
Air Pollution Forecasting
Using LSTM

DATA

- This is a dataset that reports on the weather and the level of pollution each hour for five years at the US embassy in Beijing, China.

- The data includes the date-time, the pollution called PM2.5 concentration, and the weather information including dew point, temperature, pressure, wind direction, wind speed and the cumulative number of hours of snow and rain.

- We use this data and frame a forecasting problem where, given the weather conditions and pollution for prior hours, we forecast the pollution at the next hour.

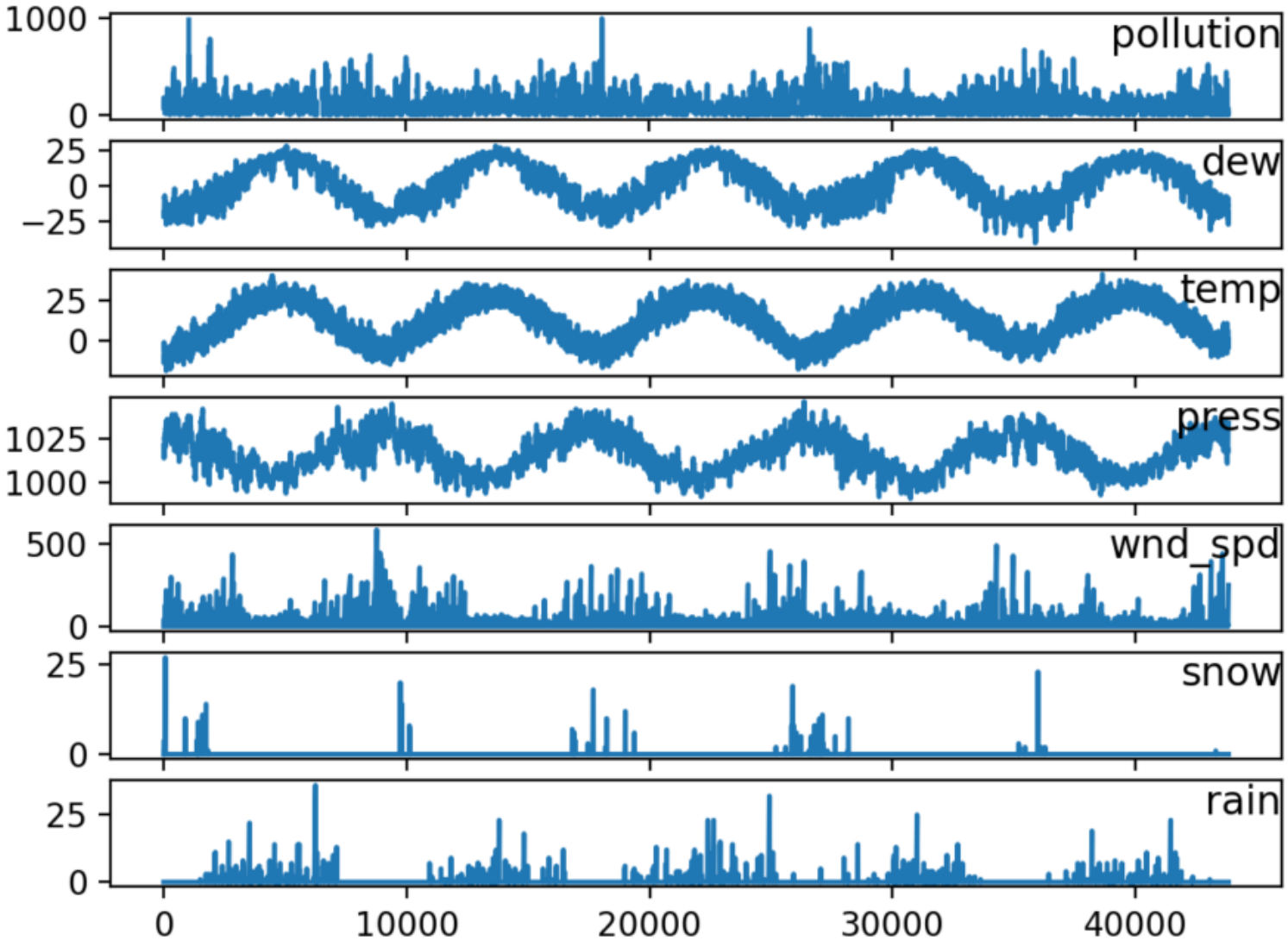
BASIC DATA PREPARATION

- The first step is to consolidate the date-time information into a single date-time so that we can use it as an index in Pandas.

- A quick check reveals NA values for pm2.5 for the first 24 hours. We will, therefore, need to remove the first row of data. There are also a few scattered “NA” values later in the dataset; we can mark them with 0 values for now.

- We plots each series as a separate subplot, except wind speed direction, which is categorical.

Plots of each series as a separate subplot



Multivariate LSTM Forecast Model

This involves framing the dataset as a supervised learning problem and normalizing the input variables. We will frame the supervised learning problem as predicting the pollution at the current hour (t) given the pollution measurement and weather conditions at the prior time step.

Define and Fit Model

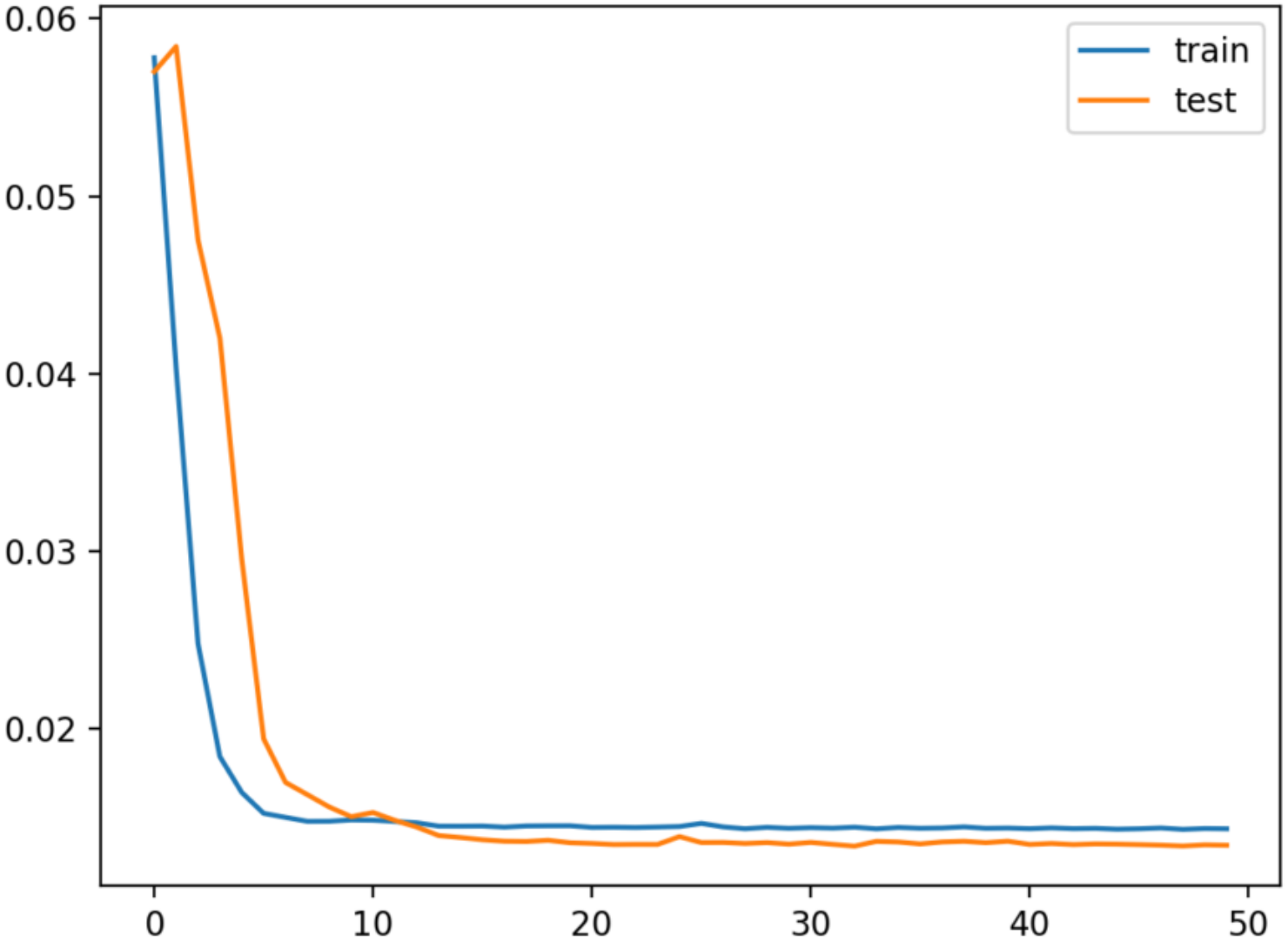
- First, we must split the prepared dataset into train and test sets. To speed up the training of the model for this demonstration, we will only fit the model on the first year of data, then evaluate it on the remaining 4 years of data.

- We will define the LSTM with 50 neurons in the first hidden layer and 1 neuron in the output layer for predicting pollution. The input shape will be 1 time step with 8 features.
-
  We will use the Mean Absolute Error (MAE) loss function and the efficient Adam version of stochastic gradient descent.
-
  Finally, we keep track of both the training and test loss during training. At the end of the run both the training and test loss are plotted.

Model Evaluation

- After the model is fit, we can forecast for the entire test dataset. We combine the forecast with the test dataset and invert the scaling. We also invert scaling on the test dataset with the expected pollution numbers.

- With forecasts and actual values in their original scale, we can then calculate an error score for the model. In this case, we calculate the Root Mean Squared Error (RMSE) that gives error in the same units as the variable itself.



Running the example first creates a plot showing the train and test loss during training

Interestingly, we can see that test loss drops below training loss. The model may be overfitting the training data. Measuring and plotting RMSE during training may shed more light on this.

The Train and test loss are printed at the end of each training epoch. At the end of the run, the final RMSE of the model on the test dataset is printed.

We can see that the model achieves a respectable RMSE of 26.496, which is lower than an RMSE of 30 found with a persistence model.

Epoch 46/50
0s - loss: 0.0143 - val_loss: 0.0133
Epoch 47/50
0s - loss: 0.0143 - val_loss: 0.0133
Epoch 48/50
0s - loss: 0.0144 - val_loss: 0.0133
Epoch 49/50
0s - loss: 0.0143 - val_loss: 0.0133
Epoch 50/50
0s - loss: 0.0144 - val_loss: 0.0133
Test RMSE: 26.496