Multi-Target Environmental Forecasting for Harveston

By TechSpark- Data_Crunch_188
University of Moratuwa

1. Problem Understanding & Dataset Analysis

- **Forecasting Objective & Expected Outcomes:**
    - The primary objective is to accurately forecast five key environmental variables crucial for agriculture in the fictional region of Harveston: Average Temperature (°C), Radiation (W/m²), Rain Amount (mm), Wind Speed (km/h), and Wind Direction (°).
    - Forecasts are required for specific future dates indicated in the test.csv dataset, covering various geographical sub-regions denoted as kingdoms.
    - The expected outcome is a predictive model that minimizes forecasting error, thereby providing actionable insights for farmers to optimize planting, irrigation, harvesting, and resource management decisions. Success is measured by the average Symmetric Mean Absolute Percentage Error (SMAPE) across the five target variables on a hidden test set.
- **Key Findings from Data Analytics:**
    - Temporal Nature: The data is time series-based, recorded daily, necessitating models that respect temporal dependencies.
    - Spatial Component: Data originates from multiple kingdoms, suggesting potential regional variations in weather patterns. Latitude and longitude were provided in the training set but are missing in the test set for these kingdoms.
    - Anonymized Time: The 'Year' column contained values from 1 to 9, indicating anonymized or indexed years rather than actual calendar years. This required a placeholder approach for creating datetime features.
    - Inconsistent Units: Initial analysis of Avg_Temperature and Avg_Feels_Like_Temperature revealed values significantly above typical Celsius ranges, suggesting some data points were recorded in Kelvin, requiring unit standardization.
    - Target Characteristics: Visualizations (or preliminary analysis) suggested likely seasonality in Temperature and Radiation. Rain Amount distribution

was expected to be highly skewed with many zero values. Wind Direction is a circular variable (0° ≈ 360°).

- Feature Availability: Key predictive features like latitude, longitude, and several weather measurements (Avg_Feels_Like_Temperature, Evapotranspiration, etc.) were present in train.csv but absent in test.csv, necessitating careful feature engineering based on historical data (lags, rolling windows).

- **Preprocessing Justification:**
  - Datetime Assembly: Combined 'Year', 'Month', 'Day' into a datetime object. Due to anonymized years, a base year (e.g., 2000) was added to the index year (Year-1) to create valid datetime objects for feature extraction (e.g., dayofyear, dayofweek) and sorting, while acknowledging these are not actual calendar dates.
  - Temperature Unit Conversion: Implemented a check for temperature values > 70 (a plausible threshold for Kelvin). Identified Kelvin values were converted to Celsius using the standard formula (K-273.15) to ensure consistent units for the target variable, Avg_Temperature, and related features.
  - Latitude/Longitude Handling: As these were missing in the test set, a mapping was created from the training data (kingdom -> average latitude/longitude). This map was used to add these spatial features to the test set based on the kingdom.
  - Missing Value Handling: No explicit imputation was performed in the final model structure. LightGBM has built-in capabilities to handle missing values (NaNs), which arise naturally from lag/rolling feature creation at the beginning ofa time series or due to NaT date conversions.
  - Outlier Handling: No specific outlier removal steps were implemented. Tree-based models like LightGBM are generally less sensitive to outliers than linear models. Extreme values were assumed to be potentially valid weather events.
  - Scaling/Normalization: Not applied, as tree-based models like LightGBM

are invariant to monotonic transformations of features and do not require features to be on the same scale.

2. Feature Engineering & Data Preparation

- **Feature Creation Techniques:**
    - **Basic Time Features:** Extracted standard time-based features: dayofyear, dayofweek, month, weekofyear. Basic cyclical features (month_sin/cos, dayofyear_sin/cos) were generated to capture simple seasonality without relying on actual calendar dates.
    - **Lag Features:** Created lagged versions of target variables and key predictors (from FEATURE_COLS_FOR_ENGINEERING) for periods [1, 3, 7, 14, 21, 30, 60] days. These capture autoregressive dependencies (how yesterday's weather influences today's). Lags were computed per kingdom to respect geographical boundaries.
    - **Rolling Window Features:** Calculated rolling window statistics (mean, standard deviation, median, min, max) over various time windows ([3, 7, 14, 30, 60] days) for the same set of variables. These capture recent trends, volatility, and central tendencies, again computed per kingdom and using shift(1) to prevent data leakage.
    - **Lag Difference Features:** Computed differences between lags (e.g., lag_1 - lag_7, lag_7 - lag_30) to capture rates of change or deviations from longer-term patterns.
    - **Spatial Features:** Used the mapped latitude and longitude as numerical features. The kingdom categorical variable was label-encoded for use in LightGBM.
    - **Wind Direction Transformation:** Transformed the circular Wind_Direction target into two linear components, Wind_Direction_sin and Wind_Direction_cos, using sin(rad(θ)) and cos(rad(θ)). Models were trained to predict these components separately.
- **Feature Selection Justification:**
    - The feature set was primarily built based on time series forecasting best practices and domain knowledge (weather patterns depend on recent history and exhibit seasonality).

- Lag and rolling window features are standard techniques to provide the model with historical context.
- Time features capture cyclical patterns.
- Spatial features (latitude, longitude, kingdom_encoded) allow the model to learn region-specific behaviours.
- Features were generated for all target variables and key related predictors available in the training data to maximize the information available to the models.
- Features directly excluded were identifiers (ID), raw date components (Year, Month, Day), the original kingdom string (replaced by encoding), helper flags (is_test), and the intermediate datetime object. Original non-target predictors from the training set were also excluded from the final feature list if only their lagged/rolling versions were deemed necessary. *Note: While Fourier features were explored, they were excluded from the final submitted model as preliminary tests suggested they did not improve the overall score for this specific configuration.*

- **Transformations:**
  - **Circularity:** The sin/cos transformation for Wind_Direction was essential to allow the linear regression-based LightGBM model to handle the circular nature of degrees (0° being close to 360°).
  - **Log Transformations:** While considered, particularly for Rain_Amount due to its skewness and zero-inflation, the submitted model, achieving the ~43 score, did *not* use log transformations for any target variable. Standard regression_l1 was used on the original scale. *(Further work identified Log1p as a strong candidate for future improvement)*.
  - **Stationarity/Differencing:** Explicit differencing was not required. Tree-based models like LightGBM can implicitly handle trends and non-stationarity through their splitting mechanism. Furthermore, lag and rolling window features provide information about changes over time.
  - **Normalization/Scaling:** Not applied, as previously justified for LightGBM.

3. Model Selection & Justification

- **Baseline & Advanced Models:**
    - The primary modeling approach utilized an advanced technique: **LightGBM (Light Gradient Boosting Machine)**.
    - While baseline models like simple persistence (predicting the previous day's value) or ARIMA could be considered, the complexity of predicting five correlated weather variables across multiple locations suggested a more powerful machine learning approach was necessary from the outset. Other advanced techniques like XGBoost, CatBoost, or LSTMs were potential alternatives.
- **Model Choice Justification (LightGBM):**
    - **Performance:** LightGBM is known for its state-of-the-art performance on tabular data, often achieving high accuracy in forecasting competitions.
    - **Speed & Efficiency:** It's generally faster and uses less memory than other Gradient Boosting implementations like XGBoost, which is beneficial given the dataset size and feature count.
    - **Handling Missing Values:** Its built-in ability to handle NaNs simplifies preprocessing, especially when dealing with lags and rolling windows that naturally create missing values.
    - **Scalability:** Capable of handling large datasets efficiently.
    - **Separate Models per Target:** A strategy of training one independent LightGBM model for each of the five(+2 for sin/cos) target variables was chosen. This simplifies the modeling process, allows for target-specific feature importance analysis and hyperparameter tuning, and avoids potential issues with multi-output regression loss functions being dominated by targets with different scales or error distributions.
- **Hyperparameter Optimization:**
    - A systematic hyperparameter optimization strategy was employed using the **Optuna** framework, which utilizes Bayesian optimization techniques

(specifically TPE - Tree-structured Parzen Estimator) to efficiently search the parameter space.

- Key hyperparameters tuned for each LightGBM model included n_estimators, learning_rate, num_leaves, regularization terms (lambda_l1, lambda_l2), feature/bagging fractions (feature_fraction, bagging_fraction, bagging_freq), and min_child_samples.
- The optimization process aimed to minimize the validation SMAPE score (using MAE - regression_l1 - as a practical, built-in proxy objective during tuning) averaged across time series cross-validation folds.

- **Time Series Validation Approach:**
  - A robust time-series cross-validation approach was implemented using **sklearn.model_selection.TimeSeriesSplit with n_splits=5**.
  - This method ensures that the validation data always comes chronologically after the training data within each fold, simulating a realistic forecasting scenario and preventing data leakage from the future into the past.
  - For each fold, the model was trained on the training portion and evaluated on the subsequent validation portion. Early stopping (based on MAE on the validation fold) was used within each fold's training process to prevent overfitting and determine a suitable number of boosting rounds implicitly. The final validation score for a set of hyperparameters was the average SMAPE across the 5 folds.

4. Performance Evaluation & Error Analysis

- **Evaluation Metrics:**
    - The primary metric, mandated by the competition, is the **average Symmetric Mean Absolute Percentage Error (SMAPE)** across the five target variables (Avg_Temperature, Radiation, Rain_Amount, Wind_Speed, Wind_Direction).
    - SMAPE = mean(2 * |Prediction - Actual| / (|Actual| + |Prediction|)) * 100. It provides a percentage error relative to the magnitude of both actual and predicted values, treating over- and under-prediction errors symmetrically. It is bounded between 0% and 200%.
    - During hyperparameter tuning and early stopping, **Mean Absolute Error (MAE)** (LightGBM objective regression_l1) was used as a proxy metric. MAE is less sensitive to outliers than Root Mean Squared Error (RMSE) and often correlates reasonably well with SMAPE minimization goals, while being a standard built-in objective.
- **Model Performance Comparison:**
    - The final LightGBM model, incorporating the extended feature set, sin/cos wind direction handling, and tuned hyperparameters, achieved a **Public Leaderboard SMAPE score of approximately 43.17**. This was an improvement over an earlier baseline using default parameters (~44.19).
    - Cross-validation results during tuning indicated varying performance across targets:
        - Avg_Temperature SMAPE: ~1.11 (Excellent)
        - Radiation SMAPE: ~6.73 (Excellent)
        - Rain_Amount SMAPE: ~74.14 (Poor)
        - Wind_Speed SMAPE: ~9.95 (Excellent)
        - Wind_Direction_sin SMAPE: ~27.12 (Okay)
        - Wind_Direction_cos SMAPE: ~54.0 (Poor)
    - The tuned LightGBM approach was selected as the best model due to its demonstrated improvement on the leaderboard and strong performance

on several individual targets during validation, despite known challenges with Rain Amount and Wind Direction Cosine.

- **Residual Analysis:** *(Note: Formal residual analysis plots were not generated as part of the final script, but this section describes potential findings/interpretations)*.
    - A full residual analysis would involve examining the errors (Actual - Prediction) for each target model on the cross-validation folds.
    - **Autocorrelation:** Plotting residuals over time might reveal autocorrelation (patterns in errors), suggesting the model isn't capturing all temporal dependencies. This could indicate the need for more sophisticated lag features or different model types (like ARIMA/LSTM components).
    - **Normality:** Histograms or Q-Q plots of residuals can be used to check for normality. Deviations might suggest issues with model assumptions or the need for target transformations (especially for Rain Amount).
    - **Heteroscedasticity:** Plotting residuals against predicted values could reveal heteroscedasticity (variance of errors changing with the prediction level). This is common in time series and might require variance-stabilizing transformations or models that explicitly handle changing variance. Given the high SMAPE for Rain Amount, its residuals are expected to show significant non-normality and likely heteroscedasticity.
- **Model Limitations, Biases, and Improvement Areas:**
    - **Limitation (Rain Amount):** The primary limitation is the poor performance on Rain_Amount, evidenced by the high validation SMAPE (~74). The standard regression approach struggles with the zero-inflated and highly skewed nature of rainfall data.
    - **Limitation (Wind Direction):** The model predicts the cos component of wind direction significantly worse than the sin component, suggesting difficulty capturing North-South wind patterns or features insufficient for this aspect.

- **Limitation (Anonymized Time):** The inability to use actual calendar dates prevents the incorporation of specific holiday effects or more precise long-term seasonal modeling tied to specific years.
- **Potential Bias:** The model might be biased towards predicting average conditions and struggle with extreme weather events due to their rarity in the training data.
- **Improvement Areas:** Focus on Rain Amount (Log1p transform, Tweedie objective, Two-stage model), enhanced feature engineering (interactions, Fourier terms if they prove beneficial upon re-evaluation, target encoding), exploring alternative models (XGBoost, CatBoost), and ensembling predictions from multiple strong models.

5. Interpretability & Business Insights

- **Real-world Application:**
    - The forecasting results provide direct value to farmers in Harveston by enabling data-driven decision-making.
    - **Irrigation Scheduling:** Accurate Rain_Amount and Avg_Temperature forecasts help optimize water usage, conserving resources and preventing crop stress. Evapotranspiration forecasts (if available or derived) would further enhance this.
    - **Planting/Harvesting Timing:** Avg_Temperature, Radiation, and Rain_Amount forecasts inform optimal windows for planting seeds and harvesting crops to maximize yield and quality.
    - **Pest/Disease Management:** Temperature and humidity (derivable from temperature and rain) forecasts can help predict outbreaks of certain pests or diseases, allowing for proactive treatment.
    - **Resource Allocation:** Wind_Speed forecasts can inform decisions about protecting young plants or infrastructure. Radiation forecasts help estimate potential solar energy generation for farm operations. Overall forecasts aid in planning labor and equipment needs.
    - **Risk Mitigation:** Advance warning of potentially damaging conditions (high winds, heavy rain, frost risk based on temperature) allows farmers to take preventative measures.
- **Suggested Improvements (Strategy & Deployment):**
    - **Forecasting Strategy:**
        - Implement target transformations (log1p) or specialized objectives (Tweedie) for Rain_Amount.
        - Develop and integrate ensemble models (averaging predictions from LGBM, XGBoost, CatBoost) for improved robustness.
        - Incorporate more granular location data or static geographical features (elevation, distance to coast) if they could be obtained or inferred for each kingdom.

        - Explore models that explicitly handle spatial-temporal dependencies if kingdom interactions are suspected.
    - **Model Deployment:**
        - **Cloud-Based Pipeline:** Deploy the preprocessing, feature engineering, and prediction pipeline on a cloud platform (AWS, GCP, Azure) for automated daily forecasts.
        - **API Access:** Provide forecasts via a simple API that can be integrated into farm management software or dashboards used by farmers.
        - **Regular Retraining:** Implement a schedule for automatically retraining the models on new incoming data (e.g., weekly or monthly) to adapt to changing climate patterns and maintain forecast accuracy.
        - **Monitoring:** Continuously monitor model performance against actual observed weather data and trigger alerts or retraining if performance degrades significantly.
        - **User Interface:** Develop a user-friendly dashboard displaying forecasts, confidence intervals (if generated), and historical trends for easy interpretation by farmers.

6. Innovation & Technical Depth

- **Novel Approaches & Advanced Techniques:**
    - **Systematic Hyperparameter Optimization:** Utilized Optuna, a modern Bayesian optimization framework, coupled with rigorous TimeSeriesSplit cross-validation to find near-optimal hyperparameters for each target-specific LightGBM model, moving beyond simple grid search or manual tuning.
    - **Handling Circular Features:** Addressed the inherent challenge of predicting Wind Direction by decomposing it into sin and cos components, training separate models, and recombining them using arctan2, allowing a standard regression model to effectively learn the circular pattern.
    - **Extensive Time Series Feature Engineering:** Implemented a comprehensive set of lag features (up to 60 days), rolling window features (multiple window sizes and statistics: mean, std, median, min, max), and lag difference features, tailored per kingdom, to capture complex temporal dependencies beyond simple autoregression.
    - **Anonymized Time Handling:** Devised a strategy using placeholder datetime objects based on an arbitrary base year to enable the extraction of meaningful relative time features (dayofyear, weekofyear, cyclical sin/cos) despite the lack of absolute calendar dates.
- **Unique Techniques for Accuracy/Efficiency:**
    - The **combination** of the above techniques represents a tailored approach to this specific problem's constraints (anonymized time, circular target, multi-target forecasting, missing test set features).
    - The specific choice of extended lags and rolling window statistics (including median, min, max) aimed to provide a richer historical context to the LightGBM models compared to using only means or shorter windows.
    - Employing separate, individually tuned models for each target variable allowed for specialized handling (like the sin/cos transform for wind direction components) and prevented negative interference between

targets with vastly different scales or distributions (like Temperature vs. Rain Amount) within a single multi-output model.

- Using MAE (regression_l1) as the objective and early stopping metric provided robustness against outliers while serving as a reasonable proxy for the SMAPE evaluation metric during the computationally intensive tuning phase.

7. Conclusion

- **Summary & Key Findings:**
    - This project successfully developed a multi-target time series forecasting system to predict five key environmental variables for the Harveston region using LightGBM.
    - The best-performing model utilized extensive feature engineering, including extended lags (up to 60 days), multi-statistic rolling windows (up to 60 days), lag differences, label-encoded kingdom information, and mapped latitude/longitude.
    - A key technique involved transforming the circular Wind Direction variable into sin/cos components for effective modeling.
    - Systematic hyperparameter optimization via Optuna with TimeSeriesSplit cross-validation was employed to tune parameters for each target model individually.
    - The final model achieved a Public Leaderboard score (SMAPE) of **~43.17**.
- **Challenges & Potential Future Improvements:**
    - **Primary Challenge:** The main difficulty lies in accurately forecasting Rain_Amount due to its sparse and skewed distribution, which significantly impacts the overall average SMAPE score. The model for the Wind_Direction_cos component also showed relatively poor validation performance.
    - **Anonymized Data:** The anonymized 'Year' column limited the use of calendar-specific features.
    - **Future Improvements:**
        - Implement target transformations (e.g., log1p) or specialized loss functions (e.g., Tweedie, requiring retuning) specifically for Rain_Amount.
        - Explore two-stage models (classification + regression) for Rain_Amount.

- Conduct further feature engineering, focusing on interactions and potentially incorporating external data sources if permissible.
- Evaluate alternative gradient boosting models like XGBoost or CatBoost, particularly CatBoost for its handling of categorical features like kingdom.
- Develop ensemble models by averaging predictions from LightGBM, XGBoost, and CatBoost to enhance robustness and potentially improve accuracy.
- Perform detailed residual analysis to better diagnose model weaknesses for specific targets.