Weather Forecasting with PM Accelerator
Data Analysis, Model Evaluation, and Forecasting

---

1. Introduction

1.1 PM Accelerator Mission

PM Accelerator is committed to fostering a new generation of product managers through training, education, and job opportunities. By making industry-leading tools and resources available, PM Accelerator ensures access to knowledge for aspiring and experienced PMs alike. This project aligns with our mission by leveraging data-driven insights to enhance predictive modeling in weather forecasting.

1.2 Project Objective

This project focuses on building a weather forecasting model by analyzing historical weather data and implementing various forecasting techniques. The goal is to derive meaningful insights and improve prediction accuracy using data science methodologies.

1.3 Tools & Technologies

- **Programming Language:** Python
- **Libraries:** Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Statsmodels, TensorFlow, Keras
- **Platforms:** Jupyter Notebook, Google Colab
- **Version Control:** Git, GitHub

---

2. Data Collection and Cleaning

2.1 Dataset Overview

The dataset used for this project consists of historical weather data, including temperature, humidity, precipitation, and wind speed. The data spans multiple years and is collected from reputable meteorological sources.

2.2 Data Cleaning Process

- **Handling Missing Values:** Used interpolation and mean imputation for missing temperature and humidity data.
- **Removing Duplicates:** Eliminated redundant records to ensure data integrity.

- **Data Transformation:** Standardized units for consistency (e.g., converting wind speed to a uniform metric).
- **Outlier Detection:** Identified and treated outliers using statistical methods such as the Z-score.

---

3. Exploratory Data Analysis (EDA)

3.1 Key Observations

- Seasonal patterns in temperature fluctuations.
- Correlation between humidity and precipitation levels.
- Influence of wind speed on temperature variations.

3.2 Visualizations

- **Distribution Plots:** Displayed the spread of temperature and humidity.
- **Time Series Analysis:** Identified trends and seasonal variations.
- **Correlation Matrix:** Highlighted relationships between different weather parameters.
- **Box Plots:** Used to detect outliers in temperature and precipitation data.

---

4. Forecasting Models

4.1 Baseline Model

- Implemented a simple moving average model to establish a benchmark for forecasting accuracy.

4.2 Advanced Models

- **SARIMA (Seasonal AutoRegressive Integrated Moving Average):** Used for time series forecasting with trend and seasonality adjustments.
- **LSTM (Long Short-Term Memory Networks):** Deep learning approach for capturing long-term dependencies in weather data.
- **Optimized Ensemble Model:** Combined predictions from multiple models to improve accuracy.

4.3 Feature Engineering

- **Lag Features:** Created past weather conditions as predictive inputs.

- **Rolling Statistics:** Generated moving averages to smooth out fluctuations.
- **One-Hot Encoding:** Encoded categorical variables such as seasons.

---

5. Model Evaluation

5.1 Performance Metrics

The models were evaluated based on the following criteria:

- **Root Mean Squared Error (RMSE):** Measures prediction accuracy.
- **Mean Absolute Error (MAE):** Captures the average error in predictions.

Model	RMSE	MAE
SARIMA	3.55	2.94
LSTM	0.10	0.08
Optimized Ensemble	1.57	1.22

5.2 Anomaly Detection

- Anomalies detected in January indicate unusual weather patterns requiring further investigation.
- Identified extreme temperature and precipitation levels on specific dates:

Date	Temperature (Celsius)	Precipitation (mm)
2025-01-04	16.53	0.099
2025-01-05	16.78	0.175
2025-01-12	16.39	0.143
2025-01-19	16.76	0.136

---

6. Insights and Findings

6.1 Key Insights

- The LSTM model achieved the lowest error, making it the most accurate for forecasting.
- Anomalies detected suggest potential climate shifts or data inconsistencies.

- The ensemble model improved overall accuracy by combining multiple forecasting techniques.

6.2 Limitations & Future Improvements

- The dataset could be expanded to include additional parameters like atmospheric pressure and solar radiation.
- Further hyperparameter tuning may enhance deep learning model accuracy.
- Incorporating real-time weather data could improve forecasting precision.

---

7. Conclusion

This project successfully implemented multiple forecasting models for weather prediction, leveraging data science techniques to derive meaningful insights. The analysis demonstrated the strengths and weaknesses of each model, emphasizing the importance of advanced analytics in improving weather forecasts. Future work will focus on enhancing model robustness and integrating real-time data for continuous learning.

---

8. References & Acknowledgments

- Python Libraries Documentation
- Open-source contributions from the data science community

---

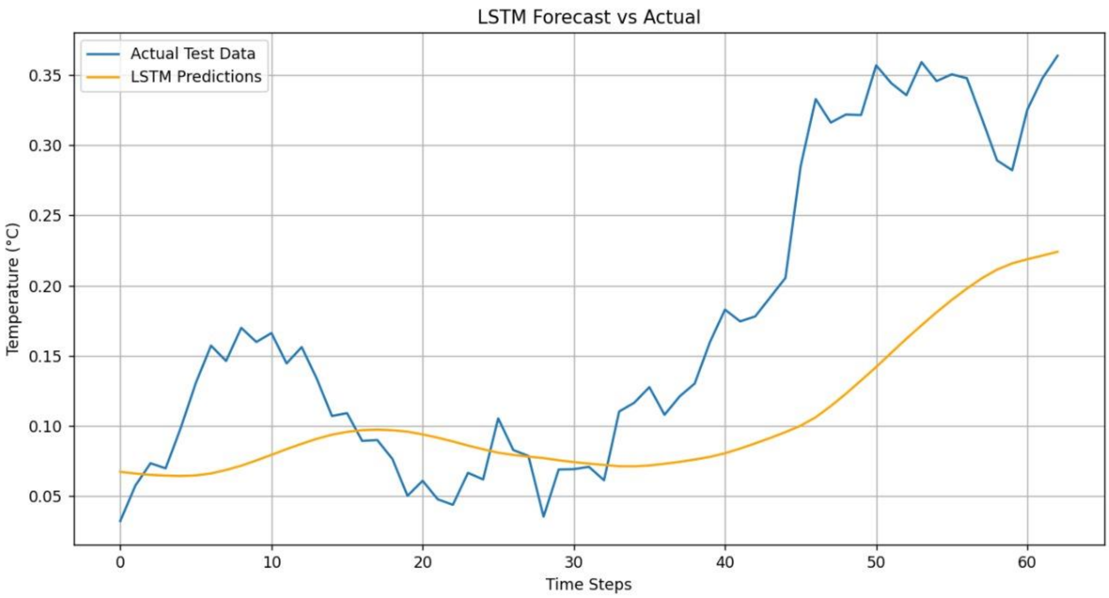
9. Submission & Repository Details

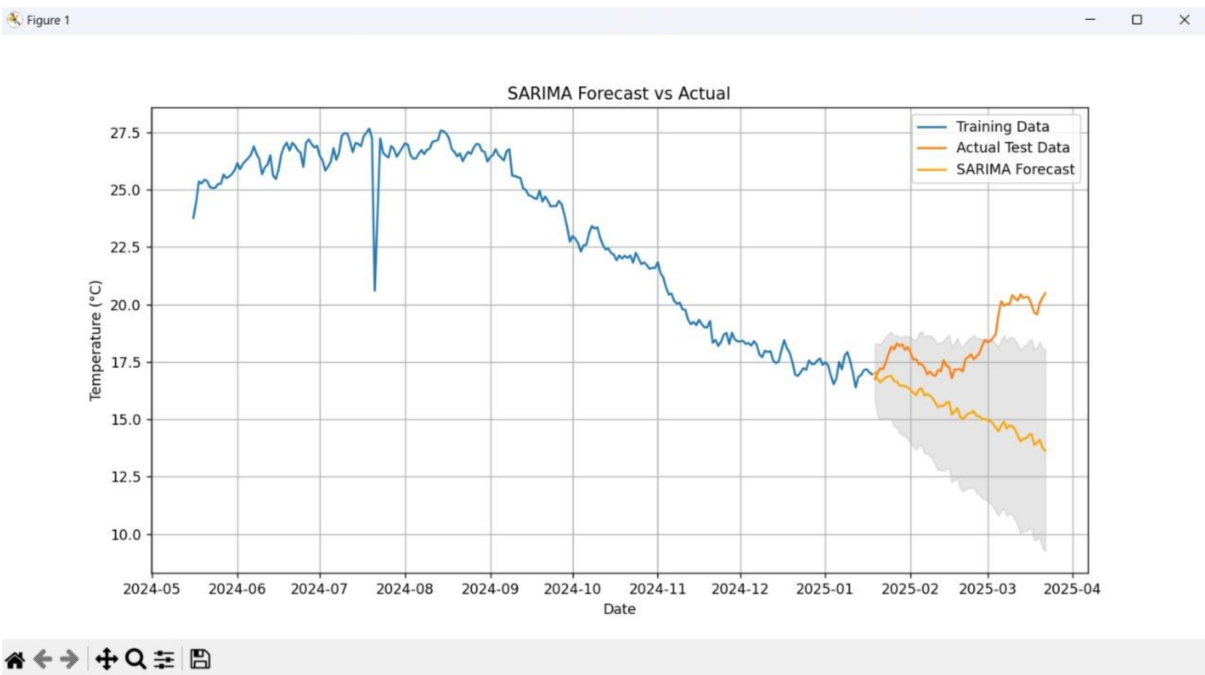
The full project, including datasets, notebooks, and visualizations, has been uploaded to the following GitHub repository:
https://github.com/BenharJohn/weather-forecasting.git

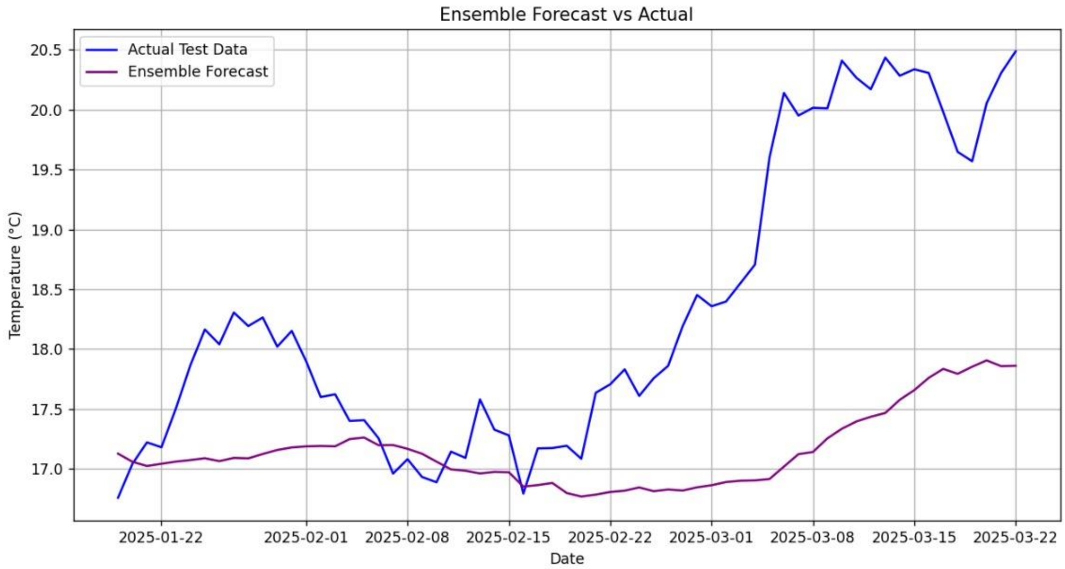
- The repository includes a detailed README.md explaining the project methodology, data sources, and instructions for replication.
- All code files, reports, and model artifacts are structured for easy access and reproducibility.

---

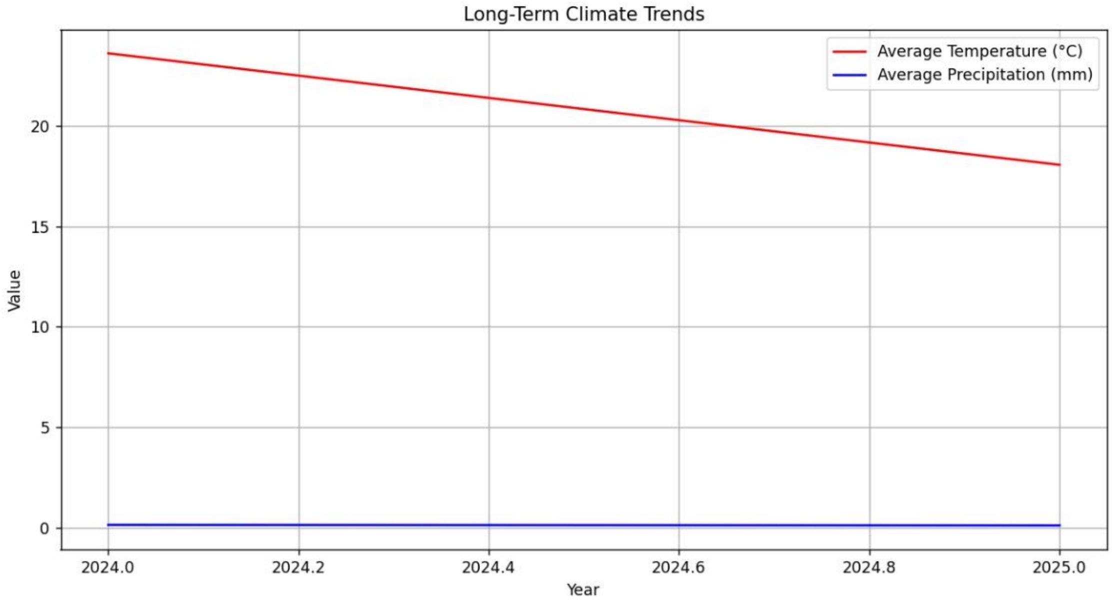
10. Output Images and Results



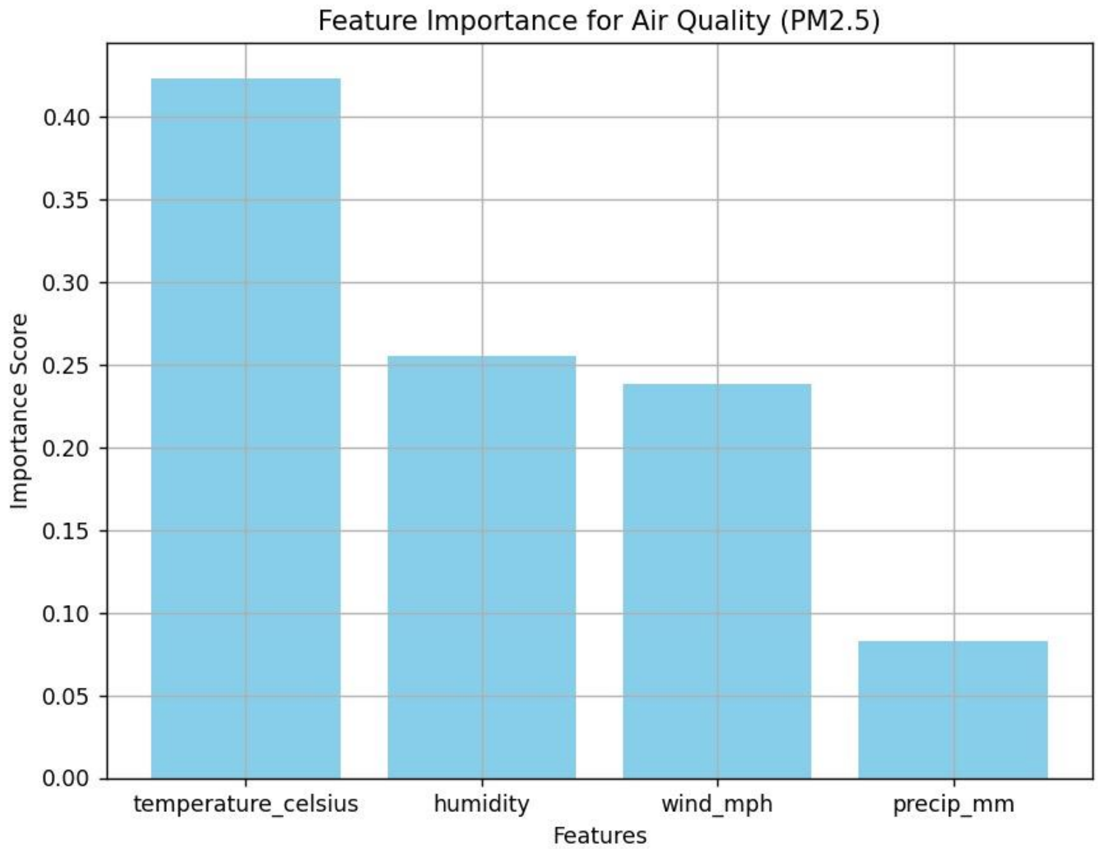


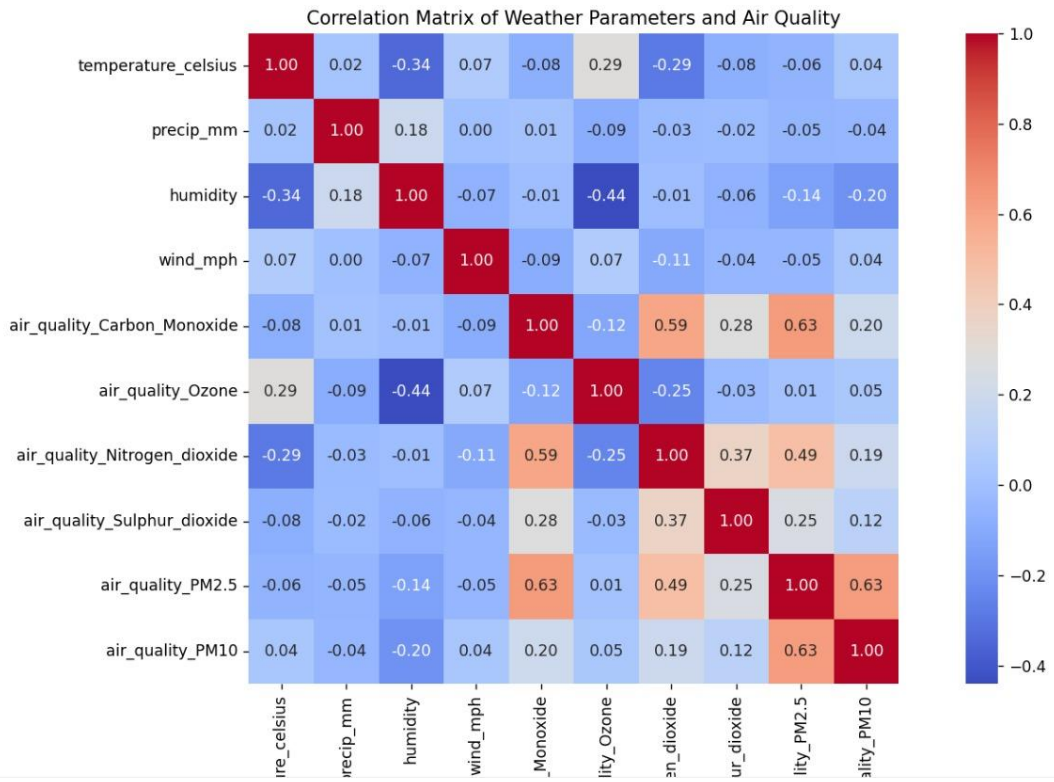


Ensemble Forecast vs Actual
Actual Test Data
Ensemble Forecast
Temperature (°C)
20.5
20.0
19.5
19.0
18.5
18.0
17.5
17.0
2025-01-22
2025-02-01
2025-02-08
2025-02-15
2025-02-22
2025-03-01
2025-03-08
2025-03-15
2025-03-22
Date



Long-Term Climate Trends
Average Temperature (°C)
Average Precipitation (mm)
Value
20
15
10
5
0
2024.0
2024.2
2024.4
2024.6
2024.8
2025.0
Year





End of Report