Weather Trend Forecasting

PM Accelerator - Tech Assessment Report

Assessment Requirements Demo Walkthrough

Daniel Chahine

***PM Accelerator** - The PM Accelerator is a program designed to support the career development of aspiring and current product managers. It provides mentorship, resources, and real-world project experience to help individuals build the skills and portfolio needed to excel in product management roles. The mission is to bridge the gap between learning and doing by offering hands-on tech assessment opportunities in data science, engineering, and product development.*

BASIC 1/3: Data Cleaning & Preprocessing

Requirement

Handle missing values, outliers, and normalize data.

How it is met (src/cleaning.py)

> Column standardization: names to snake_case (standardize_columns)

Ensures consistent access; e.g. 'Last Updated' -> 'last_updated'.

> Datetime parsing: last_updated -> datetime + daily date column (parse_datetime)

Uses the 'lastupdated' feature as required for time-series analysis.

> Duplicate removal: keep latest record per (location, country, date) (remove_daily_duplicates)

> Missing-value handling: forward + backward fill per location (fill_missing_per_location)

Grouped by location_name so fills respect each city's own history.

> Outlier detection: IQR-based flag columns for 5 key variables (flag_iqr_outliers)

Columns flagged: temperature, humidity, precipitation, wind, pressure.

Uses 1.5x IQR rule; outliers are flagged but kept, not removed.

> Output saved as Parquet for fast reload (data/processed/weather_clean.parquet)

-> Cleaned dataset shape: 125,058 rows x 47 columns

-> Remaining missing values: 0

-> Rows flagged as outliers (any column): 30,019

BASIC 2/3: Exploratory Data Analysis (EDA)

Requirement

Perform basic EDA to uncover trends, correlations, and patterns.

Generate visualizations for temperature and precipitation.

How it is met (main.py -> stage_eda)

- Missing-value bar chart per column [missing]
- Temperature & precipitation histograms (REQUIRED) [exists]
- Daily temperature time series for 5 major cities [exists]
- Correlation heatmap of key numeric features [exists]
- Interactive notebook at notebooks/eda.ipynb for deeper exploration

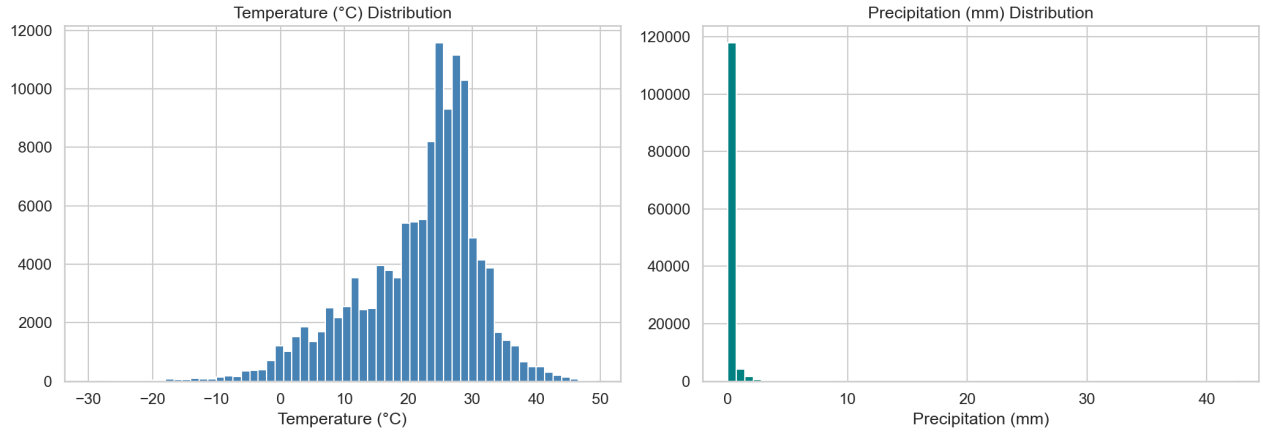

Figure: Temperature & Precipitation Distributions

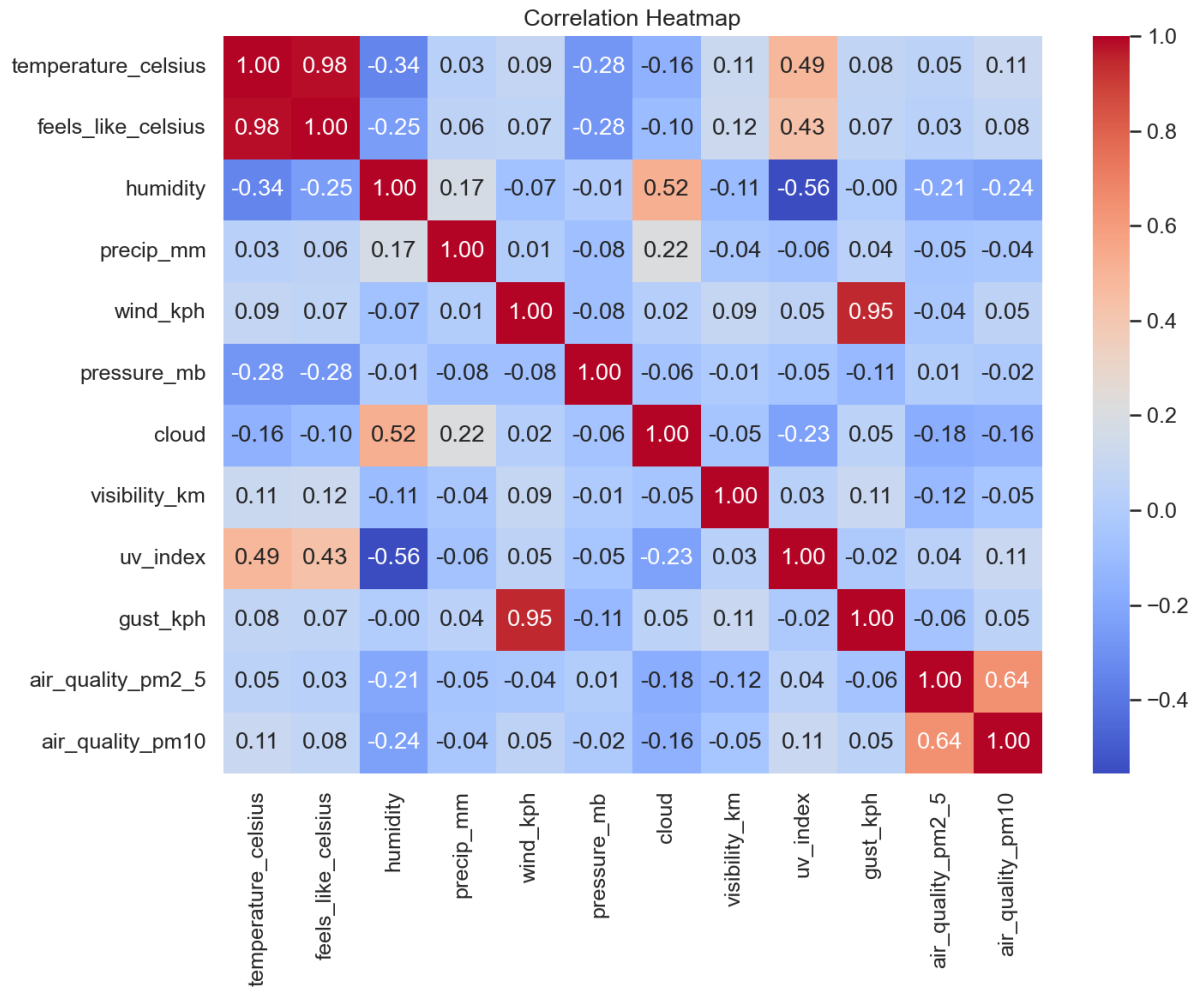

Figure: Correlation Heatmap

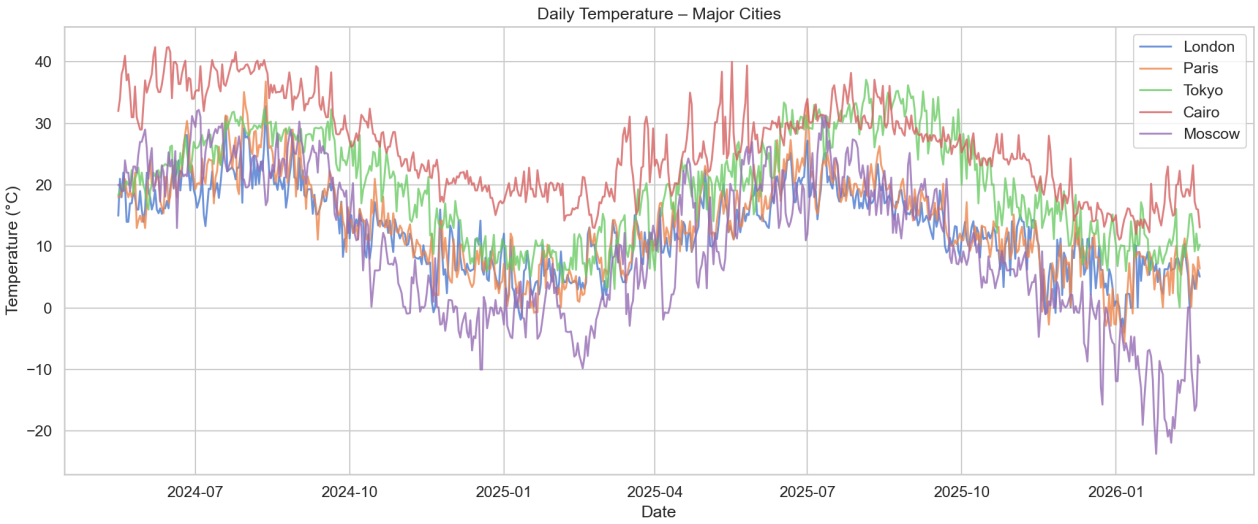

Figure: Daily Temperature - Major Cities

BASIC 3/3: Model Building & Evaluation

Requirement

Build a basic forecasting model and evaluate its performance using different metrics. Use lastupdated feature for time series analysis.

How it is met

- Time feature: 'last_updated' is parsed in cleaning.py -> 'date' column
  All time-series splits and forecasts use this date column.
- Chronological train/val/test split: 70 / 15 / 15 % (main.py -> _chrono_split)
  Prevents data leakage by never training on future data.
- Baseline models: Naive & Seasonal Naive (src/models/baseline.py)
  Naive repeats last value; Seasonal Naive repeats last 7-day cycle.
- Evaluation metrics: MAE, RMSE, sMAPE (src/evaluation.py)
  Three complementary metrics give a well-rounded view of accuracy.
- Forecast horizons: 7-day and 14-day ahead

Average Metrics Across 5 Cities

Model	Horizon	MAE	RMSE	sMAPE
Ensemble	7	1.8120	2.1651	7.7210
Ensemble	14	2.0233	2.4351	9.1318
ML_GBR	7	1.5965	1.9315	7.0786
ML_GBR	14	1.6124	1.9377	7.5005
Naive	7	3.2572	3.7393	13.4153
Naive	14	3.7543	4.3331	15.3684
Prophet	7	2.1656	2.5957	8.7693
Prophet	14	2.8168	3.3956	12.5539
SARIMA	7	2.8782	3.4253	12.0094
SARIMA	14	2.7980	3.3624	12.3379
SeasonalNaive	7	3.4514	4.1792	13.5846
SeasonalNaive	14	3.0971	3.8750	13.1150

ADVANCED 1/3: Anomaly Detection

Requirement

Implement anomaly detection to identify and analyze outliers.

How it is met (src/anomalies.py)

- STL Decomposition (stl_anomaly_detection)

  Decomposes time series into trend + seasonal + residual components.

  Flags residuals exceeding 3 sigma as anomalies.

- Isolation Forest (isolation_forest_anomalies)

  Unsupervised multivariate anomaly detection on temperature, humidity, precipitation, wind speed, and pressure. Contamination = 2%.

- IQR outlier flags already applied during data cleaning (Basic req.)

Figure: STL Anomaly Detection

ADVANCED 2/3: Multiple Models + Ensemble

Requirement

Build and compare multiple forecasting models.

Create an ensemble of models to improve forecast accuracy.

How it is met

> Naive Baseline (src/models/baseline.py)

Last observed value repeated

> Seasonal Naive (src/models/baseline.py)

Last 7-day cycle repeated

> SARIMA (src/models/arima.py)

(1,1,1)x(1,1,0,7) with automatic fallback

> Prophet (src/models/prophet_model.py)

Weekly + yearly seasonality via Facebook Prophet

> ML Gradient Boosting (src/models/ml_regression.py)

GBR with 22 engineered features

> Weighted Ensemble (src/models/ensemble.py)

Inverse-MAE weighted average of all models

> Model comparison charts saved for both horizons:

-> Best 7-day model: ML_GBR (MAE = 1.5965 C)

-> Best 14-day model: ML_GBR (MAE = 1.6124 C)

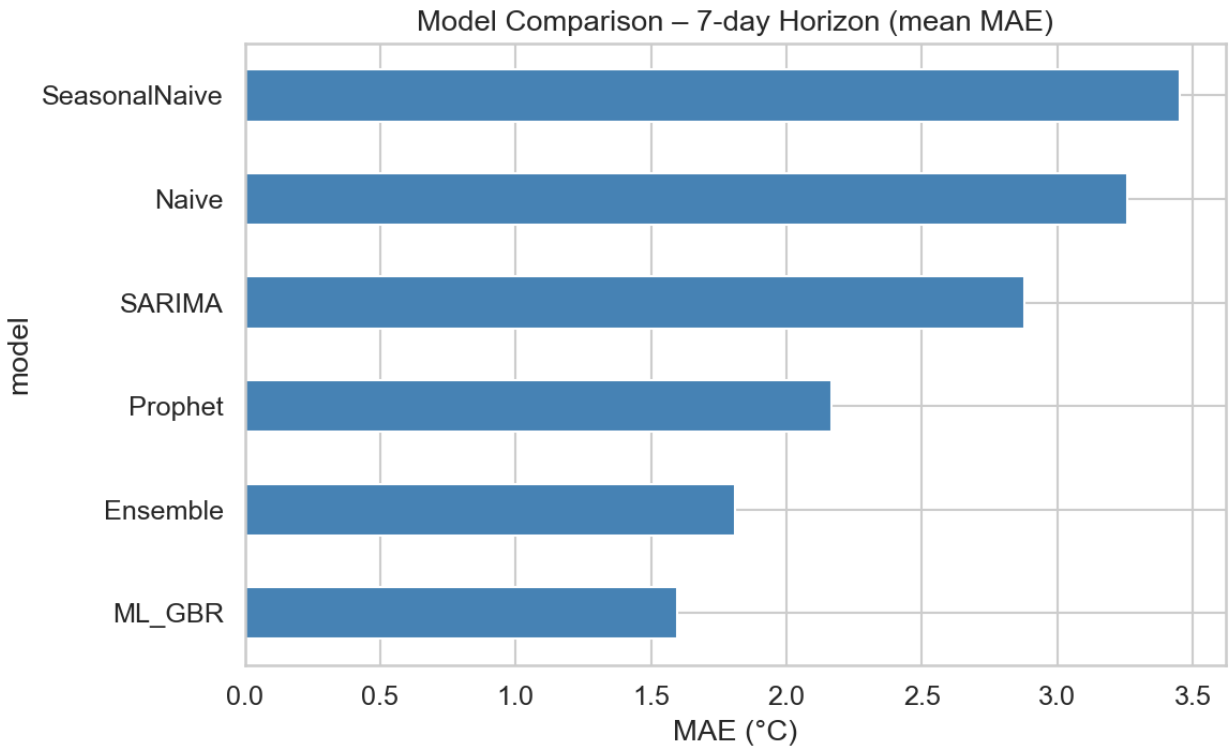

Figure: 7-Day Model Comparison

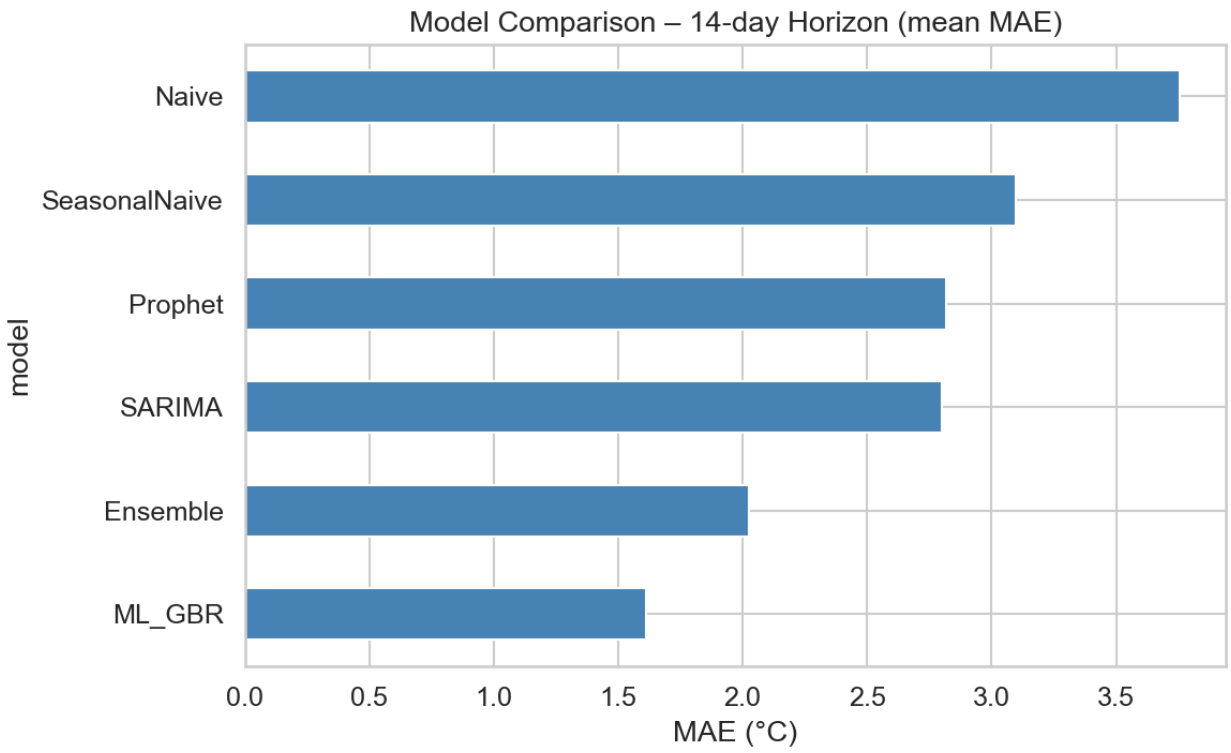

Figure: 14-Day Model Comparison

ADVANCED 3/3: Unique Analyses

A. Climate Analysis - Long-term patterns by region

Requirement: Study long-term climate patterns and variations in different regions.

- Monthly average temperature compared across 6 continents (main.py -> stage_advanced)
- Country -> continent mapping in src/utils.py (190+ countries)

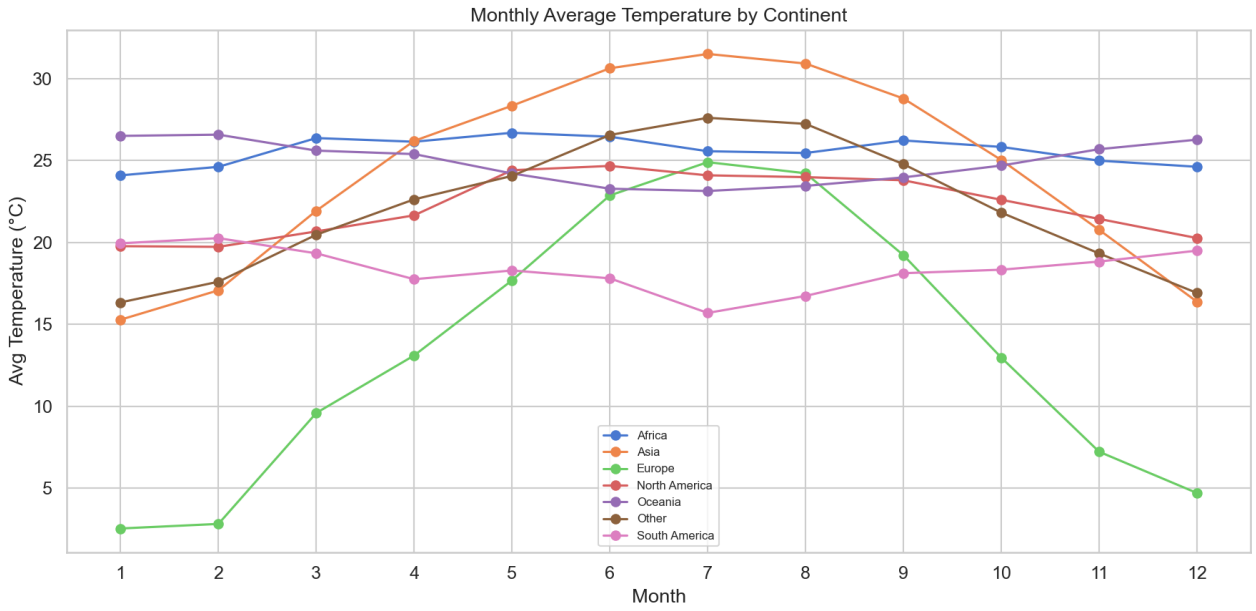

Figure: Monthly Climate by Continent

B. Environmental Impact - Air quality correlation

Requirement: Analyze air quality and its correlation with weather parameters.

- Correlation matrix: PM2.5, PM10, CO, NO2, SO2, O3 vs temp, humidity, wind, precip

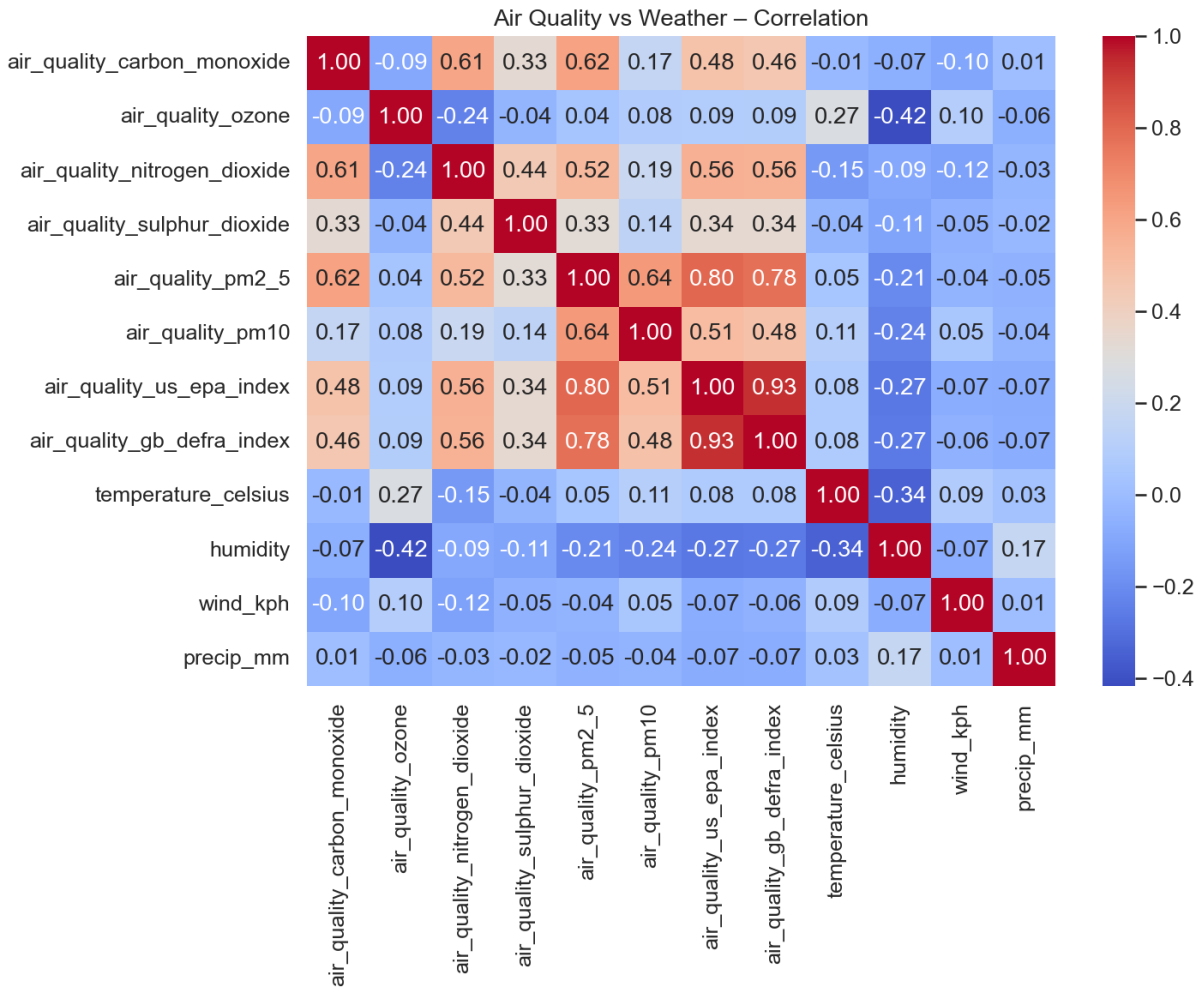

Figure: Air Quality vs Weather Correlation

C. Feature Importance

Requirement: Apply different techniques to assess feature importance.

- Gradient Boosting built-in feature_importances_ (src/models/ml_regression.py)
- 22 engineered features ranked: lags, rolling stats, calendar, lat/lon, weather

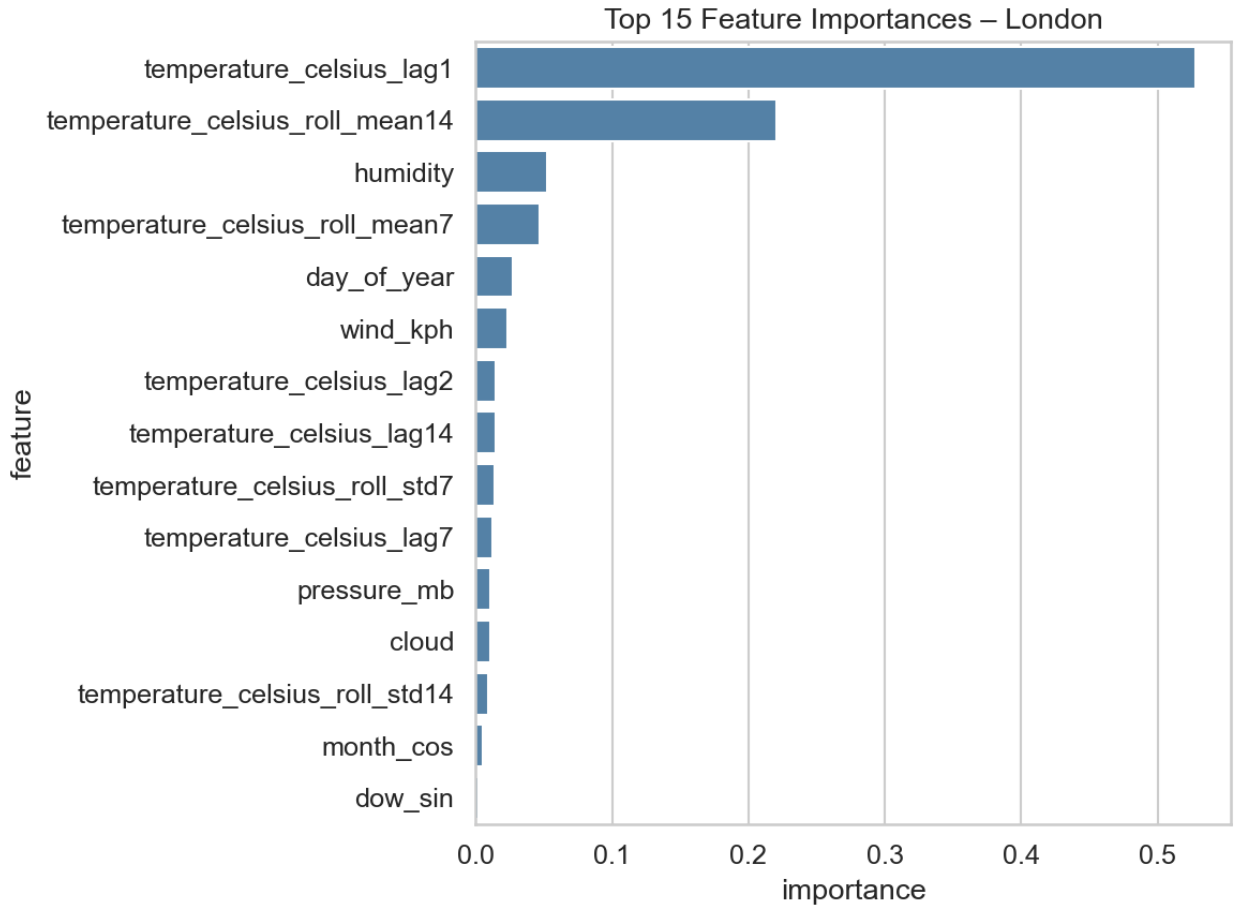

Figure: Feature Importance

D. Spatial Analysis - Geographic weather patterns

Requirement: Analyze and visualize geographical patterns in the data.

- Global scatter plot: lat/lon coloured by temperature on the latest date

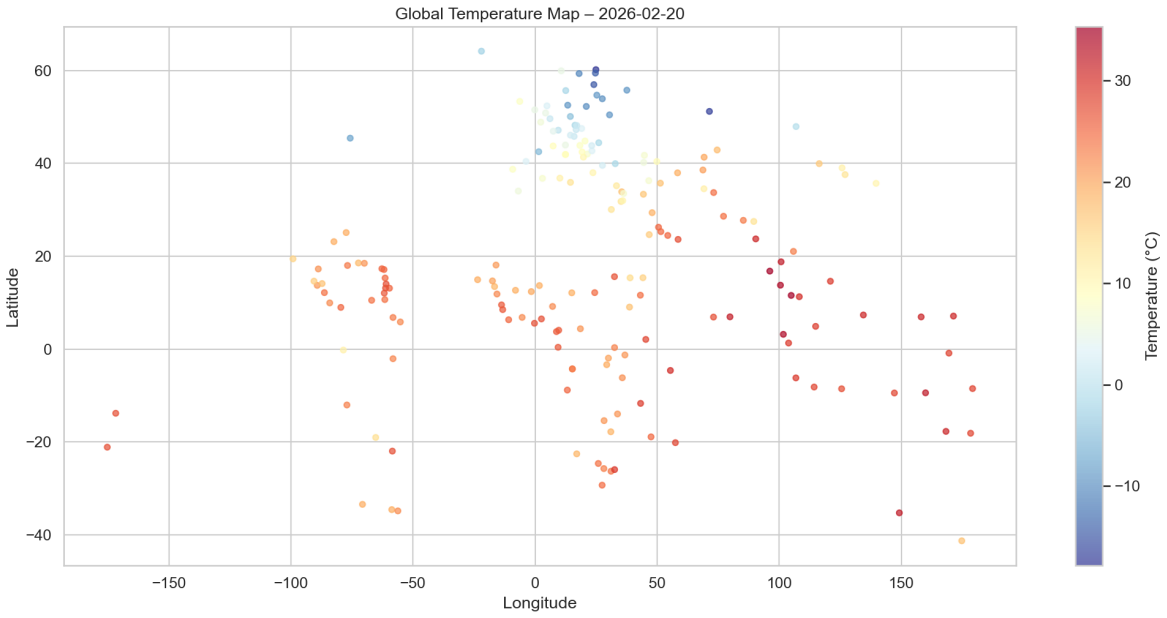

Figure: Spatial Temperature Map

E. Geographical Patterns - Cross-country/continent comparison

Requirement: Explore how weather conditions differ across countries and continents.

- Monthly climate by continent chart (see Climate Analysis above)
- Major-city time series comparison: London, Paris, Tokyo, Cairo, Moscow

Feature Engineering (src/features.py)

Features created for the ML model

- > Lag features: temperature at t-1, t-2, t-7, t-14 days
- > Rolling statistics: 7-day & 14-day rolling mean and std (shift=1 to avoid leakage)
- > Calendar features: day_of_week, month, day_of_year, is_weekend
- > Cyclical encoding: sin/cos transforms for month and day-of-week
- > Spatial features: latitude and longitude carried through
- -> Total engineered features used by GBR: 22

Deliverables Checklist

- > PM Accelerator mission displayed -> README.md (top)
- > Data cleaning & preprocessing -> src/cleaning.py
- > EDA with temp & precip visualizations -> main.py -> stage_eda + reports/figures/
- > Basic forecasting model + metrics -> src/models/baseline.py, src/evaluation.py
- > Anomaly detection (STL + Isolation Forest) -> src/anomalies.py
- > Multiple forecasting models compared -> src/models/*.py + reports/forecast_results.csv
- > Ensemble model -> src/models/ensemble.py
- > Climate analysis by region/continent -> main.py -> stage_advanced, src/utils.py
- > Air quality <-> weather correlation -> main.py -> stage_advanced
- > Feature importance analysis -> src/models/ml_regression.py
- > Spatial/geographic temperature map -> main.py -> stage_advanced
- > Geographical patterns across countries -> main.py -> stage_advanced
- > Well-organized README.md documentation -> README.md
- > GitHub repository with all code -> This repository
- > Reproducible pipeline (single command) -> python main.py

How to Reproduce

```
# 1. Install dependencies
  pip install -r requirements.txt

# 2. Run the full pipeline (cleaning -> EDA -> forecasting -> advanced)
  python main.py

# 3. Run the demo walkthrough
  python demo.py

# 4. Export this PDF report
  python export_pdf.py
```

Outputs

- Cleaned data -> data/processed/weather_clean.parquet
- Figures -> reports/figures/ (10 plots)
- Forecast table -> reports/forecast_results.csv
- EDA notebook -> notebooks/eda.ipynb
- PDF report -> reports/PM_Accelerator_Weather_Report.pdf

Summary

This project fulfills ALL requirements of the PM Accelerator Weather Trend Forecasting Tech Assessment:

Basic Assessment - data cleaning, EDA with temperature/precipitation visualizations, and a forecasting model evaluated with MAE/RMSE/sMAPE.

Advanced Assessment - anomaly detection (STL + Isolation Forest), six forecasting models compared side-by-side, a weighted ensemble, climate analysis, air-quality correlation, feature importance, spatial mapping, and geographical pattern analysis.

Deliverables - PM Accelerator mission in README, well-organized documentation, reproducible single-command pipeline, and all results saved to the reports/ directory.