# Building a Real-Time Weather Insights Platform with GCP

MGMT–467 Data Ninjas 9 Capstone Project

# Our Problem

- Weather impacts energy demand, transportation, and short-term decision-making
- Existing tools separate historical and live data
- No easy way to track real-time temperature instability

**How can we build a system that blends historical and live weather data into one automated platform that highlights real-time instability?**

# Our Data

## Batch Data

Kaggle (Historical Weather 2012–2017)

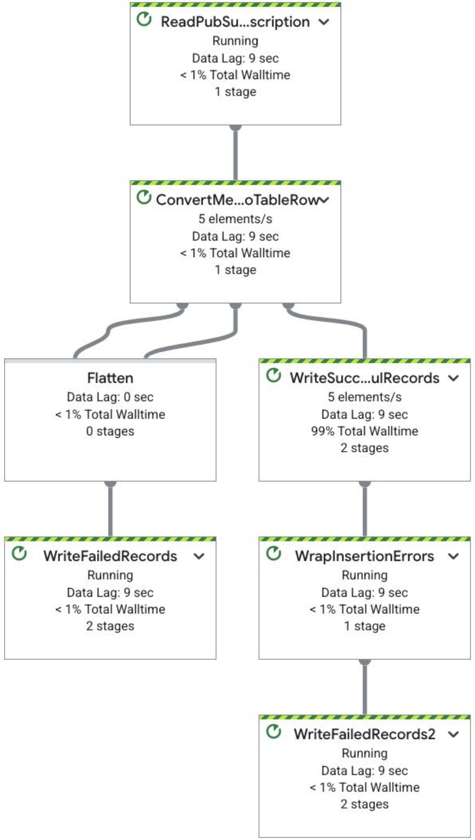
- 5 years of hourly weather for 36 global cities
- Used for baseline comparisons & feature engineering
- Loaded into BigQuery as weather_data_static

## Streaming Data

Open-Meteo API (Real-Time Weather)

- Updated live
- Includes temperature, humidity, pressure, wind metrics
- Pushed into Pub/Sub → Dataflow → BigQuery (weather_proj.live_weather)

# Data Preparation



ReadPubSu...scription
Running
Data Lag: 9 sec
< 1% Total Walltime
1 stage

ConvertMe...oTableRow
5 elements/s
Data Lag: 9 sec
< 1% Total Walltime
1 stage

Flatten
Data Lag: 0 sec
< 1% Total Walltime
0 stages

WriteSucc...ulRecords
5 elements/s
Data Lag: 9 sec
99% Total Walltime
2 stages

WriteFailedRecords
Running
Data Lag: 9 sec
< 1% Total Walltime
2 stages

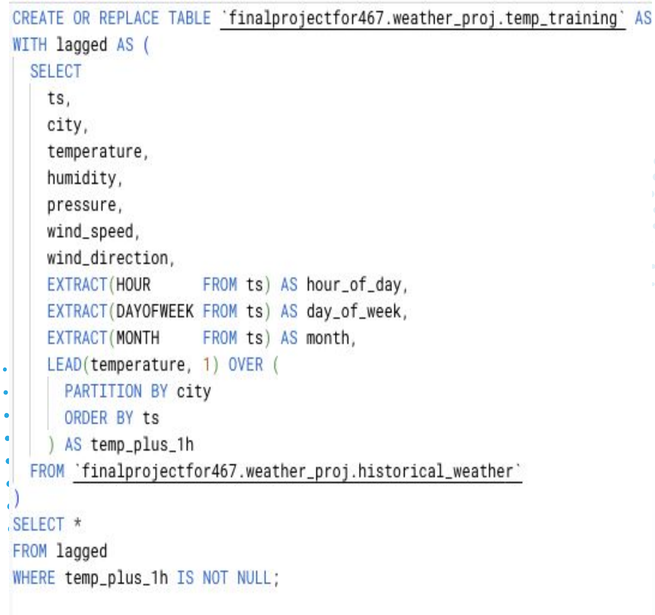
WrapInsertionErrors
Running
Data Lag: 9 sec
< 1% Total Walltime
1 stage

WriteFailedRecords2
Running
Data Lag: 9 sec
< 1% Total Walltime
2 stages

| Batch Data (Kaggle — Historical Weather 2012–2017) | Streaming Data (Open-Meteo API — Real-Time Weather) |
|---|---|
| Cleaned missing values and removed corrupted rows<br>Standardized units and timestamps<br>Engineered features (hourly averages, baseline variability, etc.)<br>Loaded curated tables into BigQuery (weather_data_static) | Cloud Function fetched live Normalized JSON response (city, timestamp, temperature, wind, pressure, humidity)<br>Published messages to Pub/Sub<br>**Dataflow Streaming Job** (graph shown):<br>● Flattened nested JSON<br>● Converted to BigQuery TableRow<br>● Wrote both successful and failed records to BigQuery<br>● Final output stored in weather_proj.live_weather and live_weather_with_delta<br>● Created additional features such as temp_delta_1h and hourly rolling averages |

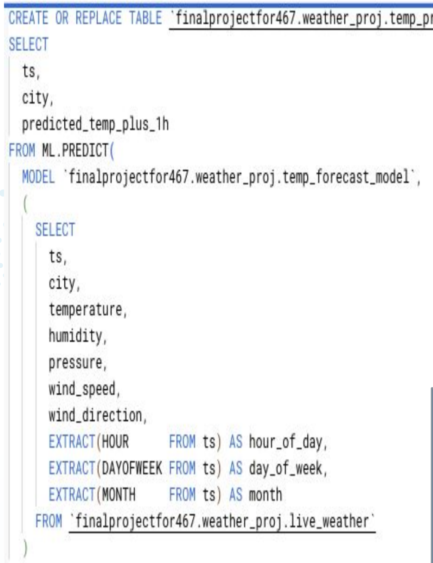
# Machine Learning Pipeline

## Trained on Historical (Kaggle) Data



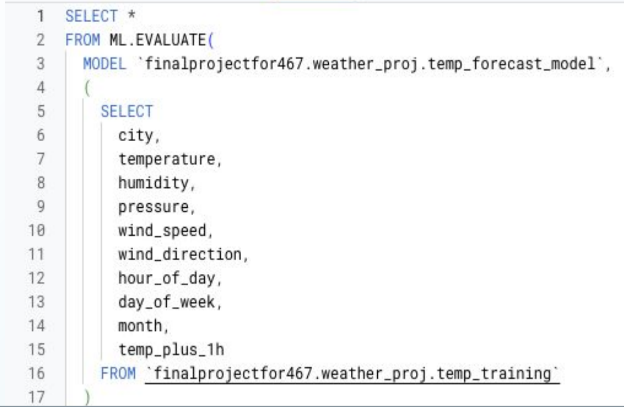
```
CREATE OR REPLACE TABLE `finalprojectfor467.weather_proj.temp_training` AS
WITH lagged AS (
  SELECT
    ts,
    city,
    temperature,
    humidity,
    pressure,
    wind_speed,
    wind_direction,
    EXTRACT(HOUR      FROM ts) AS hour_of_day,
    EXTRACT(DAYOFWEEK FROM ts) AS day_of_week,
    EXTRACT(MONTH     FROM ts) AS month,
    LEAD(temperature, 1) OVER (
      PARTITION BY city
      ORDER BY ts
    ) AS temp_plus_1h
  FROM `finalprojectfor467.weather_proj.historical_weather`
)
SELECT *
FROM lagged
WHERE temp_plus_1h IS NOT NULL;
```

## Tested On Streaming Data

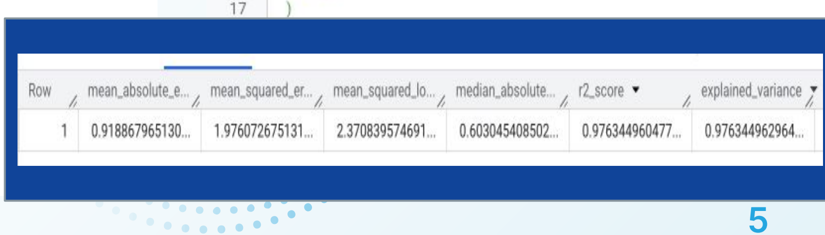


```
CREATE OR REPLACE TABLE `finalprojectfor467.weather_proj.temp_predictions_live` AS
SELECT
  ts,
  city,
  predicted_temp_plus_1h
FROM ML.PREDICT(
  MODEL `finalprojectfor467.weather_proj.temp_forecast_model`,
  (
    SELECT
      ts,
      city,
      temperature,
      humidity,
      pressure,
      wind_speed,
      wind_direction,
      EXTRACT(HOUR      FROM ts) AS hour_of_day,
      EXTRACT(DAYOFWEEK FROM ts) AS day_of_week,
      EXTRACT(MONTH     FROM ts) AS month
    FROM `finalprojectfor467.weather_proj.live_weather`
  )
```

## Evaluated



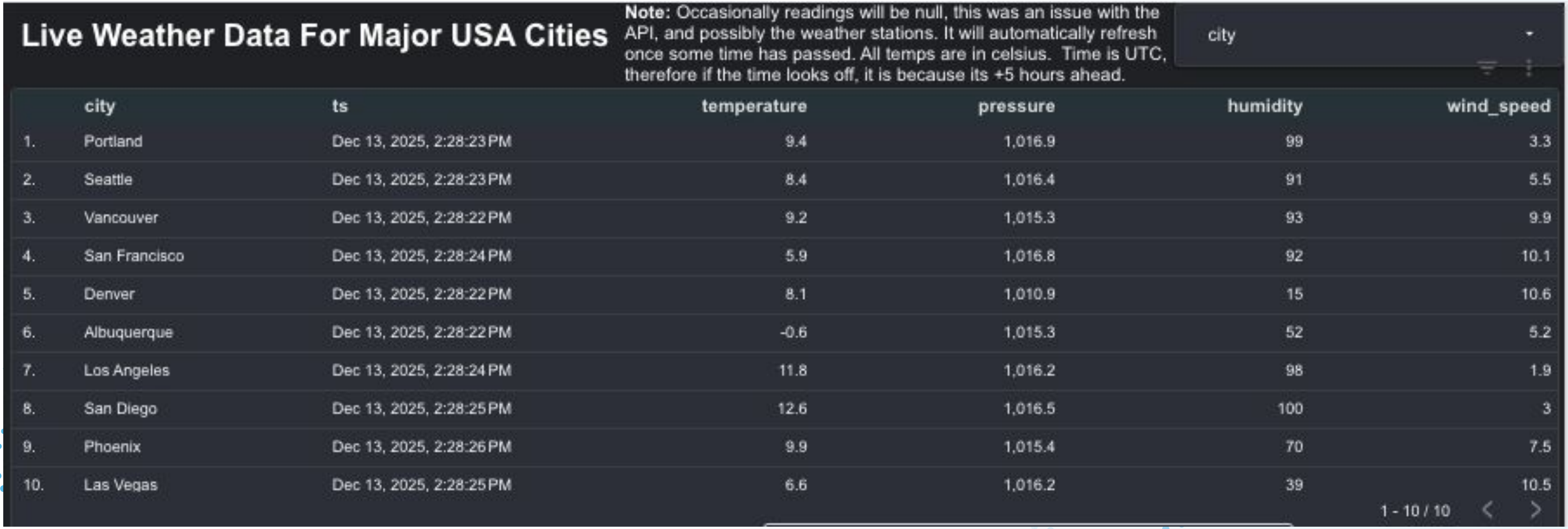
```
SELECT *
FROM ML.EVALUATE(
  MODEL `finalprojectfor467.weather_proj.temp_forecast_model`,
  (
    SELECT
      city,
      temperature,
      humidity,
      pressure,
      wind_speed,
      wind_direction,
      hour_of_day,
      day_of_week,
      month,
      temp_plus_1h
    FROM `finalprojectfor467.weather_proj.temp_training`
  )
```



| Row | mean_absolute_e... | mean_squared_er... | mean_squared_lo... | median_absolute... | r2_score ▾ | explained_variance ▾ |
|---|---|---|---|---|---|---|
| 1 | 0.918867965130... | 1.976072675131... | 2.370839574691... | 0.603045408502... | 0.976344960477... | 0.976344962964... |

# Looker Dashboard

## Live Weather Data For Major US Cities



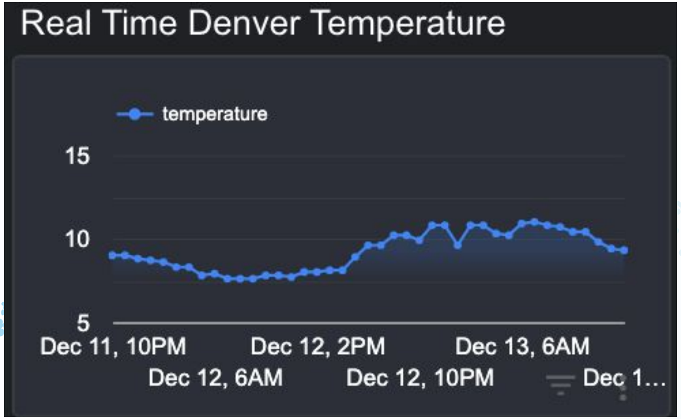
Live Weather Data For Major USA Cities

**Note:** Occasionally readings will be null, this was an issue with the API, and possibly the weather stations. It will automatically refresh once some time has passed. All temps are in celsius. Time is UTC, therefore if the time looks off, it is because its +5 hours ahead.

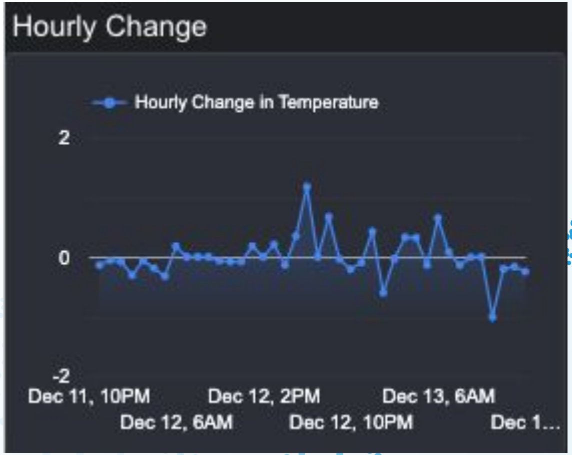
| | city | ts | temperature | pressure | humidity | wind_speed |
|---|---|---|---|---|---|---|
| 1. | Portland | Dec 13, 2025, 2:28:23 PM | 9.4 | 1,016.9 | 99 | 3.3 |
| 2. | Seattle | Dec 13, 2025, 2:28:23 PM | 8.4 | 1,016.4 | 91 | 5.5 |
| 3. | Vancouver | Dec 13, 2025, 2:28:22 PM | 9.2 | 1,015.3 | 93 | 9.9 |
| 4. | San Francisco | Dec 13, 2025, 2:28:24 PM | 5.9 | 1,016.8 | 92 | 10.1 |
| 5. | Denver | Dec 13, 2025, 2:28:22 PM | 8.1 | 1,010.9 | 15 | 10.6 |
| 6. | Albuquerque | Dec 13, 2025, 2:28:22 PM | -0.6 | 1,015.3 | 52 | 5.2 |
| 7. | Los Angeles | Dec 13, 2025, 2:28:24 PM | 11.8 | 1,016.2 | 98 | 1.9 |
| 8. | San Diego | Dec 13, 2025, 2:28:25 PM | 12.6 | 1,016.5 | 100 | 3 |
| 9. | Phoenix | Dec 13, 2025, 2:28:26 PM | 9.9 | 1,015.4 | 70 | 7.5 |
| 10. | Las Vegas | Dec 13, 2025, 2:28:25 PM | 6.6 | 1,016.2 | 39 | 10.5 |

# Looker Dashboard - KPIs

### KPI #1: Real Time Temperature

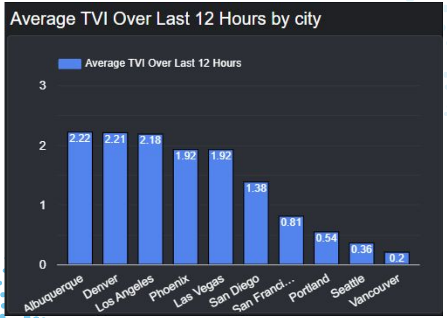


### KPI #2: Hourly Change in Temperature

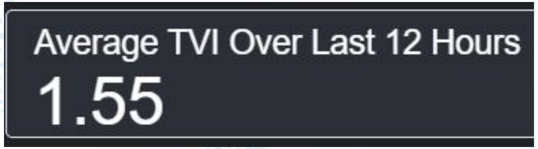


# Looker Dashboard - KPIs

### KPI #3: Average TVI Over Last 12 Hours by city



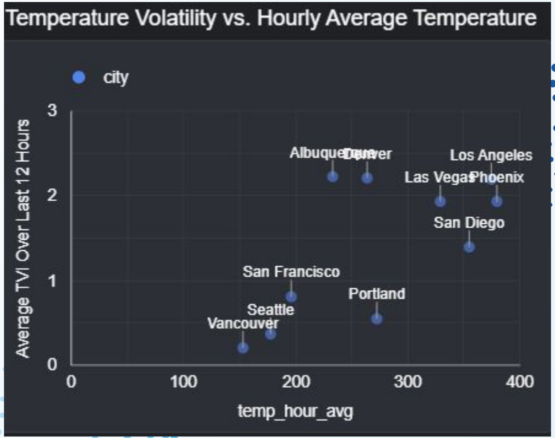
### KPI #4: Average TVI Over Last 12 Hours



Average TVI Over Last 12 Hours

1.55

### KPI #5: Temperature Volatility vs. Hourly Average Temperature



Looker Studio Dashboard

# Thank you for Listening!

Please check out our GitHub, and feel free to contact us if you have any questions!