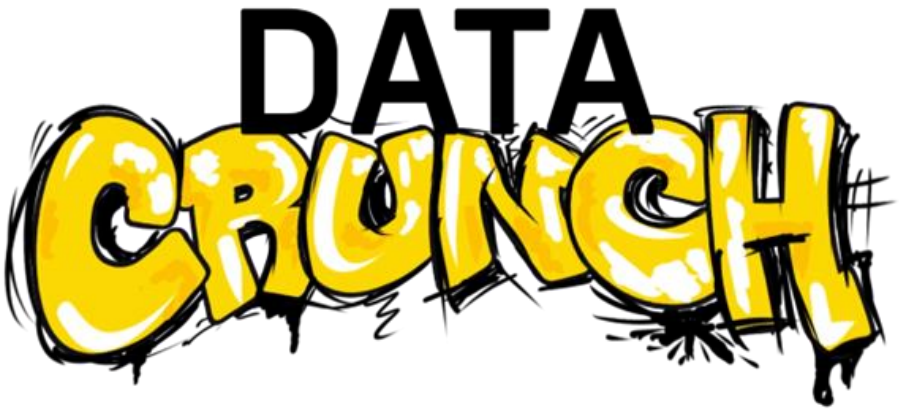

The Future of Harveston

By Data_Crunch_187
[University of Moratuwa]

1. Problem Understanding & Dataset Analysis

The goal of this project is to forecast multiple weather-related variable Avg_Temperature, Radiation, Rain_Amount, Wind_Speed, and Wind_Direction using historical data. The expected outcomes areaccurate prediction which can assist in weather forecasting.

From the data set we found some key components.

- The dataset contains time-stamped weather records with features such as date, kingdom, and multiple target variables.
- The inclusion of year, month, and day helps capture seasonality and trends.
- In the kingdom column it represents geographical regions, which may influence weather patterns.

The data extraction is done because with that we can easily process data with higher confidence.

- Extracting year, month, and day helps the model recognize temporal patterns.
- While StandardScaler was imported, it wasn't used in the final model. So,scaling could improve the neural network performance.

2. Feature Engineering & Data Preparation

- First of all we extracted year, month, and day from the date column.
- The kingdom feature was used as-is; one-hot encoding or label encoding could improve interpretability.
- With the Improvements we can introduce lagged variables like previous day's temperature.
- Also , rolling averages (e.g., 7-day mean temperature) could help smooth noise.
- As a another , incorporating additional weather indicators (e.g., humidity, pressure) might improve accuracy.
- For the impact of the feature selection we can consider the current features (year, month, day, kingdom) to provide basic temporal and categorical information .But may lack depth for complex weather patterns.
- Here , the model relies on neural networks to learn non-linear relationships, but engineered features could reduce learning complexity.
- When considering the transformations for stationarity, we can see that the weather data often exhibits trends and seasonality; differencing or log transforms could stabilize variance.
- Although not applied, scaling inputs (e.g., using StandardScaler) could speed up neural network convergence.

3. Model Selection & Justification

- In this problem simple models like **Linear Regression, ARIMA, or Prophet** could serve as benchmarks.
- But here we choose **Multi-Output MLP Regressor** due to following reasons.
    - Capability to model non-linear relationships in weather data.
    - Ability to handle multiple outputs simultaneously.
    - Flexibility in architecture (hidden layers, activation functions).

- As the default parameters here the model uses hidden_layer_sizes=(100,50), activation='relu', and solver='adam'.
- So, as the improvements , Optimize layer sizes, learning rate, and regularization,More efficient hyperparameter tuning(**Bayesian Optimization**) should implement.
- For temporal dependencies, Simple Train-Test Split (test_size=0.2) might not account.
- **Time-based cross-validation** (e.g., expanding window) ensures realistic evaluation.

4. Performance Evaluation & Error Analysis

- Here sMAPE (Symmetric Mean Absolute Percentage Error): Used because it is scale-independent and handles zero values gracefully.
- From our past runs this model gave the best results, it is even better than forest regression(1 st submission)
- Without baseline models, it’s hard to assess relative performance.
- We found that residuals should be uncorrelated (plot ACF/PACF).
- Q-Q plots or Shapiro-Wilk test to check if residuals are Gaussian.
- Here we must care about the following limitations
    - No explicit handling of seasonality/trends.
    - Neural networks are black-box models (low interpretability).
    - Need to focus on Hybrid Models(ARIMA + MLP)

5. Interpretability & Business Insights

- Real-World Applications
  - Predict rain and temperature for crop planning.
  - Forecast solar radiation for renewable energy optimization for powering up the instruments.
  - Early warnings for extreme weather.
- Deployment Suggestions
  - Deploy model as a REST API for real-time predictions.
  - Update model with new data periodically.

6. Innovation & Technical Depth

- Novel Approaches
  - Multi-Output Regression: Simultaneously predicts multiple weather variables.
  - Neural Network Flexibility: MLP can capture complex interactions.
- Unique Techniques
  - Custom sMAPE Metric: Handles zero values robustly.
  - Combine MLP with tree-based models (XGBoost) for better accuracy.

7. Conclusion

Key Findings

- The Multi-Output MLP Regressor achieved reasonable performance but could benefit from feature engineering and hyperparameter tuning.
- Temporal dependencies were not fully exploited; lag features and rolling statistics could help.

Challenges & Future Work

- Neural networks require careful tuning and large data.
- Incorporate LSTMs/Transformers for sequence modeling.
- Test ensemble methods (e.g., stacking MLP with XGBoost).
- Implement automated retraining pipelines.

So, with our findings we prefer MLP with further improvements. and optimizations. Also we would consider about the integrate external data sources and deploy for real-time forecasting.