Technical Report
Data_Crunch_102

Technical Report: Harveston Climate Prediction

---

1. Problem Understanding & Dataset Analysis

1.1 Problem Definition & Objectives

Harveston, an agricultural society, faces significant challenges due to increasingly unpredictable climate patterns (rainfall, temperature, wind), rendering traditional farming knowledge insufficient. This environmental volatility threatens Harveston's food security and economic stability. While over a decade of environmental data exists, its complexity and potential cross-kingdom inconsistencies (e.g., units) necessitate advanced machine learning techniques.

The primary project objective is to develop time series forecasting models to accurately predict five critical environmental variables for future dates provided in test.csv:

1. Average Temperature (°C)
2. Radiation (W/m2)
3. Rain Amount (mm)
4. Wind Speed (km/h)
5. Wind Direction (°)

1.2 Expected Outcomes

Successful models will provide actionable climate foresight, enabling Harveston's farmers and decision-makers to optimize planting/harvesting cycles, improve resource management (e.g., water allocation), and proactively prepare for weather extremes. This contributes directly to enhanced food security, economic stability, and overall agricultural resilience.

1.3 Dataset Overview & Initial Assessment

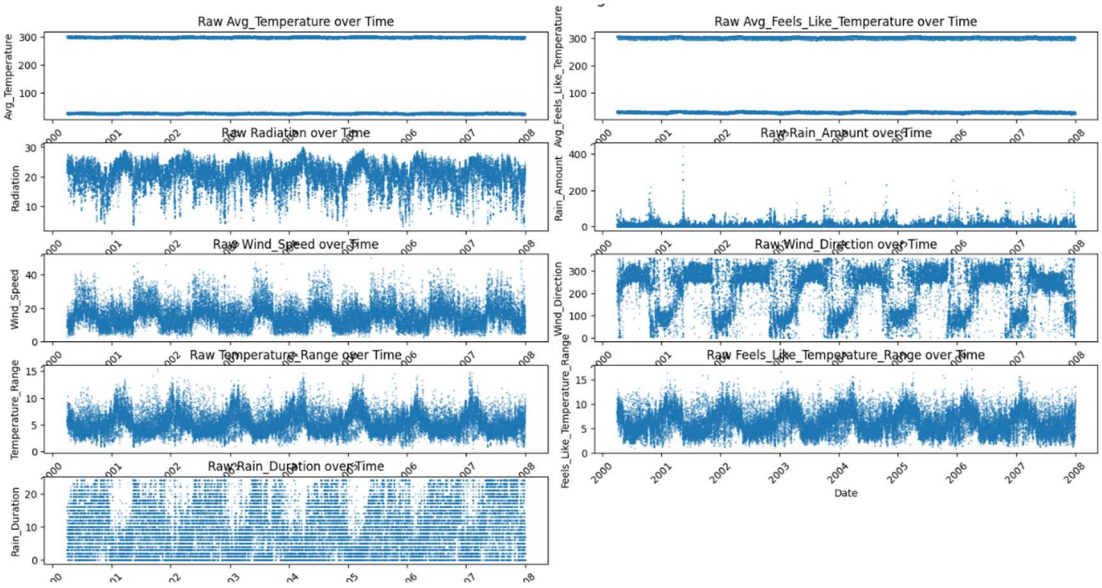
The analysis utilizes historical environmental data (train.csv) collected over 10+ years across various Harveston kingdoms, including temporal (Year, Month, Day), geospatial (latitude, longitude), and meteorological variables (temperatures, radiation, rain, wind, etc.).

A key noted challenge is inconsistent temperature units (°C or K). The test.csv file provides identifiers and future dates for prediction.

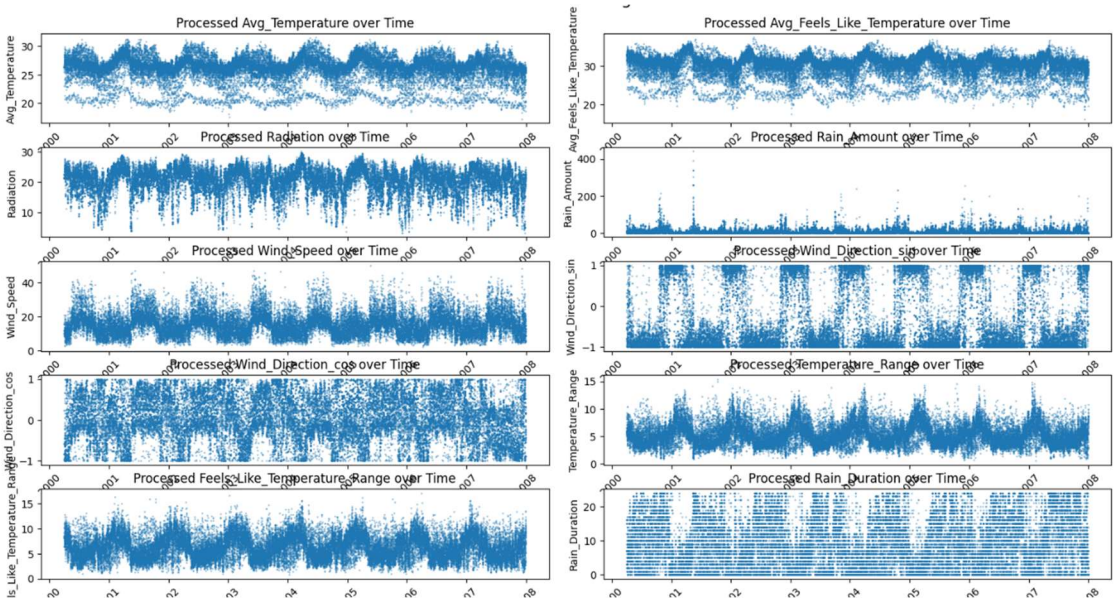
Initial assessment indicated the need for rigorous preprocessing to ensure data consistency and suitability for modeling. The following steps were essential:

- **Date Consolidation:** Combined Year, Month, Day into a single datetime column (ds), the standard for Prophet. Adjusted Year relative to a base year (2000) for timeline consistency. Rows with invalid dates were removed. *Justification: Creates a required, unambiguous time index for time series analysis.*

- **Unit Standardization (Temperature):** Converted temperature readings suspected to be Kelvin (values > 150) to Celsius (°C = K - 273.15). *Justification: Ensures consistent units (°C) required for the target variable and meaningful analysis.*
- **Missing/Invalid Data Handling:** Replaced infinite values (inf, -inf) with NaN. Rows with NaN in essential target or potential regressor columns were dropped *before* feature engineering. *Justification: Ensures numerical stability and valid inputs for subsequent feature creation, prioritizing data quality over imputation at this stage.*
- **Outlier Mitigation:** Clipped target variable values (y) at the 1st and 99th percentiles immediately before model fitting (optimization and final training). *Justification: Reduces the influence of extreme outliers for more robust model fitting without removing data points.*
- **Numeric Type Conversion:** Converted relevant columns to numeric types, coercing errors to NaN (handled in the missing data step). *Justification: Required data format for calculations and model algorithms.*



These steps transformed the raw data into a clean, standardized format ready for feature engineering and modeling.



2. Feature Engineering & Data Preparation

Following initial cleaning (Section 1.5), further feature engineering prepared the data for time series modeling, focusing on variable characteristics and temporal patterns.

2.1 Cyclical Feature Engineering: Wind Direction
To handle the cyclical nature of Wind Direction (0° ≈ 360°), the variable was transformed into two components: Wind_Direction_sin and Wind_Direction_cos.

- **Action:** Converted degrees to radians, then applied sin() and cos().
- **Justification:** This maps the data onto a circle, preserving angular proximity and allowing models to learn cyclical patterns effectively. These two components replaced the original Wind_Direction as modeling targets, reconstructed back to degrees post-prediction (Section 7).

2.2 Temporal Feature Engineering: Lags and Rolling Means
To capture historical context and trends, lagged values and rolling means were generated for the *modeling targets* (including Wind_Direction_sin/cos).

- **Action:** Created lag features (1, 2, 3, 7 days prior) and rolling means (3, 7, 14-day windows, shifted to prevent leakage). This was performed *per kingdom* using groupby('kingdom') to respect potential regional differences and avoid cross-region data leakage.
- **Handling NaNs:** Rows with resulting NaN values (at the start of each kingdom's series) were dropped to ensure complete records for modeling.
- **Justification:** Provides the model with auto-regressive information (lags) and smoothed recent trends (rolling means).

2.3 Feature Selection Strategy: Optimization vs. Final Prediction
A key strategic decision involved the use of engineered temporal features (lags, rolling means) and other non-target environmental variables (e.g., Avg_Feels_Like_Temperature).

- **During Optimization (Optuna):** These features *were used* as external regressors. The rationale was to provide richer context to potentially help Optuna find better core Prophet parameters (trend, seasonality). Prior scales for different regressor groups (lag, roll, other) were tuned separately.
- **Final Model Training:** The final Prophet models used for generating test predictions *were trained without any external regressors*. They relied solely on the timestamp (ds) and the target variable (y) for each kingdom-target pair.
- **Justification & Impact:** This simplifies the final prediction pipeline, avoiding the need to generate complex features for the test set or predict future regressor values. The trade-off is relying entirely on Prophet's tuned internal components, potentially sacrificing some accuracy for significant operational simplicity.

2.4 Data Transformations & Stationarity

- **Transformations Applied:** Unit conversion (Kelvin -> Celsius, Section 1.5), cyclical encoding (Degrees -> Sin/Cos, Section 2.1), and outlier clipping (Target variable y pre-fitting, Section 1.5).
- **Stationarity:** Explicit stationarity tests (e.g., ADF) and differencing were *not* performed.
    - **Justification:** The chosen model, Prophet, is designed to handle non-stationary time series directly by modeling trend and seasonalities explicitly. Forcing stationarity is generally unnecessary for Prophet.

These steps structured the data for Prophet, handled cyclical variables, generated temporal features strategically, and made deliberate choices regarding model complexity versus operational ease.

---

3. Model Selection & Justification

3.1 Primary Model Selection: Facebook Prophet

The primary forecasting model chosen for this project is Facebook Prophet. While establishing baseline models (e.g., naive forecast) is best practice for context, this implementation focused directly on Prophet due to its suitability for the anticipated characteristics of the Harveston climate data.

3.2 Justification for Prophet

Prophet was selected for several key reasons:

- **Handles Multiple Seasonalities:** Effectively models complex recurring patterns (yearly, weekly, monthly, quarterly) common in climate data using Fourier series, which were explicitly tuned.
- **Adapts to Trend Changes:** Automatically detects and adjusts to shifts in underlying trends (changepoints), suitable for non-stationary environmental data. Key trend parameters (changepoint_prior_scale, n_changepoints) were optimized.
- **Robustness:** Relatively resilient to missing data and outliers (further aided by preprocessing steps like clipping).
- **Handles Non-Stationarity:** Directly models trend and seasonality, eliminating the need for manual differencing often required by models like ARIMA.
- **Ease of Use & Scalability:** Offers an intuitive API and scales reasonably well for the per-kingdom modeling approach used.
-

3.3 Modeling Strategy: Per-Kingdom Forecasting
Separate Prophet models were trained for each unique kingdom.

- **Rationale:** Assumes climate patterns may differ geographically across Harveston. This allows each model to learn localized trends and seasonalities, aiming for higher accuracy than a single global model.

3.4 Hyperparameter Optimization: Optuna
Optuna, a Bayesian optimization framework, was used to automatically tune hyperparameters for each kingdom-target model, minimizing Mean Absolute Error (MAE) on a validation set.

- **Parameters Tuned:** Focused on Prophet's core components controlling trend flexibility (changepoint_prior_scale, n_changepoints), seasonality strength (seasonality_prior_scale), and seasonality complexity (*_fourier_order for yearly, weekly, monthly, quarterly). Notably, prior scales for potential regressors (lags, rolling means, others) were *also tuned during optimization*, even though regressors were excluded from the *final* prediction models (as discussed in Section 2.3).
- **Justification:** Automates the search for optimal parameter configurations, improving model performance and robustness compared to manual tuning (N_OPTUNA_TRIALS = 30).

3.5 Time Series Validation
A chronological train-validation split was used for optimization:

- **Method:** 80% of each kingdom's historical data for training, the remaining 20% for validation (MIN_VALIDATION_POINTS check applied).
- **Metric:** MAE on the validation set was minimized by Optuna.
- **Justification:** Mimics real-world forecasting, prevents data leakage, and provides a reliable estimate of generalization performance.

In summary, Prophet was chosen for its strengths in handling complex time series data. A per-kingdom approach targeted localized patterns, with performance systematically enhanced via Optuna-driven hyperparameter tuning based on a robust time series validation strategy.

4. Performance Evaluation & Error Analysis

4.1 Evaluation Metric: Mean Absolute Error (MAE)

Model performance during hyperparameter optimization was primarily evaluated using Mean Absolute Error (MAE).

- **Justification:** MAE provides an easily interpretable measure of average prediction error in the target variable's original units (e.g., °C, mm). It is relatively robust to outliers and served as the direct optimization objective for Optuna.

4.2 Model Performance Comparison (Based on Validation MAEs)

The best validation MAE achieved via Optuna for each kingdom-target combination (summarized by the script's output) serves as the basis for performance comparison:

- **Cross-Target:** Comparing average MAEs per target variable indicates relative predictability (e.g., temperature might be easier to predict than rainfall).
- **Cross-Kingdom:** Comparing average MAEs per kingdom can highlight regions with inherently more complex or noisy climate patterns.
- **Overall:** The average MAE across all models gives a high-level view of typical error magnitude.
- **Scope:** This comparison is *internal* among the optimized Prophet models. Performance relative to baseline models or alternative algorithms (e.g., ARIMA) was not assessed in this workflow.

4.3 Residual Analysis (Not Performed)

Standard diagnostic checks on model residuals (analysis of autocorrelation, normality, constant variance) were *not* conducted.

- **Implication:** This limits deeper insights into model goodness-of-fit. Patterns remaining in residuals would suggest uncaptured systematic information (e.g., complex seasonality, autocorrelation), indicating potential model deficiencies. This is a limitation based on standard time series practice.

4.4 Limitations and Areas for Improvement

The current methodology has several limitations offering avenues for future refinement:

- **Modeling Choices:**
  - *No Final Regressors:* Excluding lags, rolling means, and other variables from the *final* model simplifies prediction but may reduce accuracy by ignoring potentially valuable short-term predictors.
  - *No Baseline Comparison:* Lack of comparison against simple baselines makes it harder to quantify the value added by the Prophet+Optuna approach.
  - *No Residual Diagnostics:* As noted above, prevents thorough model validation.
- **Data Handling:**

    - *NaN Removal:* Dropping rows with NaNs (especially post-feature engineering) reduces training data. Imputation could be explored.
    - *Final fillna(0):* Filling remaining prediction NaNs with 0 is potentially unrealistic for some variables (e.g., Temperature, Wind Speed) and arbitrary for Wind Direction. More sophisticated imputation is recommended.
- **Model Assumptions & Scope:**
    - *Per-Kingdom Independence:* Assumes no interaction or shared patterns between kingdoms; hierarchical models might offer improvements.
    - *Data Representativeness:* Relies on the assumption that historical data is accurate and unbiased, which may not hold true (sensor issues, recording changes).
    - *Optimization/Validation:* The number of Optuna trials might be limited; a single train-validation split is less robust than time series cross-validation. Fixed outlier clipping percentage could also be refined.

Addressing these points, especially residual analysis, refined NaN handling, and experimenting with regressors, are key future steps.

5. Interpretability & Business Insights

5.1 Application of Predictions in Harveston Agriculture

The primary value of this project lies in translating the five predicted variables (Avg Temp, Radiation, Rain Amount, Wind Speed, Wind Direction) into actionable agricultural intelligence for Harveston. Accurate, localized (per-kingdom) forecasts enable a shift from reactive responses to proactive management, directly impacting:

- **Planting/Harvesting Optimization:** Using Temperature forecasts for optimal timing based on crop needs and Rainfall predictions for soil moisture assessment and safe harvesting conditions.
- **Resource Management:** Leveraging Rainfall and Radiation forecasts to schedule irrigation efficiently, conserving water.
- **Risk Mitigation:** Preparing for extremes using Rainfall forecasts (flooding/drought), Wind Speed predictions (crop damage, spray drift control), and Temperature forecasts (heat/cold stress).
- **Crop Health:** Using combined weather patterns to anticipate periods of higher pest or disease risk, informing preventative actions.

These data-driven insights empower farmers to improve yields, manage resources effectively, and build resilience against Harveston's evolving climate.

5.2 Suggestions for Deployment and Strategy Improvement

To maximize the practical utility and long-term success of the forecasting system, consider these enhancements:

- **Operationalization & Access:** Implement automated model retraining with new data and develop user-friendly interfaces (reports, dashboards, alerts) for easy access to forecasts.
- **Communicate Uncertainty:** Provide prediction intervals (available from Prophet) alongside point forecasts to inform risk assessment and decision confidence.
- **Expand Forecast Scope:** Model additional relevant variables like Min/Max Temperature, Humidity, or Evapotranspiration if data permits.
- **User Feedback:** Establish mechanisms for farmers to provide feedback on forecast accuracy and usability, guiding future improvements.
- **Advanced Modeling:**
    - Explore *ensemble methods* (combining Prophet with other models like ARIMA, XGBoost) for potentially higher accuracy and robustness.
    - Re-evaluate including *external regressors* (lags, other weather variables) in the final model, balancing accuracy gains against added complexity.
    - Consider moving towards *probabilistic forecasting* for richer risk management capabilities.

Implementing these suggestions would evolve the models into a comprehensive, adaptive climate advisory service for Harveston.

6. Innovation & Technical Depth

This project demonstrates technical depth through several key strategies tailored to the Harveston climate prediction challenge:

1. **Localized Modeling (Per-Kingdom):** Implemented separate Prophet models for each kingdom and target variable. This granular approach addresses potential geographic climate variations across Harveston, allowing each model to specialize in local patterns for potentially higher accuracy than a global model.
2. **Advanced Time Series Modeling (Prophet):** Utilized Prophet, a modern library adept at handling multiple seasonalities and trend changes common in environmental data. Technical depth was shown by explicitly configuring and tuning multiple seasonal components (yearly, weekly, monthly, quarterly) via Fourier orders to capture complex cyclical patterns.
3. **Sophisticated Hyperparameter Optimization (Optuna):** Employed Optuna for automated Bayesian optimization, systematically tuning key Prophet parameters (changepoint/seasonality priors, Fourier orders) and even group-specific prior scales for regressors *during* the optimization phase. This advanced method surpasses manual tuning for finding robust model configurations.
4. **Strategic Feature Utilization:** Engineered temporal features (lags, rolling means) and other variables were used as regressors *only during Optuna optimization*. While excluded from the simpler *final* models, this nuanced strategy aimed to leverage richer context during tuning to potentially refine Prophet's core trend/seasonality parameters more effectively.
5. **Correct Cyclical Feature Handling:** Properly addressed the cyclical nature of Wind_Direction by transforming it into sin and cos components for modeling, ensuring accurate representation of angular proximity. These components were modeled individually and reconstructed post-prediction.
6. **Robust Preprocessing:** Included targeted steps like automated Kelvin-to-Celsius conversion and percentile-based outlier clipping to enhance data quality and model stability.

While deep learning or ensembles were not used here, the combination of localized modeling, advanced Prophet configuration, sophisticated optimization with strategic feature use during tuning, and rigorous handling of data nuances represents a technically sound and tailored approach.