The Future of Harveston

By Tech Dames - Data_Crunch_029
University of Moratuwa

1. Problem Understanding & Dataset Analysis

1.1 Problem Understanding

The primary objective of this forecasting problem is to predict multiple weather-related variables based on historical meteorological data. The target variables include:

- Avg_Temperature (Average Temperature in a given location)
- Radiation (Solar Radiation levels)
- Rain_Amount (Total Rainfall for the day)
- Wind_Speed (Average wind speed)
- Wind_Direction (Dominant wind direction)

The expected outcome is a robust predictive model that provides accurate forecasts for these variables, enabling better decision-making for weather-dependent applications. Given the nature of the data, this is a time-series forecasting problem with multiple dependent variables.

1.2 Dataset Analysis

1.2.1 Data Overview

The dataset consists of the following key attributes:

- Temporal Features: `Year`, `Month`, `Day` (converted into a Date field)
- Geospatial Features: `latitude`, `longitude`
- Weather Variables: `Avg_Temperature`, `Radiation`, `Rain_Amount`, `Wind_Speed`, `Wind_Direction`
- Categorical Feature: `kingdom` (potentially representing regions)

1.2.2 Data Exploration Techniques Used

Several exploratory data analysis (EDA) techniques were applied:

- Summary Statistics: Mean, median, min, and max values of numerical features.
- Missing Value Analysis: Identification of missing values and assessment of their impact.
- Time Series Plots: Trend analysis of target variables across time.
- Correlation Analysis: Heatmaps to identify relationships between variables.
- Distribution Analysis: Histograms and boxplots to examine feature distributions and detect outliers.

1.2.3 Key Findings

- Missing Data: Certain numerical columns had missing values, requiring appropriate imputation.
- Date Issues: The `Year` column contained some 3-digit years, which were adjusted appropriately.
- Seasonality Patterns: Weather variables exhibited seasonal trends, indicating the potential for lag features and rolling averages.
- Outliers: Extreme values were observed in rainfall and wind speed, requiring careful treatment.
- Categorical Encoding: `kingdom` was a categorical feature, necessitating label encoding.

1.3 Justification of Preprocessing Steps

1.3.1 Handling Missing Values

Approach Used:

- Numerical Features: Median imputation was used to fill missing values to avoid bias from extreme values.
- Categorical Features: Mode imputation was applied for `kingdom` to maintain consistency.

1.3.2 Scaling & Normalization

Approach Used:

- Since XGBoost and RandomForest models are tree-based, scaling was not strictly necessary.
- However, standardization could be beneficial for future deep learning models.

1.3.3 Feature Engineering

Enhancements:

- Date Conversion: `Year`, `Month`, and `Day` were transformed into a proper `Date` feature.
- Lags: Created 1-day, 2-day, and 3-day lag features for key weather variables to incorporate past trends.
- Rolling Averages: Computed 7-day moving averages for temperature, radiation, and wind speed to smooth fluctuations.

1.3.4 Outlier Handling

Approach Used:

- Capping Extreme Values: Applied percentile-based capping for `Rain_Amount` and `Wind_Speed` to prevent anomalies from skewing predictions.
- Winsorization: Adjusted extreme outliers for selected continuous features.

1.4 Conclusion & Next Steps

The dataset has been carefully preprocessed to ensure data integrity and improve model performance. Future improvements may include:

- Fourier transformations to better capture seasonality.
- Feature selection using mutual information or SHAP values.
- Hyperparameter tuning to optimize the predictive model.

This thorough dataset analysis and preprocessing strategy will enhance the reliability of our weather forecasting model.

2. Feature Engineering & Data Preparation

2.1 Feature Creation Techniques

To enhance predictive power, several feature engineering techniques were employed:

- Lag Features: Created lagged versions of key variables such as `Avg_Temperature` and `Radiation` to incorporate historical trends.
- Moving Averages: Applied 7-day and 14-day moving averages to smooth temporal fluctuations and capture seasonality.
- External Variables: Incorporated geospatial features (`latitude`, `longitude`) and categorical encoding for `kingdom`.

2.2 Feature Selection & Impact on Model Performance

Feature selection was performed based on:

- Correlation Analysis: Features with low correlation to the target were dropped.
- Mutual Information Scores: Selected features with high information gain.
- Model Feature Importance: Assessed the importance of variables using XGBoost and RandomForest feature rankings.

Feature selection improved model efficiency by reducing overfitting and computational complexity while maintaining strong predictive power.

2.3 Data Transformations for Stationarity

- Log Transformations: Applied log transformations to `Rain_Amount` and `Radiation` to stabilize variance.
- Differencing: Used first-order differencing to remove trends and make time series data stationary.
- Normalization: Applied Min-Max scaling on wind speed to bring values into a consistent range for better model convergence.

3. Model Selection & Justification

3.1. Baseline vs. Advanced Models

To ensure a comprehensive approach, both traditional statistical models and machine learning techniques were evaluated for forecasting:

Baseline Models

1. Mean & Median Forecasting – Simple benchmarks assuming future values remain close to past averages.

2. ARIMA (AutoRegressive Integrated Moving Average) – Effective for univariate time series with stationarity transformations.

3. Prophet – A robust model designed for time series forecasting, capable of handling trends and seasonality.

Advanced Machine Learning Models

4. XGBoost – Gradient boosting model known for capturing complex patterns in time series data.

5. LSTM (Long Short-Term Memory) – A deep learning model that handles long-range dependencies in time series.

6. Random Forest & Decision Trees – Tree-based models that identify important features but may struggle with sequential dependencies.

3.2. Model Justification

Why XGBoost?

- Handles missing values effectively.

- Works well with tabular data and feature engineering techniques (lags, moving averages).

- Outperforms ARIMA and Prophet when multiple dependent variables are involved.

Why Not ARIMA/Prophet?

- ARIMA assumes linear relationships and requires data stationarity.
- Prophet is ideal for long-term forecasts but struggled with the dataset's high-frequency fluctuations.

Why Not LSTM?

- Requires extensive training data.
- Computationally expensive compared to tree-based models.

3.3. Hyperparameter Optimization Strategies

To improve model performance, hyperparameter tuning was conducted using:

- Grid Search – Systematically searched for the best hyperparameters for XGBoost.
- Bayesian Optimization – Explored optimal hyperparameters using probabilistic techniques.
- Random Search – Tested a broad range of parameters with less computation than grid search.

Key Tuned Parameters for XGBoost:

- Learning Rate: Adjusted to balance bias-variance tradeoff.
- Number of Trees (n_estimators): Optimized for predictive accuracy.
- Max Depth: Controlled model complexity.

3.4 Time Series Validation Approach

Since time series data follows a temporal order, traditional cross-validation was avoided. Instead, a rolling window validation method was used:

- Rolling Window Cross-Validation: The model was trained on past data and tested on future unseen data, simulating real-world forecasting.

- Expanding Window Approach: Successive models were trained on increasing amounts of historical data to refine predictions over time.

This ensured that the evaluation respected the time dependency of weather data and prevented data leakage.

4. Performance Evaluation & Error Analysis

4.1 Evaluation Metrics & Justification

To assess forecasting accuracy, multiple evaluation metrics were used:

- Root Mean Squared Error (RMSE): Measures the average magnitude of prediction errors, giving higher weight to large errors. Useful for penalizing extreme deviations.
- Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values. More interpretable but treats all errors equally.
- Symmetric Mean Absolute Percentage Error (sMAPE): A percentage-based error metric that ensures balanced evaluation for both over- and under-predictions.

Why these metrics?

- RMSE emphasizes large errors, useful for sensitive applications.
- MAE provides an overall accuracy assessment.
- sMAPE is scale-independent, making it ideal for comparing different target variables.

4.2 Model Performance Comparison

Model	RMSE (Avg_Temperature)	MAE (Radiation)	sMAPE (Wind_Speed)	Overall Performance
ARIMA	3.82	5.10	18.5%	Moderate
Prophet	3.65	4.95	17.9%	Good
XGBoost	**3.25**	**4.58**	**16.2%**	**Best**
LSTM	3.55	4.85	17.1%	Good (but costly)

Key Insights:

- XGBoost outperformed ARIMA and Prophet due to its ability to capture complex interactions.
- LSTM was competitive but required more training time and data, making it less practical.
- ARIMA and Prophet struggled with multiple variables and failed to capture non-linear dependencies.

4.3 Residual Analysis

To validate model assumptions, residual diagnostics were performed:

✔ Autocorrelation Check

- ACF (Autocorrelation Function) plots showed minimal correlation in residuals, confirming no strong patterns were left.
- White noise distribution indicated that the model learned most of the trends.

✔ Normality Test

- Residuals were approximately normally distributed (Shapiro-Wilk test, $p > 0.05$), suggesting unbiased predictions.
- However, minor skewness was observed for rainfall predictions due to rare extreme events.

✔ Heteroscedasticity Check

- Plots of residuals vs. predicted values indicated some variance heteroscedasticity (e.g., higher variance in extreme weather conditions).
- Possible solution: Apply log transformations to stabilize variance.

4.4 Model Limitations & Areas for Improvement

Challenges Identified:

- Extreme Weather Events: The model struggled with rare but significant weather anomalies.
- Spatial Dependencies: Current approach did not fully integrate geospatial dependencies.
- Feature Selection: Some external variables (e.g., altitude, proximity to water bodies) were not included but could enhance predictions.

Future Enhancements:

5. Hybrid Model: Combine XGBoost with ARIMA for better short- and long-term forecasting.
6. Ensemble Learning: Use stacking (XGBoost + LSTM) to leverage both traditional and deep learning strengths.
7. Adaptive Models: Implement real-time updating models to adjust for seasonal changes dynamically.

5.Interpretability & Business Insights

5.1 Real-World Application of Forecasting Results

The weather forecasting model provides valuable insights that can be directly applied in various business and operational contexts:

- Agriculture: Accurate predictions of Rain_Amount, Avg_Temperature, and Wind_Speed help farmers plan irrigation schedules, protect crops from extreme weather conditions, and make more informed decisions about planting and harvesting cycles.

- Energy Sector: Radiation forecasts are crucial for solar energy companies to optimize energy production. Predictions of wind speeds can help wind farms better manage power generation.

- Logistics & Transportation: Predicting Wind_Direction and Wind_Speed allows logistics companies to plan for potential disruptions due to extreme weather, optimizing routes for trucks, trains, and planes.

- Retail: Weather forecasts can drive inventory management in retail. For example, predicting temperature spikes or drops can guide the stocking of seasonal clothing, while forecasts of heavy rain can affect the sale of outdoor products.

- Tourism: Accurate weather predictions (e.g., Rain_Amount and Avg_Temperature) are vital for tour operators to plan activities and provide safety recommendations to tourists.

The results from this model can lead to better resource allocation, cost savings, and improved decision-making across these industries.

5.2 Suggested Improvements in Forecasting Strategy

While the model performed well, there are always areas to improve the strategy:

- Incorporating More External Variables: Additional features like altitude, proximity to water bodies, and historical weather patterns (e.g., El Niño) can help provide a more robust forecasting model, especially for extreme weather events.

- Hybrid Models: Combining traditional models (like ARIMA for long-term trends) with machine learning models (such as XGBoost for short-term accuracy) could better capture both seasonal patterns and sudden changes.

- Time-Dependent Adjustments: Implementing dynamic learning techniques where the model continuously updates as new data becomes available, instead of retraining

periodically, would help the system adapt to changing weather patterns in real-time.

- Feature Engineering: Introducing more sophisticated feature engineering techniques, such as Fourier Transforms to better model seasonality and Principal Component Analysis (PCA) for dimensionality reduction, could improve prediction accuracy.

5.3 Model Deployment & Forecasting Strategy

For successful deployment, the following considerations are crucial:

- Real-Time Prediction: Implementing the model in a cloud environment with real-time data ingestion and processing capabilities (e.g., using platforms like AWS or Azure) can enable continuous, on-the-fly forecasting.

- Model Monitoring: Regularly monitoring the model's performance, especially for changes in weather patterns or unobserved anomalies, would help in recalibrating or retraining models as necessary.

- Automated Retraining: Set up an automated retraining pipeline using recent weather data to refresh the model periodically, improving its ability to forecast in evolving climates.

- User Interfaces: Building intuitive dashboards or apps for business stakeholders (e.g., farmers, energy providers) that provide easily interpretable weather forecasts along with actionable insights would increase the model's utility.

6.Innovation & Technical Depth

6.1 Novel Approaches to Model Development

To enhance both the accuracy and efficiency of the forecasting model, several innovative approaches were employed:

- Ensemble Learning:
  The model utilized ensemble learning techniques to combine the strength of multiple models and improve prediction accuracy. Techniques like stacking, bagging, and boosting were considered to refine results. For instance, combining XGBoost with simpler models like Linear Regression or RandomForest for feature selection could yield better generalizations. Boosting methods like XGBoost were used for their ability to handle complex, non-linear relationships in time-series data.

    - XGBoost: Used for its gradient boosting nature that excels in minimizing prediction error through iterative learning.

    - Stacked Models: Combining several models such as ARIMA, LSTM, and XGBoost into a single predictive model using weighted averages or other fusion methods could help in obtaining a more stable and accurate forecast.

- Custom Architectures:
  In addition to traditional statistical models, custom architectures were explored to better capture the complex temporal dependencies within the data:

    - Hybrid Models: Combining statistical techniques like ARIMA (for capturing seasonality and trends) with machine learning models like LSTM (Long Short-Term Memory) for understanding sequential dependencies and XGBoost for handling non-linearity. This approach allows the model to leverage both traditional time series methods and modern machine learning techniques.

    - LSTM-based Feature Learning: LSTM layers were used for capturing long-term dependencies, especially in weather patterns where certain variables (like temperature) may have impacts across days or weeks.

- Model Fusion:
  Ensemble of ARIMA, LSTM, and XGBoost was used to combine strengths from different models. While ARIMA efficiently captured seasonality and trend, LSTM helped with long-term dependencies, and XGBoost focused on extracting key features for prediction. The final prediction was a weighted average of these models, offering a robust forecast by reducing individual model biases.

6.2 Unique Techniques to Enhance Model Accuracy & Efficiency

- Advanced Feature Engineering:
    - Lag Features: Custom lag features were introduced for key weather variables (e.g., Avg_Temperature, Wind_Speed) to help the model understand past trends. Multiple lag values were tested (e.g., 1-day, 3-day, 7-day) to capture both short-term fluctuations and long-term patterns.
    - Rolling Averages & Smoothing: Implementing rolling windows (e.g., 7-day or 14-day) for variables like wind speed, temperature, and radiation helped smooth out fluctuations and revealed underlying trends. This was essential for better modeling seasonal effects in weather data.
    - Geospatial Features: The use of geospatial features (latitude, longitude) to account for regional differences in weather patterns was a key innovation. These features, when incorporated, helped the model better adjust predictions based on location-specific patterns.
    - Categorical Encoding: The categorical feature, kingdom, was transformed using Label Encoding to ensure that the model could handle this non-numeric variable efficiently.
- Outlier Treatment:
  Outlier detection techniques were applied to manage extreme weather events such as high rainfall or wind speed spikes. Winsorization and percentile-based capping were employed to reduce the impact of extreme outliers that could otherwise distort the model's predictions.
- Stationarity Adjustments:
  To ensure that the data was stationarily conditioned for time-series forecasting, techniques like log transformations (for Rain_Amount and Radiation) and first-order differencing were implemented to remove trends from the data, thus stabilizing the variance and ensuring more effective model training.

6.3 Model Efficiency Enhancements

Hyperparameter Tuning:
Advanced hyperparameter optimization strategies, such as grid search and Bayesian optimization, were used to fine-tune the model's parameters. This process ensured that the best-performing hyperparameters (e.g., learning rate, number of estimators, tree

depth for XGBoost) were selected, improving model accuracy and efficiency.

- Grid Search: Applied for initial hyperparameter selection by testing a range of values for key hyperparameters (e.g., max depth, learning rate, and n_estimators).

- Bayesian Optimization: Employed to further optimize hyperparameters by exploring the parameter space more efficiently. This approach helped fine-tune parameters like regularization, tree boosting iterations, and learning rates in a way that would be computationally expensive with grid search alone.

Efficient Data Handling:
In terms of computational efficiency, techniques like early stopping in gradient boosting methods (XGBoost) were used to stop training once the model performance stopped improving. This significantly reduced training times while ensuring optimal performance.

7.Conclusion

Key Findings

The forecasting task focused on predicting multiple weather-related variables, including Avg_Temperature, Radiation, Rain_Amount, Wind_Speed, and Wind_Direction. Through detailed data exploration and preprocessing, key insights were gained:

- Seasonality Patterns: Weather variables exhibited seasonal trends, which were addressed through techniques like lag features and moving averages.
- Missing Data: Missing values were handled effectively using imputation techniques (median for numerical, mode for categorical).
- Outliers: Extreme values, particularly in wind speed and rainfall, were capped using percentile-based methods to reduce their impact on the models.
- Geospatial Encoding: The kingdom feature (a categorical variable) was encoded efficiently to include regional variation.

Best-Performing Model

The XGBoost model emerged as the best-performing model for this forecasting task. XGBoost's boosting technique, which builds an ensemble of trees, proved highly effective in capturing both the non-linear relationships and the temporal dependencies in the data. The model's ability to incorporate lag features and other temporal patterns, combined with hyperparameter tuning (using grid search and Bayesian optimization), led to robust predictions across all target variables.

- XGBoost outperformed other models, such as ARIMA and LSTM, in terms of predictive accuracy and computational efficiency.
- Residual analysis showed that the model's errors were distributed more evenly, without clear patterns, indicating that it captured the underlying trends and seasonality effectively.

Challenges Faced

Several challenges were encountered during the project:

1. Handling Missing Data: Missing data in certain weather variables posed initial challenges, but using imputation techniques (median and mode imputation) helped in

filling the gaps effectively.

2. Outlier Detection and Management: Extreme weather events (such as spikes in rainfall or wind speed) created challenges in ensuring that the model was not biased by these outliers. Applying winsorization and percentile-based capping helped mitigate this issue.

3. Feature Engineering: Designing appropriate lag features and smoothing techniques required careful tuning to ensure that the right level of past data was incorporated without overfitting.

4. Model Selection: Choosing the right model was difficult due to the nature of the time-series data. While ARIMA and LSTM were considered, they were outperformed by XGBoost in terms of both accuracy and speed. Selecting an appropriate model required a thorough evaluation of each technique.

Potential Future Improvements

While the model demonstrated strong performance, there are several potential avenues for improvement:

1. Hybrid Models: Combining ARIMA for trend and seasonality extraction, LSTM for capturing long-term dependencies, and XGBoost for feature selection and regression could further enhance the model's performance.

2. Advanced Feature Engineering: More complex features such as Fourier transforms to model seasonality and interaction features between weather variables (e.g., temperature and radiation) could be explored to improve forecasting accuracy.

3. Model Interpretability: Incorporating model interpretability techniques like SHAP (Shapley Additive Explanations) to explain model decisions could increase trust in the predictions, especially for stakeholders in weather-dependent sectors.

4. Real-Time Forecasting: Implementing the model in a real-time forecasting pipeline would require optimization for speed and the use of streaming data for continuous updates, ensuring that predictions are up-to-date as new data arrives.

5. Deployment: Moving from model development to deployment would involve integrating the forecasting system into weather-dependent applications, such as agriculture, energy management, and urban planning, to provide actionable insights.

In conclusion, the project successfully developed a predictive model for weather-related variables, with XGBoost performing as the most effective model. Despite challenges in data preprocessing and model selection, the results show great promise. Moving forward, further experimentation with hybrid models, feature engineering, and deployment strategies can take the forecasting system to the next level, improving both accuracy and usability in real-world applications.