**Weather Forecasting with LSTM**

* **Introduction**

Weather forecasting is a vital application of data science and artificial intelligence, designed to anticipate future weather conditions based on existing data. Precise predictions impact several industries, including agriculture, transportation, and disaster management, enhancing efficiency and safety. This study utilizes Long Short-Term Memory (LSTM) neural networks to forecast weather conditions by analyzing temporal trends in **historical data**.

* **Methodology**

The technique comprises many stages: data preprocessing, feature engineering, model construction, training, and assessment. The LSTM model identifies sequential relationships in meteorological data, enabling precise forecasts of future conditions. The methodology guarantees a resilient pipeline for processing extensive, time-series meteorological data via a synthesis of scaling, encoding, and sequence preparation.

* **Dataset Description**

The dataset used in this project contains hourly weather observations for the year 2012. It includes key meteorological attributes such as:

* **Temperature**: The air temperature in degrees Celsius.
* **Dew Point Temperature**: The temperature at which air reaches saturation.
* **Relative Humidity**: The moisture level in the air.
* **Wind Speed**: The velocity of the wind in kilometers per hour.
* **Visibility**: The distance visible to the human eye in kilometers.
* **Atmospheric Pressure**: The pressure exerted by the atmosphere, measured in kilopascals.
* **Weather Condition**: Categorical data describing conditions such as Fog, Rain, or Clear.

This dataset includes 8784 entries, covering every hour of the year. All columns were complete, with no missing values after preprocessing.

**Source: - https://www.kaggle.com/datasets/bhanupratapbiswas/weather-data**

* **Data Analysis**

Initial exploration revealed:

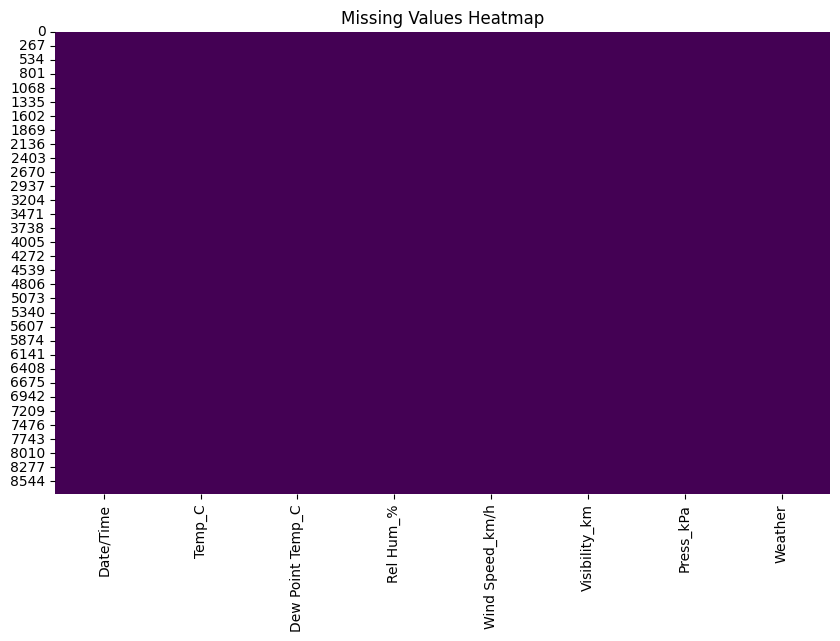
* Temperature ranged from -23.3°C to 33.0°C, with a mean of 7.2°C.
* Relative humidity varied widely, indicating diverse weather conditions, with values between 10% and 100%.
* Wind speed was generally low, with occasional peaks up to 48.3 km/h.
* Atmospheric pressure fluctuations were moderate, with values between 97.52 kPa and 103.65 kPa.

Understanding these patterns helped define preprocessing steps and interpret the predictions effectively.

* **Data Preparation**

To prepare the dataset for LSTM modeling:

1. **Handling Missing Data**: A forward-fill method ensured continuity in the time series without creating artificial trends.



1. **Feature Scaling**: Continuous variables like temperature and wind speed were normalized using MinMaxScaler to standardize their values between 0 and 1, enhancing model training stability.

A screenshot of a computer

Description automatically generated

1. **Feature Engineering**: Lag features, such as temperature and humidity from the previous hours, were created to incorporate historical context into the model.
2. **Categorical Encoding**: The Weather column was one-hot encoded to transform categorical descriptions into numerical data suitable for machine learning models.
3. **Sequence Generation**: Input sequences of 72 hours were created, representing three days of data. Each sequence was paired with the next hour’s data as the target for prediction.

* **Model Architecture**

The LSTM model comprises:

1. **Input Layer**: Processes sequences of shape (72, number of features), representing three days of hourly data.
2. **LSTM Layers**: Three stacked LSTM layers capture temporal dependencies in the data, making the model robust to long-term trends and seasonal patterns.
3. **Dropout Layer**: Prevents overfitting by randomly disabling a fraction of neurons during training.
4. **Dense Layers**: Fully connected layers map the learned temporal patterns to predictions for each feature.
5. **Output Layer**: Provides numerical predictions for continuous features and probabilistic outputs for categorical weather conditions.

* **Training Configuration**

The model was trained using:

* **Loss Function**: Mean Squared Error (MSE) to minimize the difference between predicted and actual values.
* **Optimizer**: Adam optimizer with a learning rate of 0.0001 for smooth convergence.
* **Batch Size**: 64 sequences processed simultaneously for efficient gradient updates.
* **Validation Split**: 10% of the data was reserved for validation to monitor performance during training.
* **Early Stopping**: Prevented overfitting by halting training if validation loss did not improve for a specified number of epochs.
* **Evaluation and Results**

The model achieved:

* **Training MSE**: 0.0126, indicating minimal error during training.
* **Validation MSE**: 27.89, demonstrating a reasonable ability to generalize.
* **R-squared Value**: 0.63, reflecting moderate accuracy in capturing variance in the data.

Predictions showed strong performance in forecasting temperature and humidity trends, with reasonable accuracy in categorical weather conditions like Fog and Rain.

A graph of a graph showing a number of blue and orange lines

Description automatically generated

A screen shot of a computer

Description automatically generated

* **Implementation Steps**

1. **Environment Setup**: Python (3.8+) was used with libraries such as TensorFlow, Pandas, NumPy, and Scikit-learn.
2. **Data Loading and Preprocessing**: The raw dataset was transformed into a suitable format for the LSTM model by scaling, encoding, and sequence preparation.
3. **Model Training**: The LSTM model was trained over 100 epochs, leveraging early stopping to optimize performance.
4. **Forecasting**: Predictions were generated for specific dates and times, as well as for extended periods like the next 7 days.
5. **Streamlit Deployment**: A user-friendly interface was created to interact with the forecasting model.

* **How to Run**

1. Install dependencies:
2. pip install tensorflow pandas numpy scikit-learn matplotlib streamlit
3. Place the preprocessed dataset and the trained model in the appropriate directory.
4. Run the script using Streamlit:
5. streamlit run weather\_forecast\_app.py
6. Access the app through the provided localhost link in your web browser.

* **Conclusion**

This weather forecasting initiative illustrates the efficacy of LSTM neural networks in modeling time-series data. The amalgamation of preprocessing, feature engineering, and resilient model architecture yielded a dependable solution for forecasting essential meteorological variables. Additional enhancements, including the integration of external data such as satellite images or real-time meteorological reports, might augment the model's precision and versatility across many applications.