paper06_model_best

October 12, 2021

1 Timeseries Testing and Modeling

Jorge III Altamirano-Astorga, Ita-Andehui Santiago, Luz Aurora Hernández.

Prof.: Edgar Francisco Román-Rangel.

```
                     temperature    pressure  …  wind_speed    wind_deg
datetime                                      …
2021-02-12 06:00:00    21.530000  777.410000  …    2.565310  109.799270
2021-02-12 06:05:00    21.689773  777.389432  …    2.456273  105.132299

[2 rows x 20 columns]
```

1.1 Timeseries

We use the the `timeseries_dataset_from_array` function from Keras Timeseries modeling functions. This function creates dataframes with sliding windows over time as an array.

This function work as follow:

```
tf.keras.utils.timeseries_dataset_from_array(
    X, Y,              # our dataset
    sequence_length,   #  Length of the output sequences (in number of timesteps): we need to text different s
    sequence_stride=1, # 1 is the default value for s, data[i], data[i + s], data[i + 2 * s]
    sampling_rate=1,   # 1 is the default value for timesteps data[i], data[i + r], ... data[i + sequence_leng
    shuffle=False,     # shuffle the records sorting, we set it in the default false value as the original ord
    seed=None,         # we set a fixed seed to have repeatable experiments
    #...
)
```

On our research it is relevant to have an adequate sequence length. Then we can focus a brief research on empiric good times. Our proposal are:

1.1.1 1 Minute Resampling

- 2 days before: i.e. on our resampling for every 5 min we'd have 2880 records. This is because 2 $days \times 24$ $hours \times 60$ $min \div 1$ min
- 7 days before: i.e. on our resampling for every 5 min we'd have 10,080 records. This is because 7 $days \times 24$ $hours \times 60$ $min \div 1$ min
- 15 days before: i.e. on our resampling for every 5 min we'd have 21600 records. This is because 15 $days \times 24$ $hours \times 60$ $min \div 1$ min

1.1.2 2 Minute Resampling

- 2 days before: i.e. on our resampling for every 5 min we'd have 1,440 records. This is because 2 $days \times 24$ $hours \times 60$ $min \div 2$ min
- 7 days before: i.e. on our resampling for every 5 min we'd have 5040 records. This is because 7 $days \times 24$ $hours \times 60$ $min \div 2$ min
- 15 days before: i.e. on our resampling for every 5 min we'd have 10,080 records. This is because 15 $days \times 24$ $hours \times 60$ $min \div 2$ min

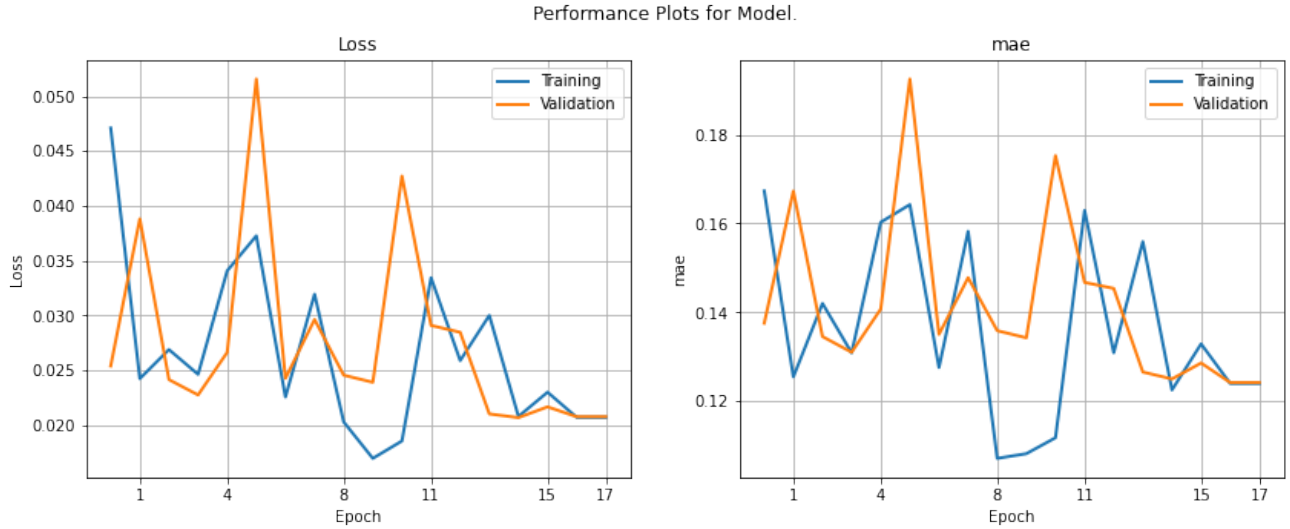
1.1.3 5 Minute Resampling

- 2 days before: i.e. on our resampling for every 5 min we'd have 576 records. This is because 2 $days \times 24\ hours \times 60\ min \div 5\ min$
- 7 days before: i.e. on our resampling for every 5 min we'd have 2016 records. This is because 7 $days \times 24\ hours \times 60\ min \div 5\ min$
- 15 days before: i.e. on our resampling for every 5 min we'd have 4320 records. This is because 15 $days \times 24\ hours \times 60\ min \div 5\ min$

We set this the number of days in a variable we set as `WINDOW_SIZE_DAYS`

1.2 5 Minute Resampling and 7 Days of History.

`Processing Time: 228.97 segundos.`

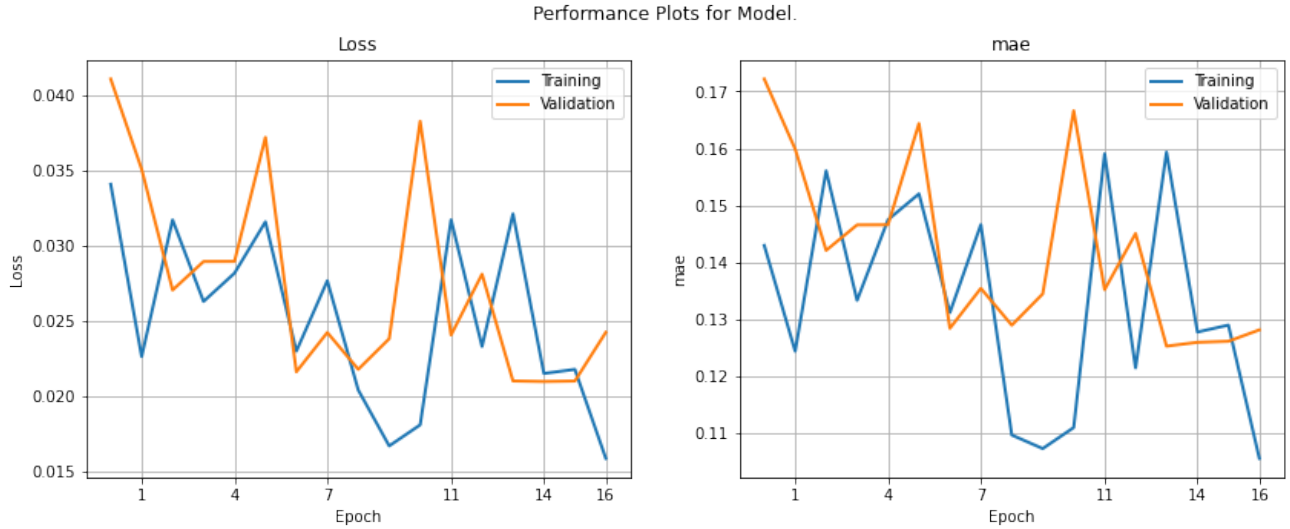


- Mean Absolute Error in Real Scale of the last Epoch: **61.5482 IAQ points**.

1.3 5 Minute Resampling and 2 Days of History with 2 Sampling Rate.

On our previous examples this quickly becomes unmanageable the we propose skipping some records and getting the previous hour by setting the `sampling_rate` parameter of 10 minute.

`Processing Time: 220.42 segundos.`



- Mean Absolute Error in Real Scale of the last Epoch: **63.5228 IAQ points**.