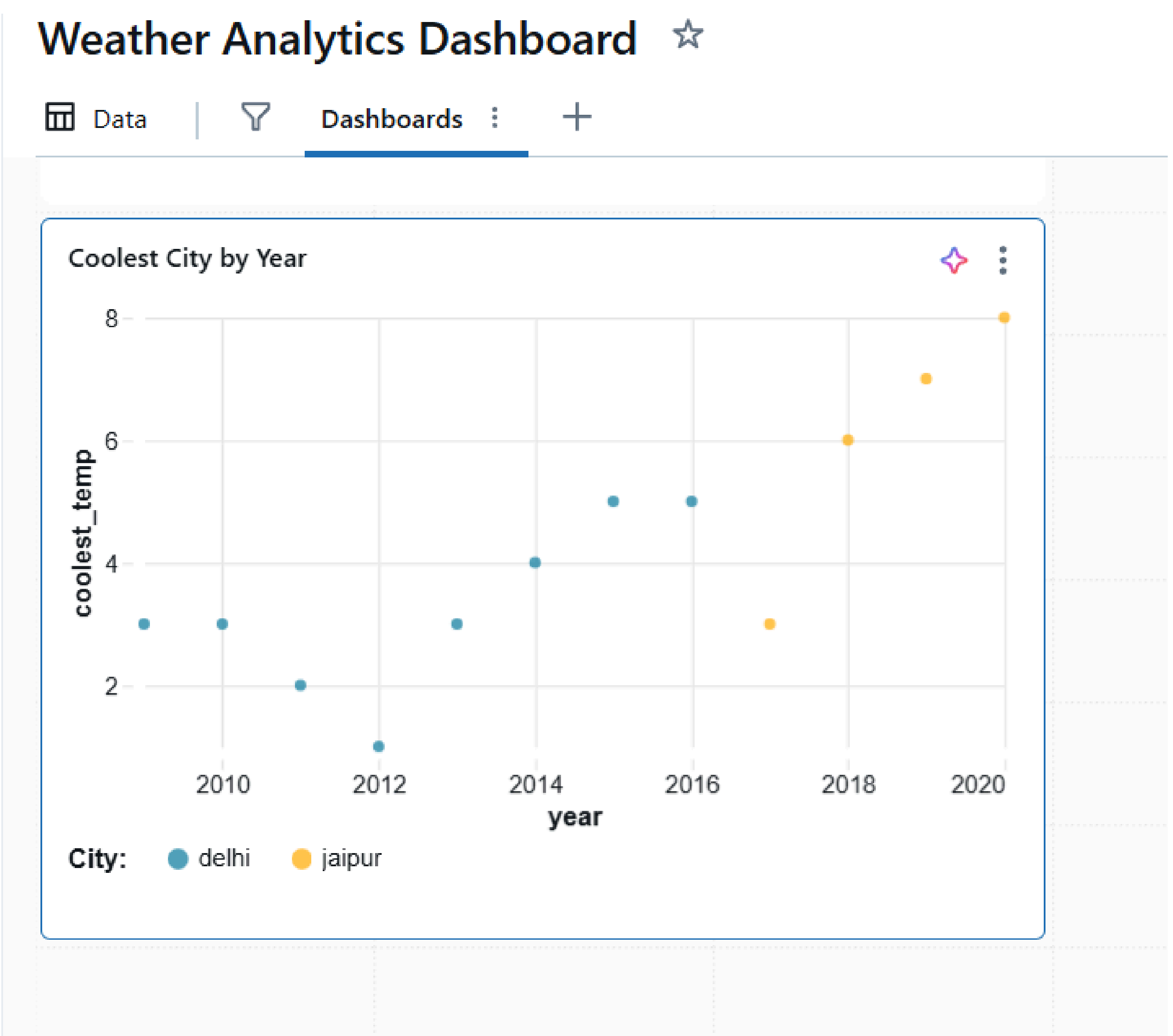
- **Unit Testing:** Check schema, nulls, and transformations in each layer.
- **Data Validation Testing:** Verify counts, value ranges, and business rules.
- **End-to-End Pipeline Checks**: Ensure Bronze → Silver → Gold workflow works correctly.
- **QA Sign-Off:** Review Gold tables and send Slack alerts for validation.

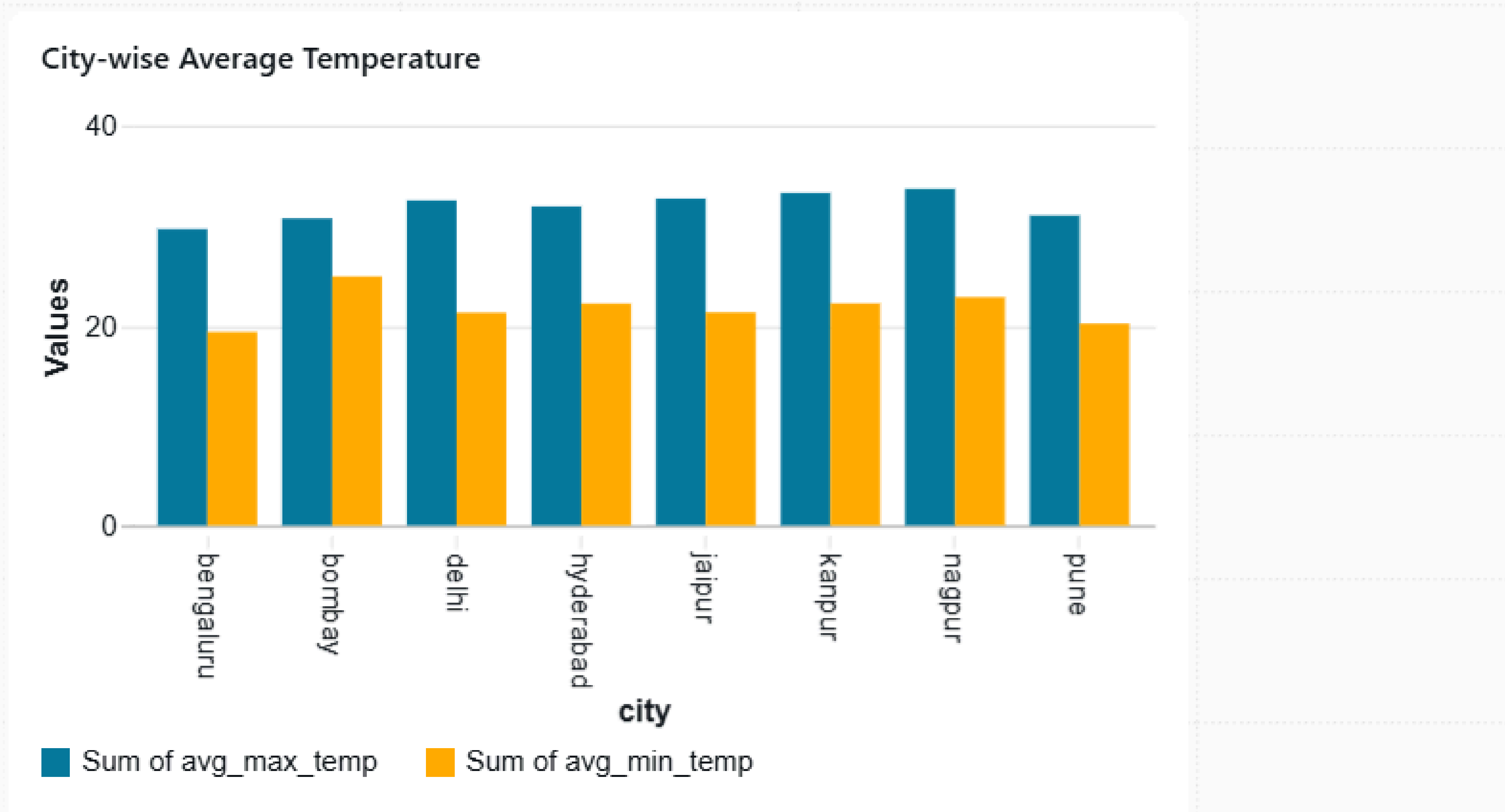
OUTCOMES & BUSINESS IMPACT

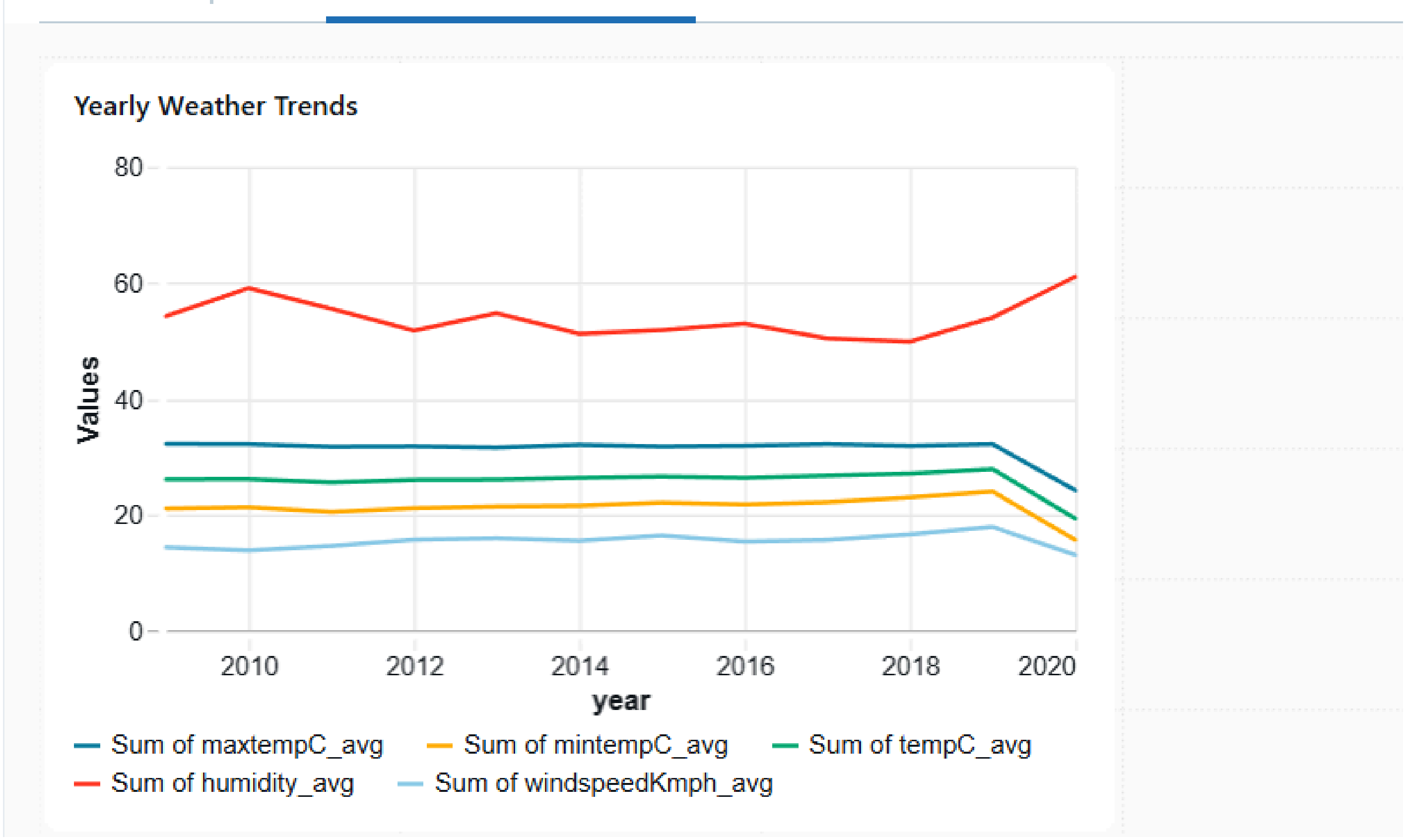
- Scalable Data Pipeline for real-time weather data
- Improved Data Quality through validations & logging
- Alerts for Extreme Weather → Faster disaster response
- Automated Orchestration & Monitoring with Databricks + AWS
- Seamless Version Control with Git integration

VISUAL OUTCOMES









THANK YOU