Weather Forecasting for Multiple Kingdoms

By [Synexis-Data_Crunch_151]
[University of Moratuwa]

1. Problem Understanding & Dataset Analysis

The objective is to forecast five weather variables Average Temperature (°C), Radiation, Rain Amount, Wind Speed, and Wind Direction for multiple kingdoms over a future period. The expected outcome is a submission file with predictions aligned with test.csv (4530 rows), minimizing the symmetric Mean Absolute Percentage Error (sMAPE)

Key Insights from Data Analysis

- Dataset Structure: train.csv contains 84,960 rows across 17 columns of temporal (Year, Month, Day), spatial (latitude, longitude, kingdom), and target variables. test.csv contains 4,530 rows with ID, Year, Month, Day, and kingdom only.
- Kingdoms: 30 unique kingdoms (e.g., Arcadia, Atlantis, Winterfell), each with ~2,832 training rows (with uniform distribution assumption), and ~151 test rows.
- Seasonal Patterns: Seasonality on an annual and weekly basis identified in exploratory analysis (plotted in Prophet's trend/seasonality components).
- Missing Values: Minimal missing values in train_df, resolved using forward and backward fill by kingdom.
- Temperature Scale: Avg_Temperature and Avg_Feels_Like_Temperature in Kelvin (above 100), rescaled to Celsius.

Preprocessing Description

- Date Conversion: Year, Month, Day consolidated into a date column (base year 2015 appended for two-digit years) for time series modeling.
- Temperature Conversion: Scaled values >100 by subtracting 273.15 to change Kelvin to Celsius, consistent with real-world interpretation.
- Missing Values: Imputed using forward-fill (ffill) followed by backward-fill (bfill) within each kingdom to maintain temporal continuity.

- Sorting: Sorted train_df by kingdom and date, and test_df by date, to maintain chronological order for Prophet.

2. Feature Engineering & Data Preparation

Feature Creation Techniques

- Temporal Features: Added month and day_of_year to capture seasonality (in code3.ipynb).

- Lags and Moving Averages: For each target in train_df, computed lag-1 (shift(1)) and 7-day moving averages (rolling(7)), grouped by kingdom, to leverage recent history and suppress noise.

- Wind Direction Encoding: Transformed Wind_Direction to wind_dir_sin and wind_dir_cos based on sine/cosine of radians for cyclic continuity (in code3.ipynb).

- Test Set Features: Rolled over last values of lags and moving averages of train_df over to test_df by kingdom, to preserve feature presence.

Feature Selection Rationale

- Prophet Focus: In final submission, used only Prophet's internal seasonality (annual, weekly) without external regressors because addition of lags/features in submission2.ipynb enhanced sMAPE (40%) compared to the base (38%).

- Impact: Simplicity kept generalizability intact, avoiding overfitting observed with complex features.

Data Stationarity Transformations

- No Explicit Transformations: Prophet handles non-stationarity using seasonality and changepoint detection, so log transformations or differencing were not required.

- Filling: Preserved continuity using ffill/bfill, stabilizing series naturally.

3.Model Selection & Rationale

Model Evaluation

- Baseline: Each Prophet model per kingdom (changepoint_prior_scale=0.05, yearly seasonality) earned a 38% sMAPE.

- Advanced Attempt: more features (lags, moving averages) and Prophet, but hit 40%, indicating overfitting or misalignment.

- Final Model: used a tuned Prophet model per kingdom, balancing flexibility and regularization.

Model Choice Justification

- Prophet: Chosen for its ability to model seasonality and trends with minimal feature engineering, suitable for multi-kingdom daily weather data with clear yearly cycles.

- Dataset Fit: 30 kingdoms with ~8 years of daily data (2015-2023) take advantage of Prophet's capacity for long-term seasonal trends.

Hyperparameter Optimization

- Manual Tuning: changepoint_prior_scale=0.05 (baseline flexibility) and seasonality_prior_scale=10.0 (moderate strength of seasonality) adopted from baseline following trial of stronger values (e.g., 0.01) increased errors.

- Seasonality: Added weekly_seasonality=True to submission2.ipynb to capture short-term trends lost in baseline.

Validation Approach

- Time-Based Split: In early tries (not shown), utilized last 20% of train_df by each kingdom as a validation, but final submission used Kaggle feedback due to timing and one outstanding submission.

- Assumption: Sequential kingdom split in test_df (151 rows by each kingdom, adjusted for last).

4. Performance Evaluation & Error Analysis

Performance Evaluation & Error Analysis

Evaluation Metrics

- sMAPE: Used as per Kaggle requirements, a symmetric percentage error measure, zero-robust to targets like Rain_Amount.

- Baseline: 38% sMAPE (single Prophet).

- Feature-Heavy: 40% sMAPE poorer due to feature noise.

Model Comparison

- Best Model: Prophet (simplified, weekly seasonality) likely superior to 40% and could surpass 38%, granted return to baseline proficiency.

- Rationale: Overcompleteness of code3.ipynb out of synch with test set, while simpleness plays off Prophet strength.

Residual Analysis

- Not Performed: Competition form limited by absence of test set ground truth, but Prophet diagnostics suggest seasonality explains bulk variance.

- Assumptions: Residuals likely uncorrelated for temperature/radiation, but Rain_Amount (skewed, zero-heavy) and Wind_Direction (cyclic) likely biased.

Limitations

- Kingdom Misalignment: Equal test rows assumed by kingdom (4530 ÷ 30 = 151), but uneven spread may skew forecasts.
- Overfitting Risk: Feature-rich models overfit training patterns not present in test_df.

5.Interpretability & Business Insights

Interpretability & Business Insights

Real-World Application

- Weather Forecasting: Forecasting enables kingdoms to plan agriculture (temperature, rain), energy (radiation), and infrastructure (wind).
- Example: High Rain_Amount forecasts enable flood preparation; stable Wind_Direction aids in wind farm placement.

Forecasting Strategy Improvements

- Ensemble: Combine Prophet with XGBoost or LSTM for hybrid accuracy (not done due to time).
- Deployment: Dynamic changepoints with live updates of new data.

Innovation & Technical Depth

Novel Approaches

- Weekly Seasonality: Added to submission1.ipynb to address short-term weather cycles, enhancing baseline.
- Feature Engineering: Lags, moving averages, and wind encoding in code3.ipynb tried out advanced inputs, though not perfect.

Unique Techniques

- Kingdom-Specific Models: Trained separate Prophet instances for each kingdom, preserving local patterns.
- Fallback Simplicity: Retailed to bare features past 40% sMAPE, a strategic change to maximize final submission.

6. Innovation & Technical Depth

Innovation & Technical Depth

Novel Approaches

- Weekly Seasonality: Added to submission1.ipynb to capture short-term weather patterns, enhancing baseline.
- Feature Engineering: Lags, moving averages, and wind encoding in code3.ipynb explored advanced inputs, though not optimal.

Unique Techniques

- Kingdom-Specific Models: Trained separate Prophet instances for each kingdom, preserving local patterns.
- Fallback Simplicity: Deflated to bare minimum features after 40% sMAPE, a strategic reorientation towards harnessing ultimate submission.

7.Conclusion

Key Findings

- Top Performing Model: Refined Prophet with yearly/weekly seasonality, with the goal of 10-20% sMAPE.
- Performance: Feature-rich (40%) was beaten by baseline (38%) to inform a less complex end strategy.

Challenges

- Influence of Features: Lags and moving averages affected performance, presumptively through test set mismatch.
- Submission Limit: Single attempt forced reliance on baseline thinking over extended validation.

Future Improvement

- Ensemble Models: Integrate Prophet with machine learning for potency.
- Validation: Optimize parameters prior to submission using rolling window cross-validation.
- Kingdom Alignment: Test set kingdom distribution validation to enhance slicing.