Weather forecasting with LSTM

Julen Legido, Adrià Porta, Alfonso Castelijns

Index

Objective and Motivation

We are implementing a Long Short-Term Memory (LSTM) neural network. Inspired by the work in the Medium article by Ozdogar [1].

We aim to adapt a PyTorch-based LSTM model to forecast weather variables, in our case temperature.

We would like to compare our LSTM with different models and implement xAI techniques such as SHapley Additive exPlanations (SHAP) to measure the importance of features.

Dataset

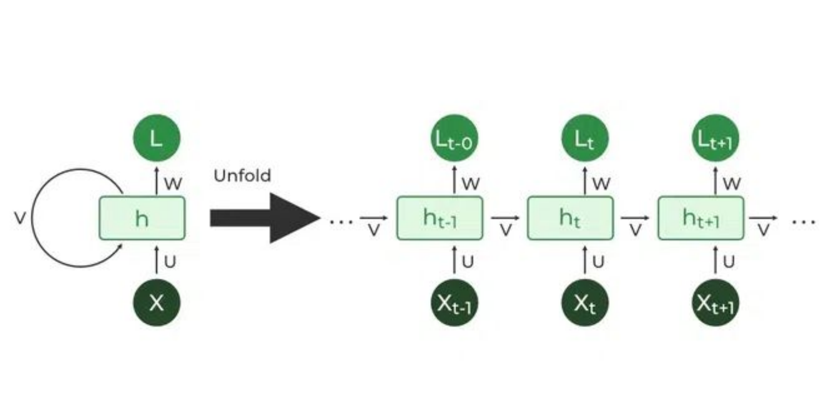
We have used a New York dataset, which contains the historical daily weather data for New York City [2] between **2023 and 2024**.

It includes features like **temperature, humidity, wind speed, sunrise/sunset, conditions**, etc.

We cleaned the data, encoded categorical features and scaled numeric features to make it suitable for the model.

Background

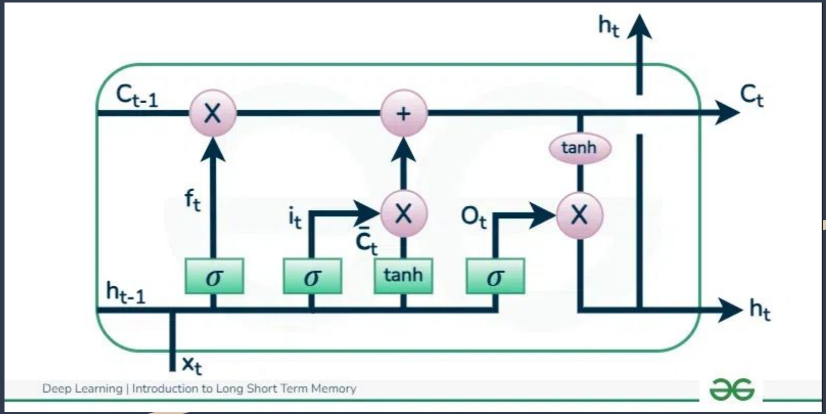
Recurrent Neural Network (RNN)



- Processes sequences by maintaining a hidden state that “remembers” past inputs
- At each time step, combines current input x_t with previous state h_t−1 to produce h_t.
- Captures short-term temporal patterns (e.g., recent weather trends)
- Can struggle with very long-term dependencies due to vanishing/exploding gradients

Background

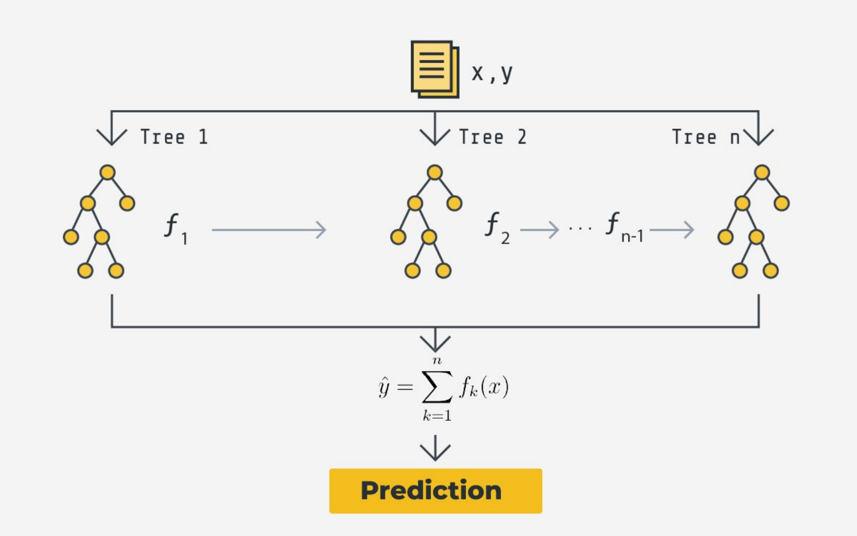
Long Short Term Memory (LSTM)



- An RNN variant with gated cells (input, forget, output)
- Mitigates vanishing/exploding gradients for long sequences
- Maintains a cell state C_t to carry forward relevant information
- Captures longer-term dependencies (e.g., seasonal weather patterns)

Background

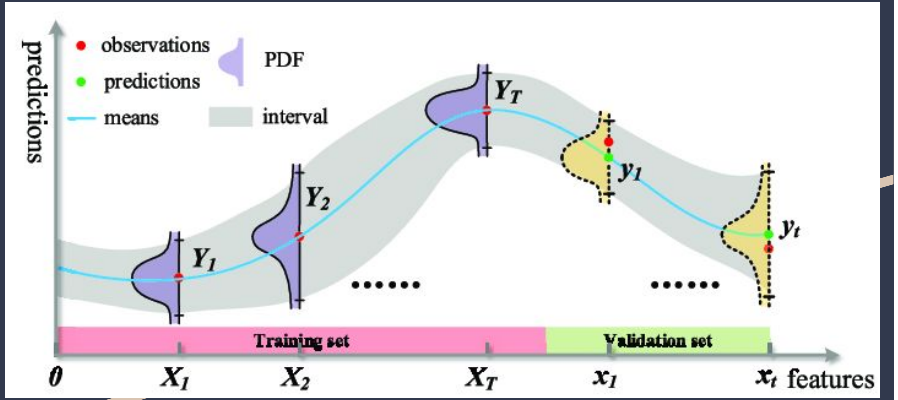
Extreme Gradient Boosting (XGBoost)



- Ensemble of decision trees trained via gradient boosting
- Each tree fits the residual errors of the current model
- Fast, scalable, and handles complex non-linear patterns
- Used as a baseline to compare against our LSTM and RNN models

Background

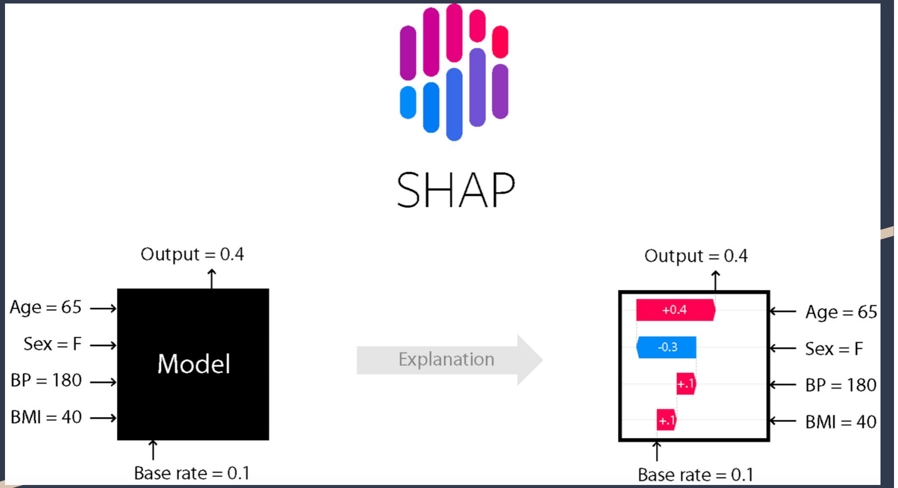
Gaussian Processes (GP)



- Probabilistic model used for regression
- Process:
  - Defines a distribution that could fit the data
  - Measures similarity between input points
  - Makes predictions by weight averaging all the distributions by how well they fit the data
- Often performs well on small datasets

Background

SHapley Additive exPlanations (SHAP)



- Game theoretic approach to explain the output of any machine learning model
- Treats each feature as a “player” contributing to the final prediction of the model
- Computes each feature’s contribution by averaging over all possible combinations of features it could be added to
- Provides consistent, locally accurate explanations for any model

Methodology

We implemented a Multivariate LSTM over a **31-day window**.

The dataset was split sequentially into **80% training and 20% testing** sets to preserve time order and prevent data leakage.

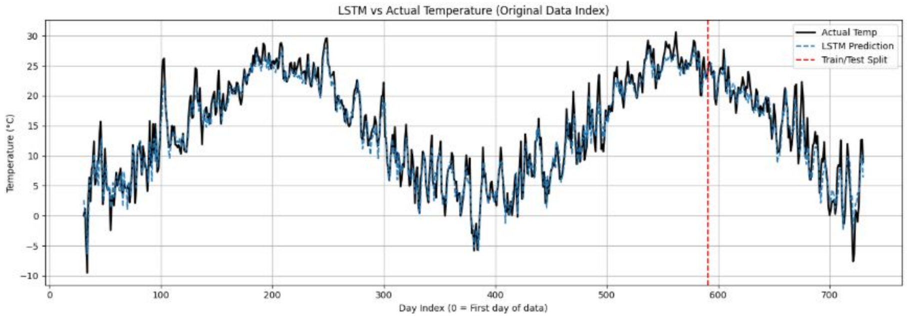
We evaluated the model's ability to learn temporal dependencies using Mean Squared Error (**MSE**), Mean Absolute Error (**MAE**) and **R^2 Score**.

As baseline models, we trained: Gaussian Processes (**GP**), Extreme Gradient Boosting (**XGBoost**) and a simple Recurrent Neural Network (**RNN**).

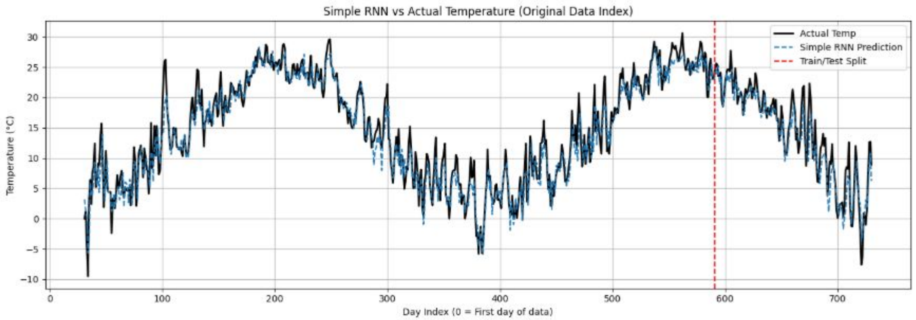
We applied **SHAP** values to quantify the contribution of each feature in the prediction process.

Results

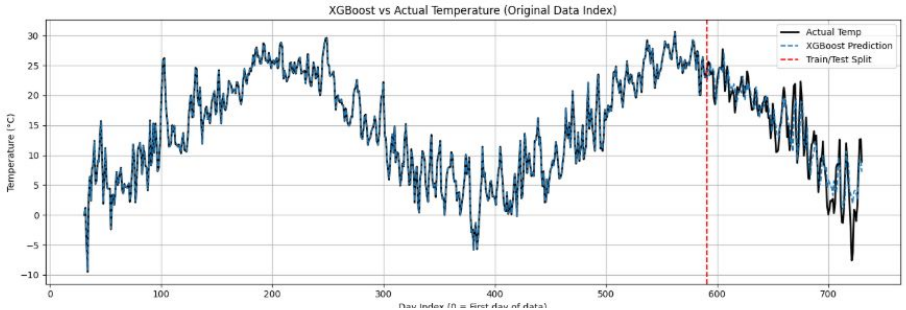
LSTM



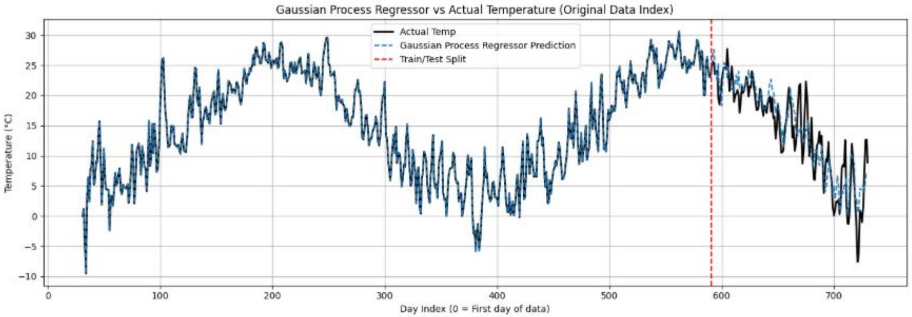
RNN



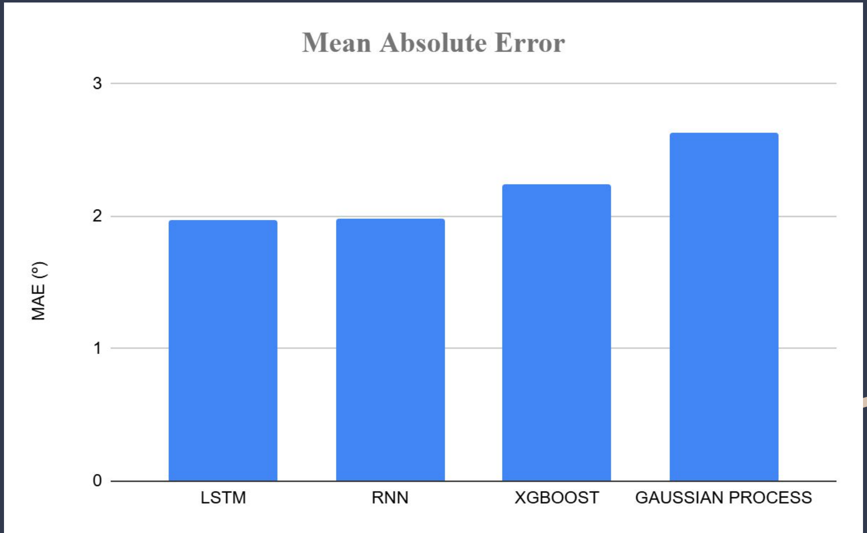
XGBoost



GP



Results



Mean Absolute Error

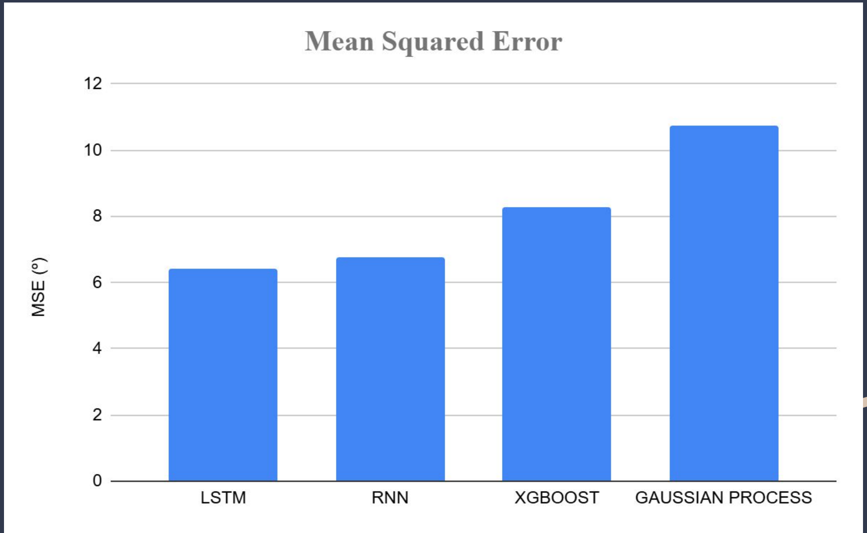
LSTM: 1,9730º

RNN: 1,9872º

XGBoost: 2,2417º

Gaussian Process: 2,6328º

Results



Mean Squared Error

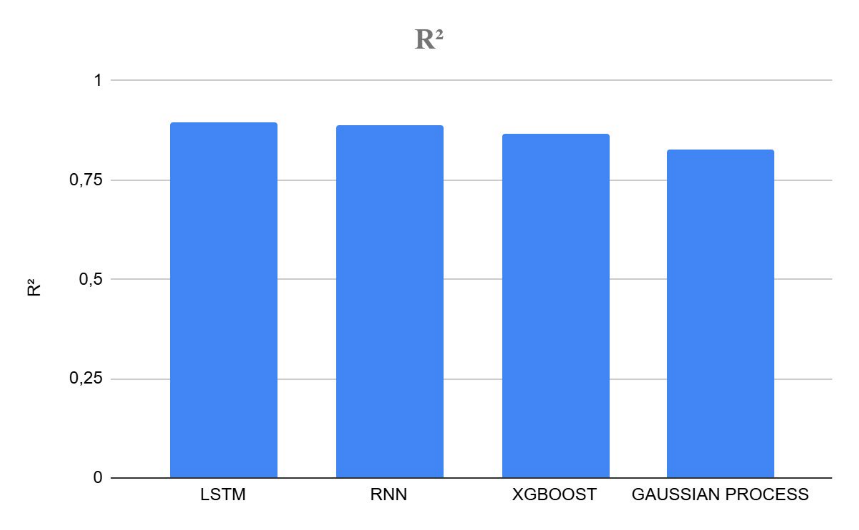
LSTM: 6,4115º

RNN: 6,7737º

XGBoost: 8,2851º

Gaussian Process: 10,7245º

Results



Coefficient of determination

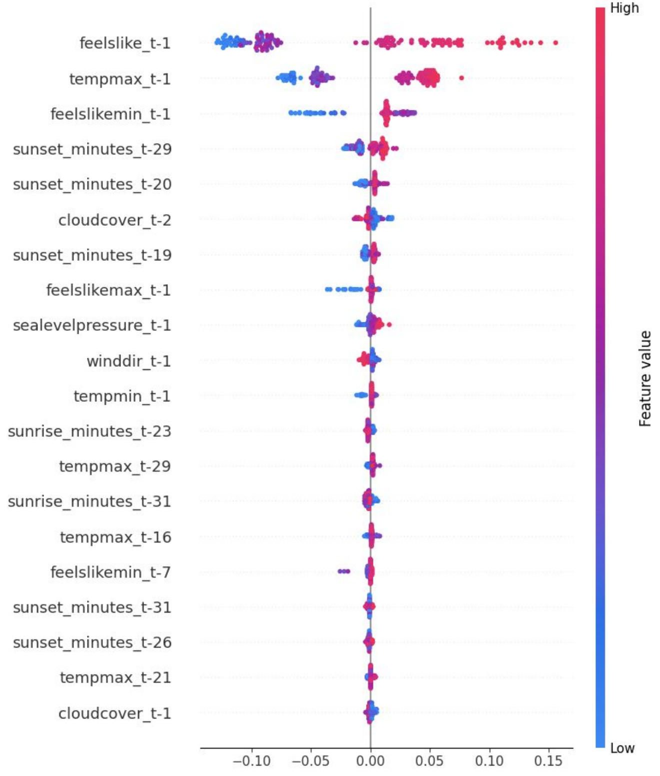
LSTM: 0,8955

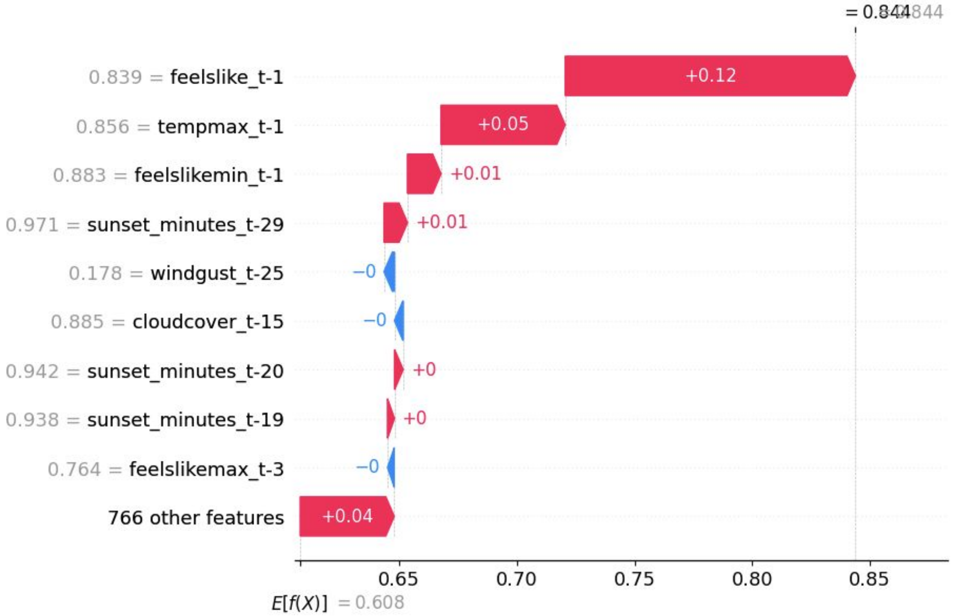
RNN: 0,8896

XGBoost: 0,8650

Gaussian Process: 0,8253

SHAP Results





Conclusions

- LSTM was the best performing model
- We don’t have a large enough dataset so that the differences between LSTM and RNN can be properly demonstrated
- Both sequential models outperformed the non-sequential models
- From SHAP results we see that the model heavily relies on recent past weather conditions, particularly temperature-related metrics.
- The results we got were satisfactory, since they were an improvement to the work in the Medium article by Ozdogar [1].

Future Work

For future works we could investigate the use of Transformers or other attention-based architectures to compare their performance with LSTM in learning temporal weather patterns.

We could also extend the study to other cities such as Valencia and reframe the task as an anomaly detection problem, focusing on extreme rainfall patterns to support early flood warning systems.

References

1. https://medium.com/@ozdogar/time-series-forecasting-using-lstm-pytorch-implementation-86169d74942e
2. New York Dataset, 2023/2024, https://drive.google.com/drive/u/1/folders/1JO_CjXqvz4Jser5GnLIsiYjAv9Atmbli

Thank you for listening!