**PM Accelerator Mission**

PM Accelerator is committed to making industry-leading tools and educational resources accessible to individuals from diverse backgrounds, thereby leveling the playing field for future product management leaders. Its mission is to empower aspiring and experienced product managers by granting them unparalleled access to industry expertise, fostering a robust product management ecosystem, and equipping them with advanced skills in AI-driven product management.

**Data Understanding & Cleaning**

The dataset is loaded from GlobalWeatherRepository.csv using pandas. This file contains 60,218 records and 41 columns that capture various weather-related metrics, geographic information, and air quality indicators

**Using df.info() and df.describe(),isnull() to understand the data**

* There are 41 columns with no missing values
* Data types include:

Object: 11 columns (e.g., country, location\_name, timezone, condition\_text, sunrise, sunset, etc.)

Float64:23 columns(eg, latitude,longitude, temperature\_celsius,)

Int64: 7 columns(last\_updated\_epoch, wind\_degree, humidity, cloud, and various air quality indexes).

**Converting Datetime Information**

The last\_updated column, originally loaded as an object, is converted into a datetime format. This is crucial for time series analysis and forecasting.

* Handling Duplicates

The descriptive statistics indicate that while the dataset contains 60,218 records, there are only 10,073 unique last\_updated timestamps. This suggests that many timestamps have duplicate records—likely because multiple readings were recorded within the same period

* Aggregation Strategy:

To prepare the data for time series analysis, duplicate timestamps are aggregated. For instance, we take the mean of temperature\_celsius for each unique timestamp. This ensures that the time series has a unique index

**Exploratory Data Analysis (EDA)**

Temperature Distribution

We created a histogram of all temperature values (temperature\_celsius). Most readings are between 10°C and 35°C, with a peak around 25–30°C. A few very low (below -20°C) and very high (above 40°C) temperatures show that the data covers a range of climates and possible extreme weather.

Time Series View

Plotting temperature over time (from May 2024 to April 2025) shows daily ups and downs, likely because the data includes different locations with varying climates. We notice some big drops below 0°C and spikes above 40°C, which may be real extremes or unusual measurements.

Initial Data Size

The dataset has 60,218 rows and 44 columns. There are no missing values, meaning every column has an entry for every row.

A blue graph with numbers

AI-generated content may be incorrect.

**Anomaly Detection**

We used an IsolationForest model to spot unusual temperature readings. This model labels points that seem very different from the rest of the data as “outliers.”

The model flagged **565** temperature readings as outliers, most of which are at the extreme high or low ends of the temperature scale. We highlighted these points in red on the time series plot.

After dropping these flagged outliers, our dataset went from 60,218 rows down to **59,623**. By removing them, we focus on the more typical temperature patterns, which can help produce clearer results in later steps like forecasting and advanced analysis.

Overall, the EDA shows a wide range of temperatures, and the anomaly detection step helps us clean the data by filtering out extreme values that might otherwise skew our models and insights

A blue and red graph

AI-generated content may be incorrect.

**Forecasting with Multiple Models**

**Time Series Preparation**

* The temperature data is grouped by date (e.g., taking the average daily temperature). This ensures each day has a single temperature value.
* The resulting time series is set to a daily frequency (.asfreq('D')), and any missing dates are filled using faorward-fill. This creates a continuous time series with no gaps.
* For the Prophet model, the time series is reformatted into two columns:ds: The date`,y: The target variable (daily temperature)

**ARIMA Forecasting**

ARIMA stands for **AutoRegressive Integrated Moving Average**.

It’s a classical time series model that uses past values and past errors to forecast future values.

1. **Model Training:**
   * The code specifies an ARIMA order (p=5, d=1, q=0), though these parameters can be tuned for better performance.
   * The model is fit on the historical daily temperatures.
2. **Forecast Generation:**
   * Once trained, ARIMA predicts temperature for the next 30 days (or your chosen forecast horizon).
   * A plot shows both the observed data and the ARIMA forecast in green.

A graph showing a graph

AI-generated content may be incorrect.

The ARIMA model predicts that temperatures will remain relatively steady at around 4–5°C in the months following March 2025. It does not show a rebound or further drop, suggesting the model expects minimal change in temperature going forward.

**Prophet Forecasting**

Prophet (developed by Facebook/Meta) is a time series forecasting tool that automatically handles trend, seasonality, and potential holiday effects.

It often requires less manual tuning than ARIMA and can be easier to interpret for some use cases.

**Model Training & Forecasting:**

The time series is loaded into Prophet by renaming columns to ds (date) and y (temperature).

Prophet is fit on this data, then we extend the timeline by 30 days to predict future temperatures.

A plot shows the observed data and Prophet’s forecast in orange.

A graph showing a graph

AI-generated content may be incorrect.

The **blue line** shows historical temperatures, which drop from around 20–30°C down to about 5°C near the end of the observed period.The **orange line** is the Prophet model’s forecast, predicting that temperatures will keep decreasing from about 5°C to roughly 2–3°C over the next few months. Essentially, Prophet expects the recent downward trend in temperature to continue.

**Ensemble Forecast**

1. **Why Ensemble?**
   * Different models capture different patterns or features in the data.
   * By **averaging** predictions from ARIMA and Prophet, we often get a more stable and accurate forecast.
2. **How it Works:**
   * Both ARIMA and Prophet produce a forecast for each future date.
   * The **Ensemble** column is simply the average of the two forecasts.
3. **Ensemble Plot:**
   * The final plot shows the observed data and the ensemble forecast (in purple).
   * This forecast can sometimes outperform each individual model by balancing their strengths and weaknesses.

A graph showing a graph

AI-generated content may be incorrect.

* Observed Data (Blue Line): Shows the historical temperatures, starting around 25–30°C and eventually dropping to around 5°C by early 2025.
* Ensemble Forecast (Purple Line): Combines predictions from two models (ARIMA and Prophet) by averaging their forecasts. It projects temperatures to stay near 5–7°C in the coming months and gradually shift downward.

Because it’s an ensemble (an average of two forecasts), the purple line tends to be more moderate than either individual model’s forecast. This approach often yields a more balanced prediction when different models capture different patterns.

**Comparison**

A screen shot of a graph

AI-generated content may be incorrect.

The chart shows that ARIMA and Prophet can produce markedly different forecasts if the data is limited, has strong seasonal components, or contains anomalies.

The ensemble helps mitigate these discrepancies by blending the forecasts, often resulting in a more stable prediction that can handle sudden shifts better than any single model.

The sharp negative dip from ARIMA signals a need to revisit model assumptions, data coverage (especially seasonality), and parameter tuning. Meanwhile, Prophet’s forecast remains near 25°C, suggesting it interprets the trend as relatively stable.

Final insight: Always compare forecasts to actual test data and refine model parameters. The ensemble approach is typically more robust and often yields better real-world performance.

**6.Unique Analyses**

**Annual Average Temperature Trend by Country**

1.Goal: To see how each country’s average temperature changes year by year.

2. The data is grouped by country and year ,computing the mean of temperature\_celsius.Each country is plotted as a separate line on the graph

A graph showing a number of colored lines

AI-generated content may be incorrect.

Interpretion

* + Lines that slope downward indicate countries where average temperatures decreased over the observed period.
  + Lines that slope upward suggest countries that experienced a rise in average temperatures.
  + Because there are many countries, the chart can look crowded. Interactive .or segmented views (e.g., by region) might be clearer

1. **Environmental Impact: Correlation Matrix**

Goal:

To understand the relationships between air quality (e.g., air\_quality\_PM2.5) and key weather parameters (temperature\_celsius, humidity, pressure\_mb, wind\_mph).

Method:

A correlation matrix is computed using df\_clean[cols\_for\_corr].corr().

A heatmap (plt.imshow) visualizes positive correlations (yellow) and negative correlations (purple/blue).

A chart of different colors

AI-generated content may be incorrect.

Interpretation:

Positive Correlation (close to +1): Variables tend to increase or decrease together.

Negative Correlation (close to -1): One variable increases while the other decreases.

Near Zero: Little to no linear relationship.

**3. Feature Importance (Random Forest)**

Goal:

To see which weather or air quality factors are most influential in predicting temperature.

Method:

A RandomForestRegressor is trained to predict temperature\_celsius using features like humidity, pressure\_mb, wind\_mph, and air\_quality\_PM2.5.

The model’s feature\_importances\_ attribute reveals how much each feature contributes to the prediction.

A graph of blue bars

AI-generated content may be incorrect.

Interpretation:

High Importance: The feature strongly influences temperature predictions.

Lower Importance: The feature has less effect on predicting temperature (but might still be relevant in specific contexts).

MSE (Mean Squared Error): Indicates how well the model performs overall. A lower MSE means better predictions.

**4. Spatial Distribution of Temperature**

Goal:

To visualize how temperature varies by geographic location (latitude, longitude).

Method:

A scatter plot is created where each point’s position is given by (longitude, latitude).

The color of each point (c=df\_clean['temperature\_celsius']) reflects the temperature, with a color scale (cmap='viridis') showing cooler to warmer temperatures.

A diagram of a temperature

AI-generated content may be incorrect.

Interpretation:

Clusters of similar color: Regions with similar temperatures (e.g., hot climates vs. cooler zones).

Geographic Patterns: Higher latitudes often correlate with lower temperatures, though local conditions can vary.

**5. Comparing Average Temperature by Country**

Goal:

To see which countries have the highest and lowest average temperatures in the cleaned dataset.

Method:

The data is grouped by country, taking the mean of temperature\_celsius.

A bar chart is plotted in descending order of average temperature.

A barcode with a red line

AI-generated content may be incorrect.

Interpretation:

Countries on the left (taller bars): Tend to have higher average temperatures.

Countries on the right (shorter bars): Tend to be cooler on average.

The chart provides a quick way to compare typical climate conditions across countries