Temperature Prediction Project Documentation

This project involves predicting temperature using a machine learning model trained on environmental and temporal features. The primary objective is to develop a model that captures patterns from particulate matter levels, gas concentrations, and various temporal variables to predict temperature accurately. The chosen model for this task is the XGBoost Regressor, owing to its robustness with tabular data and high efficiency in handling large datasets.

Data Exploration and Preprocessing

The initial dataset consists of multiple features potentially relevant to temperature prediction, including pollutants like particulate matter, SO2, O3, CO, and NO2, alongside weather and temporal data such as pressure, dew point, wind speed, and moisture percentage. Before diving into modelling, it was essential to preprocess the data, focusing on handling missing values, ensuring data consistency, and adding engineered features to capture temporal patterns.

Null values in the dataset were handled thoughtfully. We opted to impute these values with a mix of forward-fill (ffill) and backward-fill (bfill) techniques to preserve temporal integrity in the dataset. Additionally, specific missing values were filled based on the feature's characteristics. For example, temporal variables were forward-filled to ensure continuity, while pollution levels were also interpolated to maintain realistic variations.

Feature Engineering

Feature engineering played a pivotal role in enhancing model performance. First, three lag features were introduced, capturing the values of the target variable at the previous three timestamps. These lag features help the model learn from recent trends in temperature changes. Next, several temporal attributes were created, including the hour, day of the week, quarter, month, year, day of the year, day of the month, and week of the year. These features help capture cyclical and seasonal patterns, essential in temperature prediction tasks.

The final feature set included:

- Environmental and pollution variables: Particulate matter, SO2 concentration, O3 concentration, CO concentration, NO2 concentration, pressure, dew point, precipitation, wind speed, and moisture per cent.
- Lag features: Three lagged values of the target variable.
- Temporal features: Hour, day of the week, quarter, month, year, day of the year, day of the month, and week of the year.

These features form a robust base for modelling, providing information on immediate environmental factors and long-term temporal trends.

Model Selection and Training

Given the structure and requirements of the problem, an XGBoost Regressor was selected. XGBoost (Extreme Gradient Boosting) is a popular choice for tabular data. It efficiently handles large datasets, offers powerful hyperparameter tuning, and provides fast computation due to its tree-based learning method.

The model was initialized with a set of hyperparameters tuned to balance accuracy and computational efficiency:

- Learning rate: Set to 0.01 to allow for gradual convergence.
- Max depth: Set to 7, enabling the model to capture intricate patterns without overfitting.
- Early stopping rounds: 50, helping prevent overfitting by halting training when improvements cease.

The model was trained on the feature matrix (`X_train`) and the target variable (`y_train`), where a validation set was also passed to monitor the root mean squared error (RMSE) at each iteration. The early stopping mechanism ensured the model would stop training once further iterations ceased improving RMSE on the validation set.

During training, the model showed progressive improvements in RMSE. Starting with an RMSE of around 17, the model achieved an RMSE of approximately 1.13 by the end of the 1000 iterations, demonstrating a strong fit to the training data.

Model Evaluation

Post-training, the model's performance was evaluated on a separate test dataset. The scoring metric used was a custom metric designed to maximize accuracy, given by the formula: `score = max(0, 100 - mean_squared_error(y_test, y_pred))`. The model achieved an impressive score of 96.91266, indicating a high level of accuracy in predicting temperature.

Improvements and Future Work

To further enhance model performance, the following improvements could be considered:

1. Hyperparameter tuning: Performing an extensive search using GridSearchCV or Bayesian Optimization to find optimal values for parameters such as `n_estimators`, `learning_rate`, and `max_depth`.
2. Feature importance analysis: Utilizing XGBoost's feature importance scores to prune redundant or less significant features, potentially reducing overfitting and enhancing generalization.
3. Temporal smoothing techniques: Applying techniques like moving averages or exponential smoothing to input features could help capture trends more effectively.
4. External data sources: Incorporating additional relevant data such as local historical temperature patterns, geographical factors, or seasonal indices could further enhance predictive accuracy.

Conclusion

This project successfully demonstrates the application of machine learning in temperature prediction using environmental and temporal data. The combination of thoughtful feature engineering, effective model selection, and robust evaluation has resulted in a model that performs with high accuracy. By leveraging the XGBoost Regressor, we efficiently captured complex patterns within the dataset, laying a strong foundation for future improvements and potential deployment in practical applications. The final trained model, along with the code, provides a valuable resource for further exploration and refinement in predictive temperature modelling.