Delhi weather Dataset for temperature prediction

- This dataset contains a variety of meteorological variables commonly used for weather analysis and forecasting. Here's a brief description of each column and how these measurements are typically taken:

1. **datetime_utc**

   • **Description**: This column represents the date and time of each observation, typically in Coordinated Universal Time (UTC). • **Measurement**: Recorded as a timestamp (e.g., YYYY-MM-DD HH:MM:SS), which can be formatted into features such as day of the week, hour, or seasonal indicators.

2. **_conds**

   • Description: Represents observed weather conditions, often as text (e.g., "Clear," "Cloudy," "Rainy"). • **Measurement**: Recorded by human observers or sensors, this is categorical and can be transformed into dummy/indicator variables for model training.

3. **_dewptm (Dew Point in °C or °F)** • **Description**: Dew point temperature, indicating the temperature at which air becomes saturated with moisture. • **Measurement**: Measured in degrees Celsius or Fahrenheit, typically with specialized hygrometers.

4. **_fog**

   • **Description**: Binary indicator (0 or 1) of fog presence. • **Measurement**: Derived from visibility sensors or direct observation.

5. **_hail**

   • Description: Binary indicator of hail occurrence. • Measurement: Based on direct observation or automated weather stations with hail **sensors**.

6. **_heatindexm** (Heat Index)

   • Description: Apparent temperature considering both air temperature and humidity, indicating how hot it feels. • Measurement: Calculated using a formula based on temperature and humidity readings.

7. **_hum** (Humidity)

   • Description: Relative humidity, the percentage of moisture in the air relative to the maximum amount it can hold. • Measurement: Measured as a percentage using a hygrometer.

8. **_precipm** (Precipitation)

   • Description: Amount of precipitation, such as rain or snow, in millimeters (mm) or inches.
   • Measurement: Recorded by rain gauges or radar systems.

9. **_pressurem** (Pressure)

    • Description: Atmospheric pressure, often measured in millibars (mb) or hectopascals (hPa). • Measurement: Measured using barometers at ground level or derived from weather balloons.

10. **_rain**

    • Description: Binary indicator of rain occurrence. • Measurement: Based on rain sensor or gauge detection.

11. **_snow**

    • Description: Binary indicator of snow occurrence. • Measurement: Derived from snow sensors or manual observation.

12. **_tempm** (Temperature)

    • Description: Air temperature at the observation point. • Measurement: Measured in °C or °F with thermometers or digital sensors.

13. **_thunder**

    • Description: Binary indicator of thunder occurrence. • Measurement: Based on lightning detection systems or auditory sensors.

14. **_tornado**

    • Description: Binary indicator of tornado occurrence. • Measurement: Typically based on direct reports or radar data in meteorological services.

15. **_vism** (Visibility)

    • Description: Measure of visibility in meters or kilometers, indicating the distance one can clearly see. • Measurement: Derived from visibility sensors or manual observation.

16. **_wdird** (Wind Direction in Degrees)

    • Description: Wind direction in degrees, where 0° = North, 90° = East, 180° = South, and 270° = West. • Measurement: Recorded using wind vanes and converted to degrees.

17. **_wdire** (Wind Direction as Text)

    • Description: Wind direction represented as a compass direction (e.g., “N,” “NE”). • Measurement: Derived from _wdird and converted into text.

18. **_wgustm** (Wind Gust Speed)

    • Description: Maximum instantaneous wind speed during a given period, typically in meters per second or miles per hour. • Measurement: Measured with anemometers.

19. **_windchillm** (Wind Chill)

    • Description: Perceived decrease in air temperature felt by the body on exposed skin due to wind. • Measurement: Calculated using a formula based on temperature and wind

speed.

20. **_wspdm** (Wind Speed)

    • Description: Average wind speed, often in meters per second or miles per hour. • Measurement: Measured with anemometers at various observation points.

Summary of Usage

```
•    Binary Indicators (_fog, _hail, _rain, _snow, _thunder, _tornado): Useful for classi
•    Continuous Variables (_tempm, _dewptm, _hum, etc.): Provide valuable information for
•    Categorical and Text Variables (_conds, _wdire): Can be converted to numeric represe
```

Using this dataset to predict temperature can be quite insightful, as temperature prediction is influenced by multiple atmospheric variables. Here's an overview of how temperature prediction can be approached and what factors to consider:

Key Points on Temperature Prediction

- **Influential Variables**: • Variables like humidity (_hum), dew point (_dewptm), wind speed (_wspdm), and pressure (_pressurem) are critical, as they correlate closely with temperature fluctuations. • Precipitation (_precipm) and weather conditions (_conds) can also affect temperature. For example, cloudy or rainy conditions generally lead to cooler temperatures. • Time-based Factors: datetime_utc is particularly important for capturing seasonal patterns and daily cycles in temperature.
- **Data Preprocessing**: • Normalization/Standardization: Since temperature is a continuous variable, scaling it along with other continuous features (e.g., dew point, wind speed) can improve model performance. • Handling Categorical Data: For _conds or _wdire, consider one-hot encoding or embedding layers (if using neural networks) to convert these categories into numeric representations.
- **Model Approach**: • Sequence Models: LSTM, GRU, or Transformer models are ideal for predicting temperature, as they can capture temporal dependencies in weather data. • Feature Engineering: Creating features such as previous temperature readings or moving averages can help capture trends and cycles that influence temperature.
- **Challenges and Considerations**: • Seasonal Variation: Temperature varies widely with seasons, and capturing this variation is crucial for accuracy. • Daily Cycles: Temperature tends to peak in the afternoon and drop during the night. Including hourly patterns in your model (e.g., encoding time of day) can enhance performance. • External Factors:

Temperature can also be affected by local geographical conditions, which might be beyond the scope of the dataset but are worth considering for real-world deployment.

- **Evaluation and Interpretation** • Metrics: Use metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) to assess the accuracy of your predictions. • Visualization: Plot predicted vs. actual temperatures over time to check if your model captures the expected daily and seasonal patterns.

Temperature prediction can provide valuable insights and is essential for various applications, from weather forecasting to energy demand management and agriculture planning.

Understanding the LSTM Model for Temperature Forecasting

In your LSTM model for time series forecasting, the code uses a **sliding window approach** to create sequences of input data for training, capturing patterns over a specific recent period. Here's a breakdown of how it works based on your code:

---

1. Input Sequences Creation

- The dataset is split into sequences, where each sequence represents a short period of time (often a few days or hours) leading up to the prediction point.
- In your code, you're using sequences of **10 time steps** (or days if the data is daily), meaning each sequence contains data from the past 10 days to predict the temperature on the next day.

---

2. How LSTM Uses Past Days for Prediction

- The LSTM model processes each 10-day sequence and learns the **temporal relationships** between values in each step of the sequence. It captures both:
  - **Short-term dependencies** (e.g., yesterday's conditions affecting today's).
  - **Longer-term trends** (e.g., seasonal patterns).
- This is achieved by "remembering" patterns in sequences of 10 days, using the **cell state** and **hidden state** to retain relevant information from each step in the sequence.

---

3. Prediction for Each Sequence

- During training, the model learns to predict the temperature (`_tempm`) based on the last day in each 10-day sequence. This sliding window moves across the dataset, generating sequences that help the model generalize and capture recurring patterns.

- When making predictions on new data, you feed in a 10-day sequence, and the model outputs the predicted temperature for the next day.

---

4. Why Use 10 Days?

- The choice of **10 days** as the sequence length (often called the **look-back window**) is an adjustable parameter:
    - A **longer window** might capture longer-term trends but requires more computation and risks overfitting.
    - A **shorter window** might capture only immediate patterns but miss long-term trends.
- If you change this look-back window to, say, **30 days**, the model will take a longer history into account for each prediction, which can be useful if temperature trends are influenced by monthly or seasonal cycles.

Summary

In essence, your LSTM model is learning to predict temperature by analyzing temperature and other factors over the past 10 days in each sequence. It finds patterns in these sequences, which it then applies to predict future temperatures based on the most recent 10-day window of data.