Research

© Leon Jerg

Contents

May, 2022

Introduction

My enthusiasm for the analysis of process parameters has accompanied me throughout my studies and professional experience.

I consider it very important not only to recognize anomalies and behavioral patterns, but also to derive the right measures from them so that an economic added value can be created. No matter what industry you are in, data-driven business analysis always plays a major role and this is where my focus lies.

An indication of my approach can be taken from the following report. I had the idea to analyze the health data of my Smart Watch and Phone. Here is a little outlook.

Health Data Report

To keep things simple, I chose the number of steps per day I walked myself over the last few years as a process parameter. Accordingly, the imported time-series data looks like this.

```
## # A tibble: 6 x 2
##   Date                 Step.Count
##   <dttm>                    <int>
## 1 2016-11-23 00:00:00        4810
## 2 2016-11-24 00:00:00        4900
## 3 2016-11-25 00:00:00       17241
## 4 2016-11-26 00:00:00        8984
## 5 2016-11-27 00:00:00        5258
## 6 2016-11-28 00:00:00        3451
```

```
Average.Per.Day <- mean(Dt$Step.Count)
Average.Per.Day
```

```
## [1] 6811.798
```

From the year 2016 on, I have walked an average of about 7,000 steps per day, which is quite a bit :-)

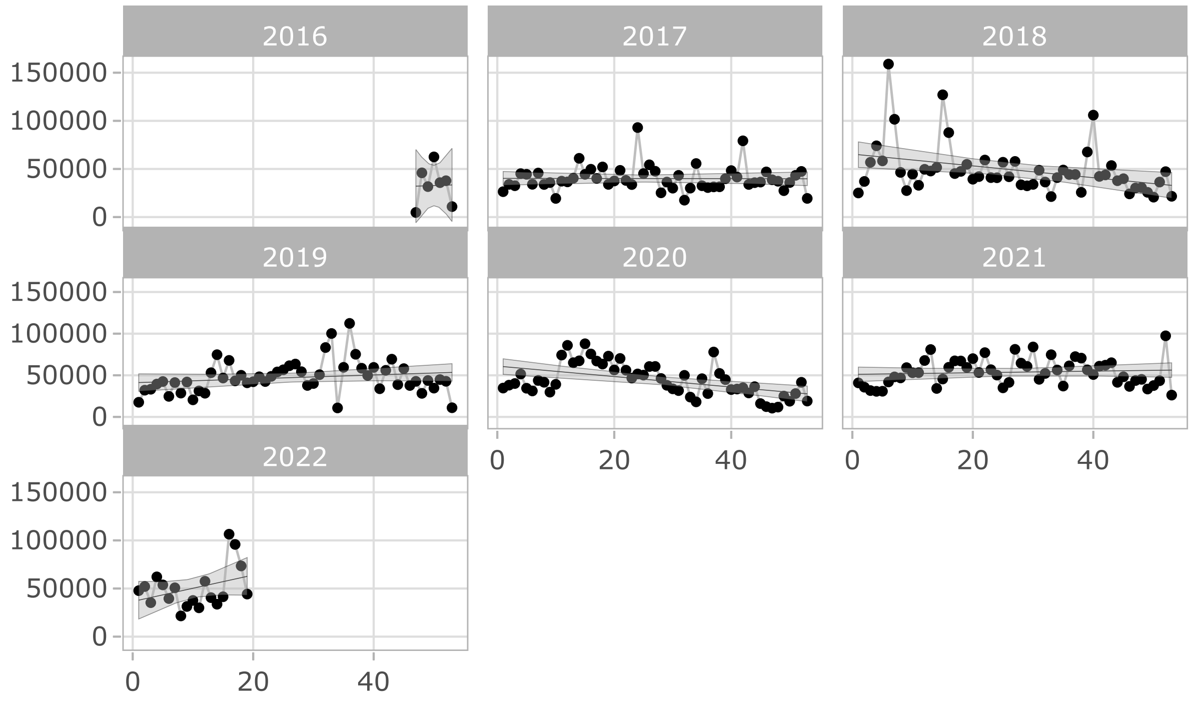
Best-practice data analysis always involves two basic things.

- Propper data visualization
- Derivation and implementation of enhancing measures

Especially for time-series analysis this approach is very important to understand the data one is dealing with.

Visualization

Let's start with an appropriate data visualization by plotting the calendar week against the summed number of steps per week. Please note that all integrated plots are interactive and can be played with by clicking on the navigation bar in the upper right hand corner :-)



It can be seen that I have become better and more consistent on average over the years, especially in 2021. This has a particularly positive effect on my fitness, which is a great advantage as an active tennis and soccer player :-)

Since I naturally want to get better and better over time, I wonder how my performance will develop in the short and medium term. Accordingly, I had the idea to train a Machine Learning algorithm based on the historical data, which forecasts my performance.

This helps me to identify low-performance periods in advance to become even more consistent in my development.

Deployment

I decided to train a **Long-Short-Term Memory** *LSTM* **Neural Network** *NN* on the basis of the historical data with the aim of predicting the weekly *Step.Count.Sum* average for 50 weeks in advance [~1 year]. An *LSTM* has the advantage that it takes into account long-term dependencies and does not suffer from the vanishing gradient problem.

The approach is to train the model and determine the architecture using all the available data from 2016 to 2021. The model is then validated using the remaining 2021 and 2022 data and a forecast is generated for the remaining 2022 and some 2023 weeks.

Only *Step.Count.Sum* observations from past weeks within past years are used as a source of information for algorithm training. In consequence, the data is prepared with regard to the underlying dimensions. The detailed `R` Code is hidden, but can be made available upon request.

In order to predict the target values for the upcoming pred = 50 weeks, the algorithm lapse back lag = 50 weeks starting from the very beginning and at each iteration. As a result and regarding the underlying data, no target values are conducted by the algorithm for the first 50 observational weeks. However, the estimated target values of the 50 future weeks are delivered instead as a forecast.

```r
#Step_04
pred <- 50
lag <- pred
```

The model architecture consists of several hidden layers with batch normalization and dropout regularization to avoid overfitting. In addition, backpropagation is performed using a gradient-based optimization technique. The **Mean Absolute Error** *MAE*, is used as an evaluation criterion to address the gap between the original observations [G.Truth] and the predictions [V.Fitted].

The exact model architecture is summarized as follows.

```r
#Step_06
##Seed
library("keras")
library("tensorflow")

set.seed(2022)
set_random_seed(2022, disable_gpu = TRUE)

##Model architecture
lstm_model <- keras_model_sequential()

set.seed(2022)
lstm_model %>%
  layer_lstm(units = 50,
             batch_input_shape = c(1, 50, 4),
             return_sequences = TRUE,
             stateful = TRUE) %>%
  layer_batch_normalization() %>%
  layer_dropout(rate = 0.5) %>%
  layer_lstm(units = 200,
             return_sequences = TRUE,
             stateful = TRUE) %>%
  layer_batch_normalization() %>%
  layer_dropout(rate = 0.5) %>%
  layer_lstm(units = 200 ,
           return_sequences = TRUE,
           stateful = TRUE) %>%
  layer_batch_normalization() %>%
  layer_dropout(rate = 0.5) %>%
  layer_lstm(units = 200 ,
           return_sequences = TRUE,
           stateful = TRUE) %>%
  layer_batch_normalization() %>%
  layer_dropout(rate = 0.5) %>%
  time_distributed(keras::layer_dense(units = 1))

##Optimizer and evaluation metric
set.seed(2022)
lstm_model %>%
  compile(loss = "mae", optimizer = "adam", metrics = "accuracy")

##Model architecture summary
summary(lstm_model)
```

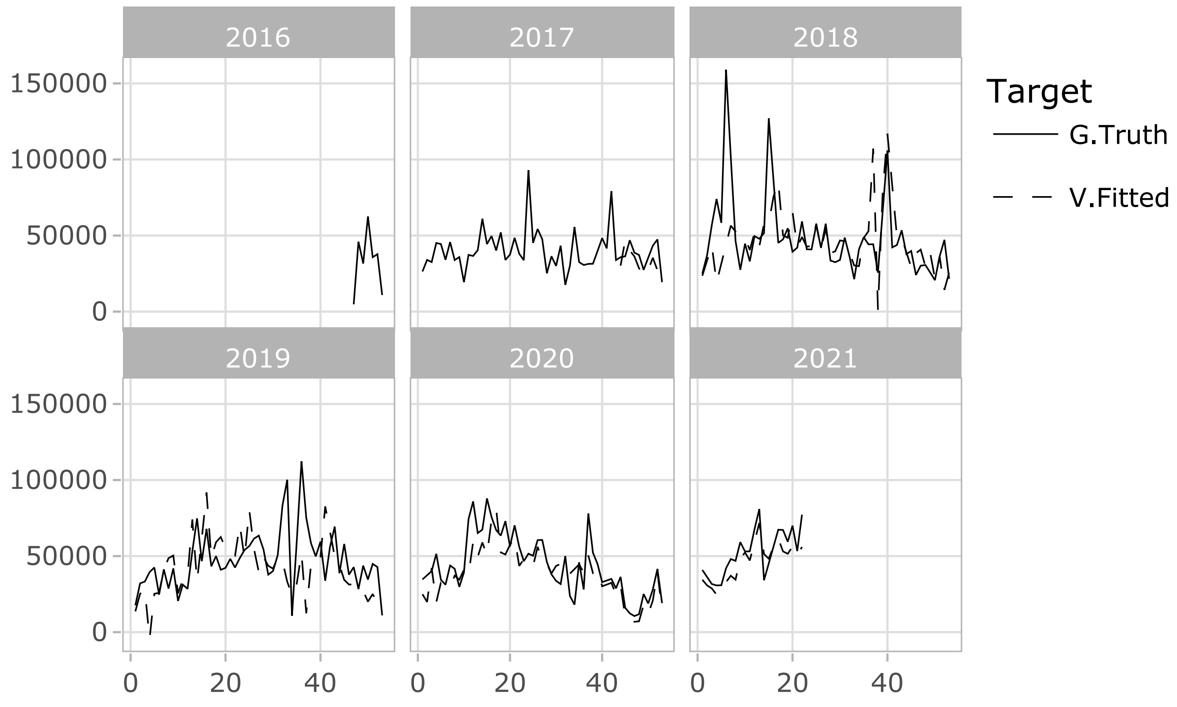
```
## Model: "sequential_1"
## ________________________________________________________________________________
##  Layer (type)                  Output Shape               Param #    Trainable
## ================================================================================
##  lstm_7 (LSTM)                 (1, 50, 50)                11000      Y
##  batch_normalization_7 (BatchN  (1, 50, 50)               200        Y
##  ormalization)
##  dropout_7 (Dropout)           (1, 50, 50)                0          Y
##  lstm_6 (LSTM)                 (1, 50, 200)               200800     Y
##  batch_normalization_6 (BatchN  (1, 50, 200)              800        Y
##  ormalization)
##  dropout_6 (Dropout)           (1, 50, 200)               0          Y
##  lstm_5 (LSTM)                 (1, 50, 200)               320800     Y
##  batch_normalization_5 (BatchN  (1, 50, 200)              800        Y
##  ormalization)
##  dropout_5 (Dropout)           (1, 50, 200)               0          Y
##  lstm_4 (LSTM)                 (1, 50, 200)               320800     Y
##  batch_normalization_4 (BatchN  (1, 50, 200)              800        Y
##  ormalization)
##  dropout_4 (Dropout)           (1, 50, 200)               0          Y
##  time_distributed_1 (TimeDistr  (1, 50, 1)                201        Y
##  ibuted)
## ================================================================================
## Total params: 856,201
## Trainable params: 854,901
## Non-trainable params: 1,300
## ________________________________________________________________________________
```

The next step is to train and tune the model. The goal is to find hyperparameters that achieve low training and validation error at the same time. The following parameters are finally picked.

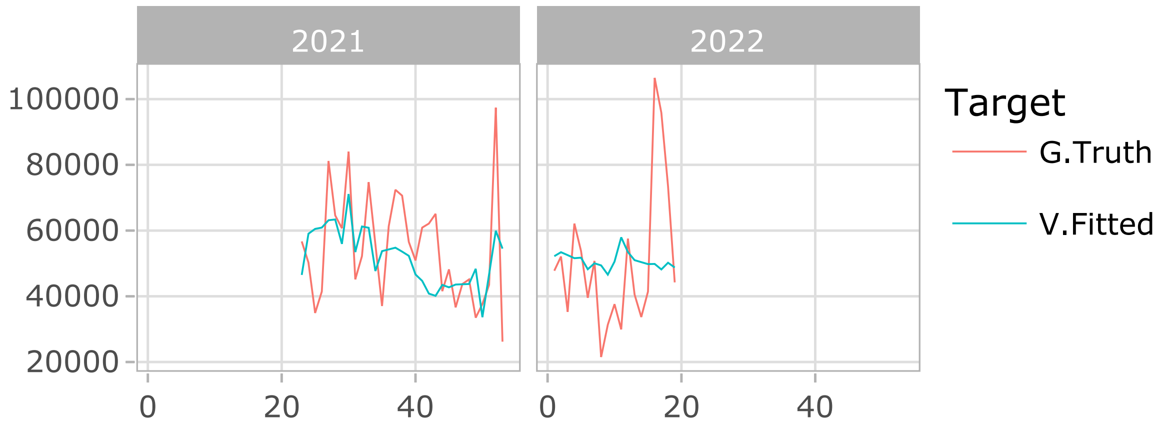
```r
#Step_07
##Hyperparameter setting & Network training
set.seed(2022)
```

```
lstm_model %>% fit(
  x = x_tr.arr.01,
  y = y_tr.arr.01,
  batch_size = 1,
  epochs = 100,
  verbose = 0,
  shuffle = FALSE
)
```

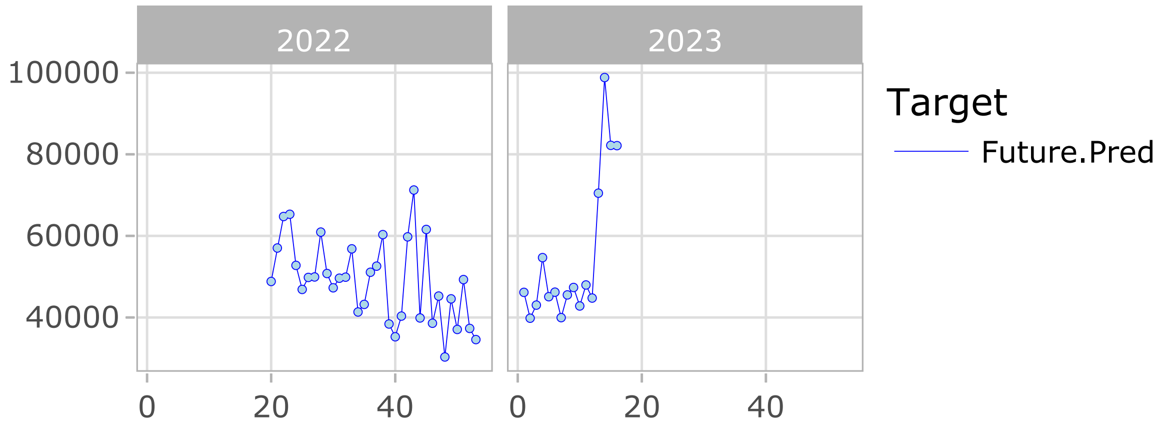
The evaluation of the training data is shown graphically as follows.



Since the training performance looks very reasonable, it is further investigated how well the model performs on unseen data. Therefore, the evaluation of the validation data is shown graphically as follows.



Since both the training and validation performance look reasonable, the model can be used to generate a forecast. The forecast covers most of the year 2022 and even a short period in 2023, whereby it is particularly relevant for my personal scheduling. It is shown graphically as follows.



This small case study shows my approach to solving data-driven problems. In my opinion, the potential of AI-based Business Analytics methods around the globe is still underestimated and the possibilities unimaginable.

As with my “step-a-day” performance, I constantly try to achieve even better and more robust results with my research and I am very interested in learning and developing in a professional environment :-)