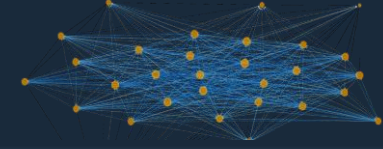




# **Präzise Prognosen: Warum individuelle Modelle in der Zeitreihenanalyse die generative KI (noch) übertreffen**

# Über das Unternehmen



Strukturen in Daten  
erkennen und  
Zusammenhänge  
visualisieren

Kundenspezifische  
Angebote  
auswählen

Bilder klassifizieren  
und Objekte  
lokalisieren

Punktwolken  
analysieren und 3D-  
Objekte erkennen

In Videos  
Bewegungen  
analysieren

Zeitreihen  
fortsetzen

Sensordaten  
analysieren und  
Geräteausfälle  
prognostizieren

Prognosen mit  
Bayes'schen  
neuronalen Netzen  
statistisch  
bewerten

Texte auslesen,  
übersetzen und  
zusammenfassen

Chatbots und KI-  
Agenten

Dynamische  
Systeme mit  
Reinforcement  
Learning steuern

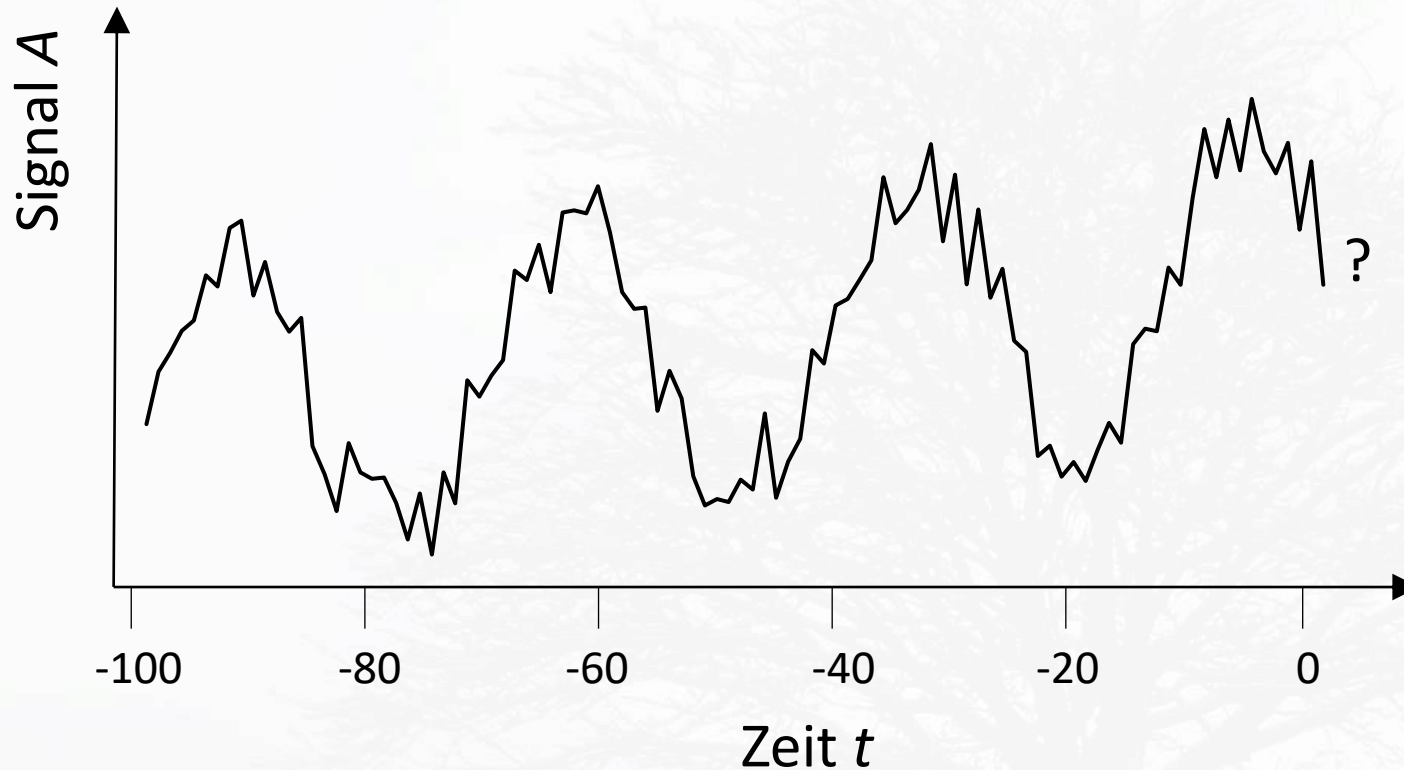
Und hier ist Raum  
für Ihr individuelles  
Projekt

- Seit 06/2020:      Selbständiger Dozent für Machine Learning und Deep Learning
- 2016 – 2020:      Postdoc am Karlsruher Institut für Technologie
- 2011 – 2015:      Promotion und Postdoc an der University of St Andrews, UK
- 2009 – 2011:      Masterstudium Photonik an der Hochschule München
- 2008 – 2010:      Ingenieur bei der PNDetector GmbH in München
- 2003 – 2007:      Diplomstudium Allgemeine Physik an der nta in Isny



- Langfristige und kurzfristige Prognose von Verkaufszahlen
- Zukünftige Preisentwicklung von Rohstoffen
- Arbeitsmarktzahlen, Wirtschaftswachstum
- Wetter
- Unwetter, Katastrophen
- Klimamodelle
- Text-, Video- und Musikgeneration

# Zeitreihenprognosen

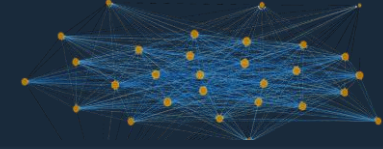


Aus Daten der Vergangenheit, die Zukunft abschätzen

Vorgehensweise:  
Modell der bisherigen Daten erstellen:

$$A[t > 0] = f(A[t < 0])$$

# Generative KI mit Foundation Models



Foundation Models können ohne individuelle Anpassung auf neuen Daten verwendet werden

## Große Sprachmodelle

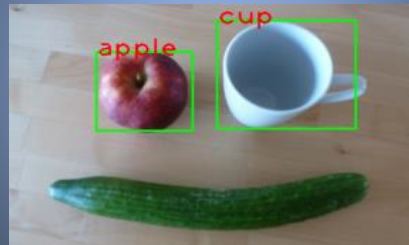
(248,668)

Was ist die Hauptstadt von Japan? ►

*Die Hauptstadt von Japan ist Tokio*

## Objekterkennung

(3,597)



## Bildklassifikation

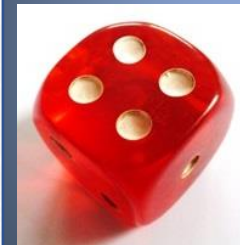
(18,325)



*Agaric (80.6%)*

## Vision-Sprachmodelle

(10,881)



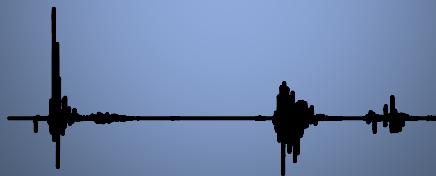
Was ist auf diesem Bild? ►

*Ein roter Würfel mit 4 Augen*

## Text-to-Speech

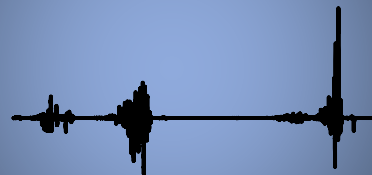
(3,105)

Text to Speech ►



## Speech-to-Text

(24,766)

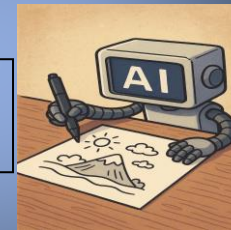


*Speech to Text*

## Bildgeneration

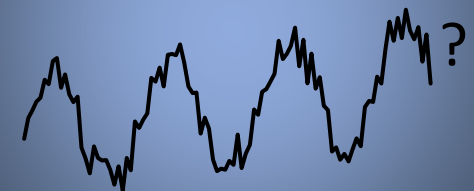
(80,064)

Eine KI die ein Bild zeichnet ►



## Zeitreihenprognose

(112)



( ) Anzahl der Modelle bei Hugging Face (Stand: 28.06.2025), gesamt: 1,825,019

# Foundation Models für die Zeitreihenprognose



License: arXiv.org perpetual non-exclusive license  
arXiv:2310.03589v3 [cs.LG] 27 May 2024

## TimeGPT-1

Azul Garza\*, Cristian Challu\*, Max Mergenthaler-Canseco

Nixtla San Francisco, CA, USA {azul, cristian, max}@nixtla.io

Authors contributed equally.

### Abstract

In this paper, we introduce TimeGPT, the first foundation model for time series, capable of generating accurate predictions for diverse datasets not seen during training. We evaluate our pre-trained model against established statistical, machine learning, and deep learning methods. [Demo](#)

License: arXiv.org perpetual non-exclusive license  
arXiv:2403.07815v3 [cs.LG] 04 Nov 2024

## CHRONOS: Learning the Language of Time Series

Abdul Fatir Ansari<sup>1</sup>, Lorenzo Stella<sup>1</sup>, Caner Turkmen<sup>1</sup>, Xiyuan Zhang<sup>3</sup>, Pedro Mercado<sup>1</sup>, Huibin Shen<sup>1</sup>, Oleksandr Shchur<sup>1</sup>, Syama Sundar Rangapuram<sup>1</sup>, Sebastian Pineda Arango<sup>4,2</sup>, Shubham Kapoor<sup>1</sup>, Jasper Zschiegner<sup>2</sup>, Danielle C. Maddix<sup>1</sup>, Hao Wang<sup>1,5,2</sup>, Michael W. Mahoney<sup>2,6,2</sup>, Kari Torkkola<sup>2</sup>, Andrew Gordon Wilson<sup>2,7,2</sup>, Michael Bohlke-Schneider<sup>1</sup>, Yuyang Wang<sup>1</sup> {ansarnd, stellalo}@amazon.com

<sup>1</sup>AWS AI Labs, <sup>2</sup>Amazon Supply Chain Optimization Technologies, <sup>3</sup>UC San Diego, <sup>4</sup>University of Freiburg, <sup>5</sup>Rutgers University, <sup>6</sup>UC Berkeley, <sup>7</sup>New York University

Equal contribution. Xiyuan Zhang and Sebastian Pineda Arango contributed to this work during their internships at AWS. Hao Wang, Michael W. Mahoney, and Andrew Gordon Wilson hold concurrent appointments at Amazon and their corresponding universities, and this paper describes work performed at Amazon.

### Abstract

We introduce CHRONOS, a simple yet effective framework for pretrained probabilistic time series models. CHRONOS tokenizes time series values using scaling and quantization into a fixed vocabulary and trains existing transformer-based language model architectures on these tokenized time

License: CC BY-NC-SA 4.0  
arXiv:2407.07874v2 [cs.LG] null

## Toto: Time Series Optimized Transformer for Observability Technical Report

Ben Cohen

Emaad Khwaja

Kan Wang

Charles Masson \AND Elise Ramé

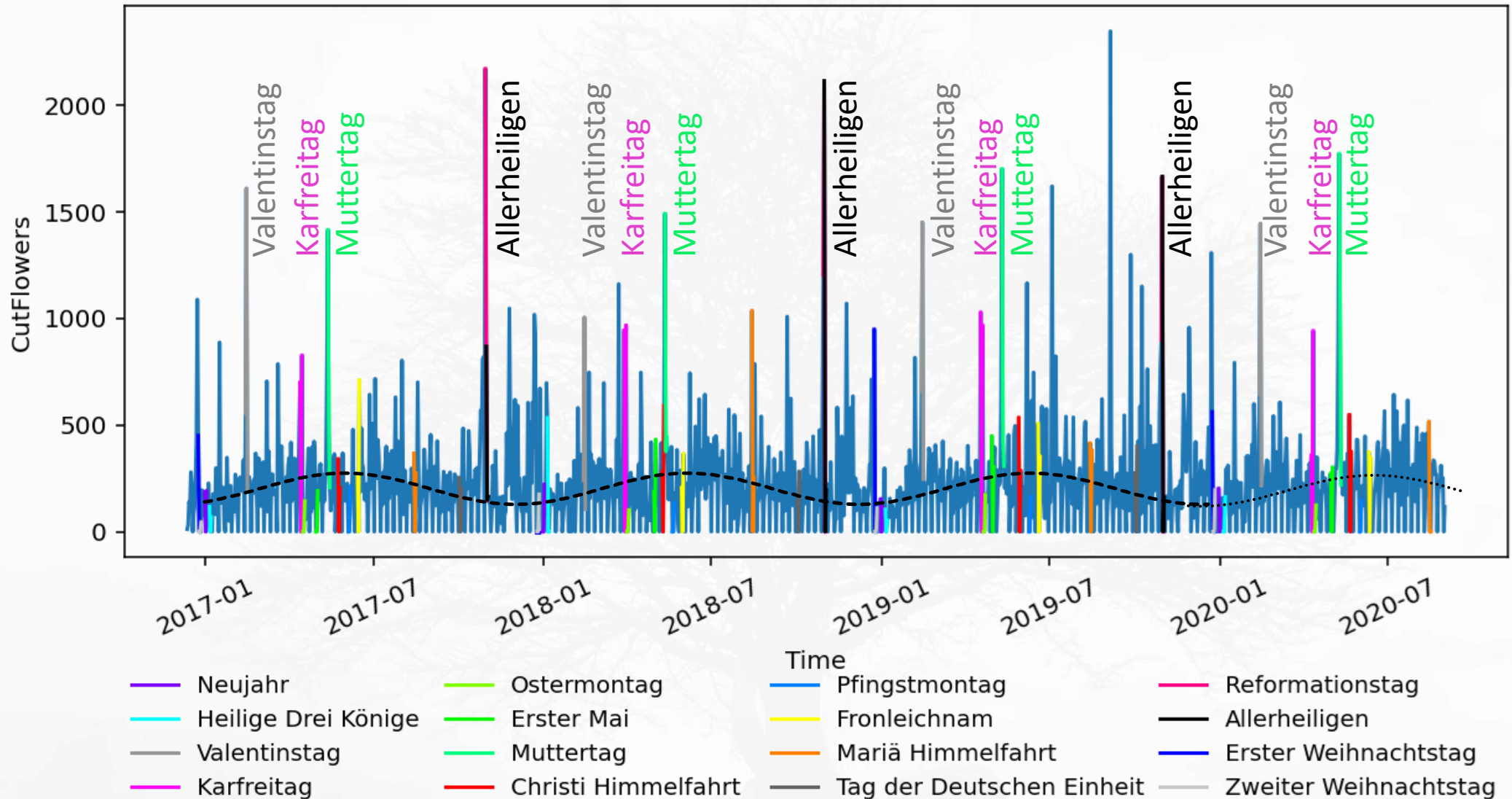
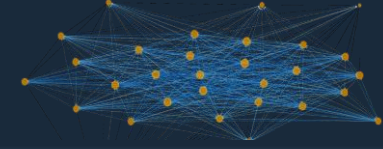
Youssef Doubli

Othmane Abou-Amal \AND {ben.cohen, emaad, kan.wang, charles.masson, elise.rame, youssef.doubli, othmane}@datadoghq.com

### Abstract

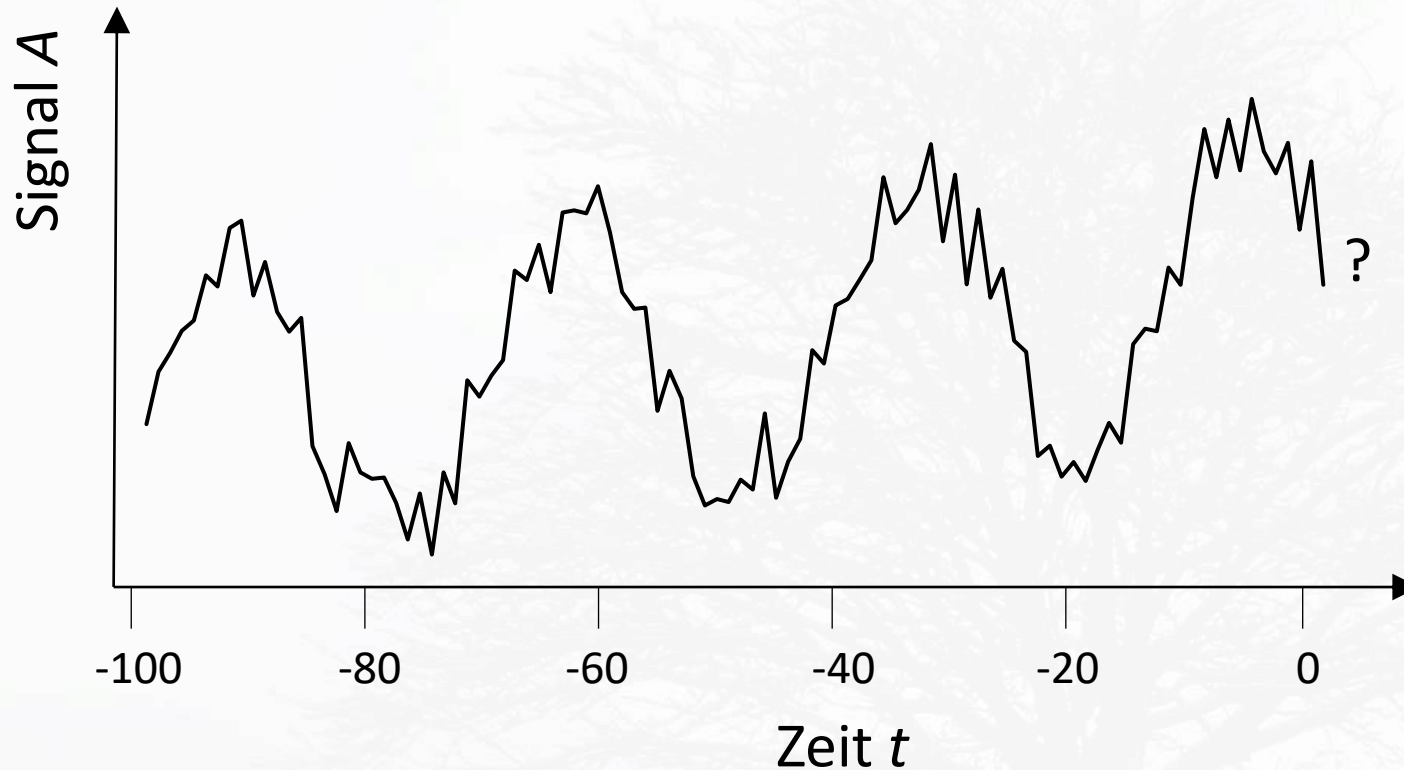
This technical report describes the Time Series Optimized Transformer for Observability (Toto), a new state-of-the-art foundation model for time series forecasting developed by Datadog. In addition to advancing the state of the art on generalized time series benchmarks in domains such as elec-

# Verkaufsdaten des Blumenhändlers



**Machine Learning Outperforms Classical Forecasting on Horticultural Sales Predictions**, Florian Haselbeck, Jennifer Killinger, Klaus Menrad, Thomas Hannus, Dominik G Grimm  
*Machine Learning with Applications* (2022) <https://doi.org/10.1016/j.mlwa.2021.100239>, MIT license

# Datenaufbereitung (Sequenzbildung)



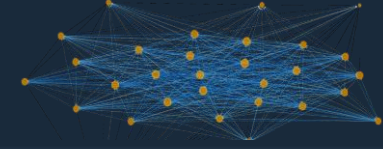
- Modellinput, Merkmale,  $x$
- Zu erwartendes Ergebnis, Label,  $y$
- Anwendungsdaten für die Prognose

# Datenaufbereitung (Sequenzbildung)



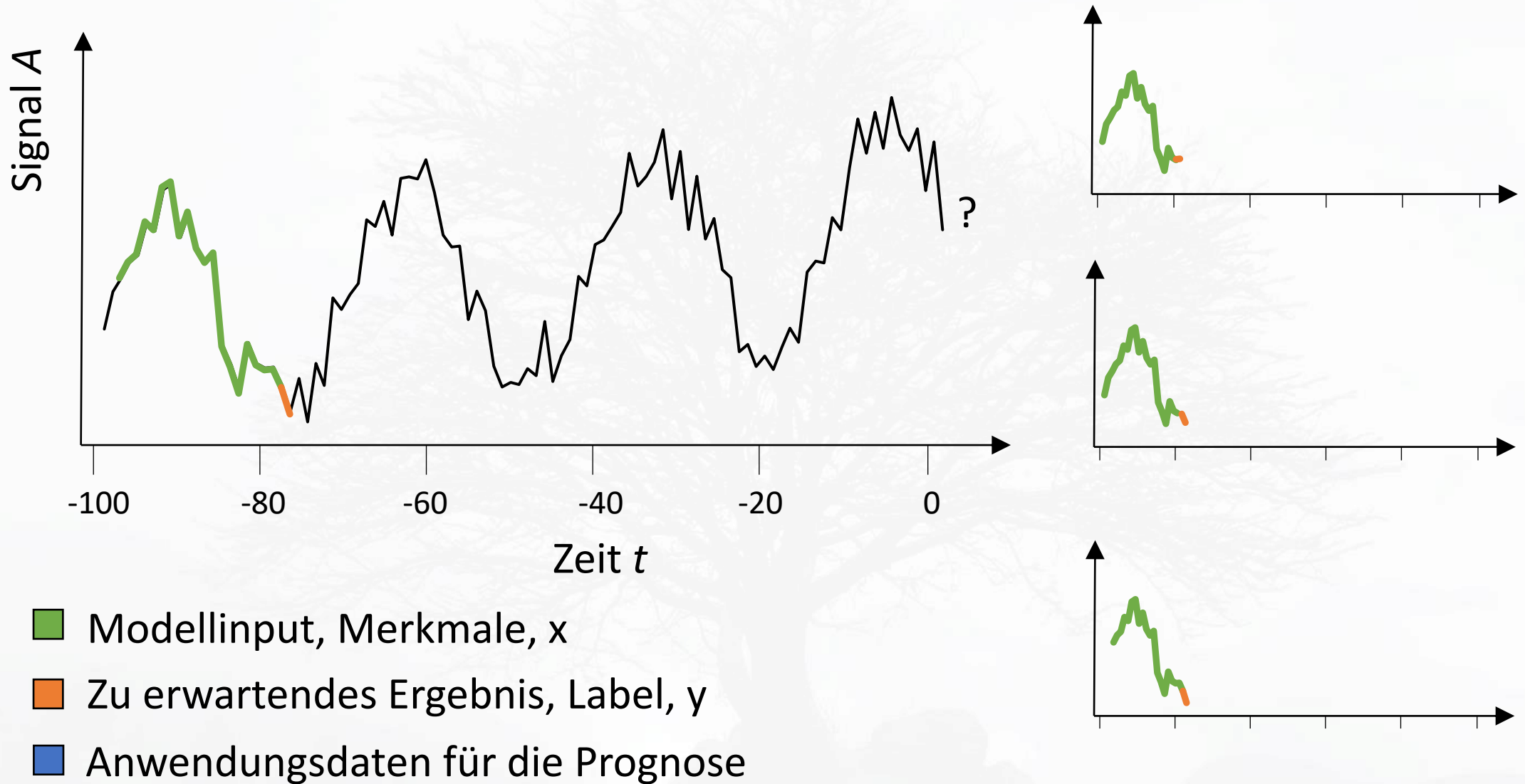
- Modellinput, Merkmale,  $x$
- Zu erwartendes Ergebnis, Label,  $y$
- Anwendungsdaten für die Prognose

# Datenaufbereitung (Sequenzbildung)

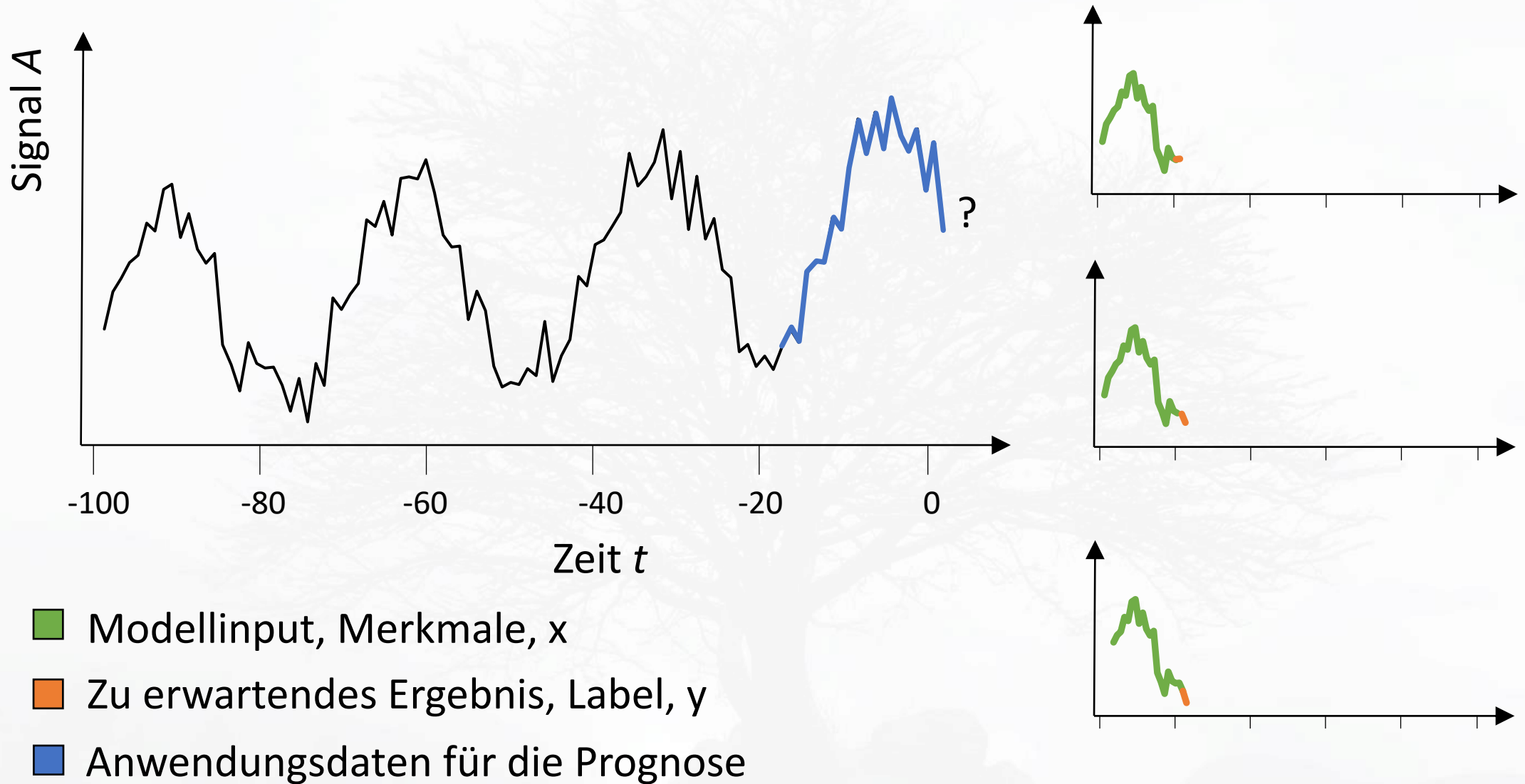


- Modellinput, Merkmale,  $x$
- Zu erwartendes Ergebnis, Label,  $y$
- Anwendungsdaten für die Prognose

# Datenaufbereitung (Sequenzbildung)



# Datenaufbereitung (Sequenzbildung)



# Datenaufbereitung (Externe Merkmale)



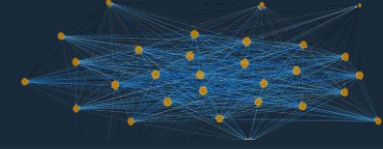
|                                                                                         |            |        |              |           |            |           |             |            |             |               |              |                |             |               |                |               |
|-----------------------------------------------------------------------------------------|------------|--------|--------------|-----------|------------|-----------|-------------|------------|-------------|---------------|--------------|----------------|-------------|---------------|----------------|---------------|
| CashierData (github.com grimmmlab HorticulturalSalesPredictions MIT-Lizenz).csv - Excel |            |        |              |           |            |           |             |            |             |               |              |                |             |               |                |               |
| Suchen                                                                                  |            |        |              |           |            |           |             |            |             |               |              |                |             |               |                |               |
| Anmelden                                                                                |            |        |              |           |            |           |             |            |             |               |              |                |             |               |                |               |
| Teilen                                                                                  |            |        |              |           |            |           |             |            |             |               |              |                |             |               |                |               |
| P113                                                                                    |            |        |              |           |            |           |             |            |             |               |              |                |             |               |                |               |
| no                                                                                      |            |        |              |           |            |           |             |            |             |               |              |                |             |               |                |               |
|                                                                                         | B          | C      | D            | E         | F          | G         | H           | I          | J           | K             | L            | M              | N           | O             | P              | Q             |
| 1                                                                                       | CutFlowers | PotOwn | PotPurchased | Wholesale | FruitsVegs | Commodity | mean_temp   | mean_humid | mean_prec_h | total_prec_he | mean_prec_fl | total_prec_fle | mean_sun_du | total_sun_dur | public_holiday | school_holida |
| 2                                                                                       | 12         | 0      | 0            | 0         | 0          | 0         | 5.20833333  | 89.125     | 0.00833333  | 0.2           | 0.5          | True           | 1.72222222  | 0.68888889    | no             | no            |
| 3                                                                                       | 32         | 2.9    | 68.25        | 0         | 8.4        | 31.9      | 3.84583333  | 93.375     | 0.05833333  | 1.4           | 0.33333333   | True           | 5.5         | 2.2           | no             | no            |
| 4                                                                                       | 140        | 0      | 5            | 0         | 9.35       | 126.55    | 4.67083333  | 99.4166667 | 0.0125      | 0.3           | 0.5          | True           | 0           | 0             | no             | no            |
| 5                                                                                       | 126.9      | 0      | 61           | 0         | 6.9        | 72.7      | 3.57083333  | 90.5416667 | 0           | 0             | 0            | False          | 1.55555556  | 0.62222222    | no             | no            |
| 6                                                                                       | 279.5      | 0      | 59.4         | 0         | 14.3       | 47.35     | -0.34583333 | 82.8333333 | 0           | 0             | 0            | False          | 11.3333333  | 4.53333333    | no             | no            |
| 7                                                                                       | 209        | 4      | 7.9          | 0         | 0          | 103.3     | -1.09166667 | 92.9166667 | 0           | 0             | 0.08333333   | True           | 0           | 0             | no             | no            |
| 8                                                                                       | 0          | 0      | 0            | 0         | 0          | 0         | 0.275       | 97.5       | 0.00833333  | 0.2           | 0.5          | True           | 0           | 0             | no             | no            |
| 9                                                                                       | 83.7       | 0      | 73.5         | 0         | 3.3        | 137.42    | 0.15416667  | 94.5833333 | 0           | 0             | 0.08333333   | True           | 0.72222222  | 0.28888889    | no             | no            |
| 10                                                                                      | 116.4      | 0      | 98.2         | 0         | 42.06      | 153.05    | -3.10416667 | 90         | 0           | 0             | 0            | False          | 17.1666667  | 6.86666667    | no             | no            |
| 11                                                                                      | 257.1      | 0      | 75.3         | 0         | 18.7       | 123.89    | -1.52083333 | 93.125     | 0           | 0             | 0            | False          | 0           | 0             | no             | no            |
| 12                                                                                      | 439.2      | 0      | 83.1         | 0         | 7.7        | 145.57    | -0.89166667 | 91.5833333 | 0.05833333  | 1.4           | 0.20833333   | True           | 0           | 0             | no             | no            |
| 13                                                                                      | 1088.55    | 0      | 203.8        | 0         | 38.89      | 126.7     | 0.75833333  | 99.3333333 | 0.01666667  | 0.4           | 0.5          | True           | 0           | 0             | no             | no            |
| 14                                                                                      | 450.9      | 0      | 75.2         | 0         | 25.6       | 51.95     | 4.9375      | 89.5       | 0.03333333  | 0.8           | 0.33333333   | True           | 1.72222222  | 0.68888889    | no             | no            |
| 15                                                                                      | 0          | 0      | 0            | 0         | 0          | 0         | 6.82916667  | 89.7916667 | 0.02083333  | 0.5           | 0.91666667   | True           | 0           | 0             | Erster Weihn   | no            |
| 16                                                                                      | 0          | 0      | 0            | 0         | 0          | 0         | 6.6625      | 87.5416667 | 0.00416667  | 0.1           | 0.5          | True           | 0           | 0             | Zweiter Weihn  | no            |
| 17                                                                                      | 45         | 0      | 38.7         | 0         | 5.88       | 25.4      | 4.20833333  | 77.875     | 0           | 0             | 0.625        | True           | 0           | 0             | no             | no            |
| 18                                                                                      | 196        | 54.9   | 63.6         | 0         | 0.9        | 55.45     | 4.1375      | 84.5833333 | 0           | 0             | 0.25         | True           | 0           | 0             | no             | no            |
| 19                                                                                      | 105        | 0      | 125.5        | 0         | 10.05      | 34.9      | -0.08333333 | 92.75      | 0           | 0             | 0            | False          | 21.1111111  | 8.44444444    | no             | no            |
| 20                                                                                      | 77.7       | 3      | 23           | 0         | 15.5       | 18.8      | -3.6125     | 96.5833333 | 0           | 0             | 0.08333333   | True           | 11.6111111  | 4.64444444    | no             | no            |
| 21                                                                                      | 186.6      | 0      | 0            | 0         | 26.75      | 42.6      | -6.77916667 | 92.2083333 | 0           | 0             | 0            | False          | 22          | 8.8           | no             | no            |
| 22                                                                                      | 0          | 0      | 0            | 0         | 0          | 0         | -6.67916667 | 94.75      | 0           | 0             | 0            | False          | 0           | 0             | Neujahr        | no            |
| 23                                                                                      | 190.5      | 2.5    | 3.9          | 0         | 5.1        | 51.28     | -1.50416667 | 95.7916667 | 0.1         | 2.4           | 0.5          | True           | 0           | 0             | no             | no            |
| 24                                                                                      | 27         | 5      | 3.9          | 0         | 9.8        | 31.44     | -0.47916667 | 92.75      | 0.03333333  | 0.8           | 0.33333333   | True           | 0           | 0             | no             | no            |
| 25                                                                                      | 227.6      | 17.5   | 27.9         | 0         | 21.45      | 54.8      | 1.075       | 87.7916667 | 0.0625      | 1.5           | 0.91666667   | True           | 0.5         | 0.2           | no             | no            |
| 26                                                                                      | 117.91     | 29.9   | 21.9         | 0         | 14.8       | 26.49     | -3.9125     | 75.6666667 | 0.02083333  | 0.5           | 0.25         | True           | 8.55555556  | 3.42222222    | no             | no            |
| 27                                                                                      | 0          | 0      | 0            | 0         | 0          | 0         | -12.9416667 | 84.1666667 | 0           | 0             | 0            | False          | 23.6666667  | 9.46666667    | Heilige Drei K | no            |
| 28                                                                                      | 68.1       | 3.9    | 6.9          | 0         | 14.6       | 51.65     | -10.1083333 | 82.7916667 | 0.05416667  | 1.3           | 0.33333333   | True           | 0.5         | 0.2           | no             | no            |
| 29                                                                                      | 0          | 0      | 0            | 0         | 0          | 0         | -2.79583333 | 94.2916667 | 0.03333333  | 0.8           | 0.54166667   | True           | 0           | 0             | no             | no            |
| 30                                                                                      | 177.7      | 2.5    | 113.6        | 0         | 2.15       | 31.23     | -0.80416667 | 94.6666667 | 0           | 0             | 0.625        | True           | 0           | 0             | no             | no            |
| 31                                                                                      | 105.5      | 3.4    | 19.5         | 0         | 23.8       | 8.79      | -0.9625     | 74.9583333 | 0           | 0             | 0.125        | True           | 12.5        | 5             | no             | no            |
| 32                                                                                      | 81         | 8      | 91.4         | 0         | 20.1       | 17.7      | -0.62916667 | 88         | 0.12916667  | 3.1           | 0.625        | True           | 0.55555556  | 0.22222222    | no             | no            |
| 33                                                                                      | 299        | 9.9    | 18.9         | 0         | 2.8        | 38.39     | 3.2875      | 81.375     | 0.175       | 4.2           | 0.375        | True           | 0           | 0             | no             | no            |
| 34                                                                                      | 295.2      | 2.8    | 28.9         | 0         | 35.58      | 17.24     | 2.1125      | 79         | 0.1375      | 3.3           | 0.75         | True           | 0           | 0             | no             | no            |

1. Ergänzen von public\_holiday mit Valentinstag und Muttertag

2. Entfernen von anderen potentiellen Verkaufszahlen wie PotOwn, PotPurchased ...

3. Entfernen von redundanten Merkmalen wie mean\_prec\_height

# Datenaufbereitung (Externe Merkmale)



|    | B          | C      | D           | E             | F             | G                     | H |
|----|------------|--------|-------------|---------------|---------------|-----------------------|---|
| 1  | CutFlowers | PotOwn | mean_temp   | total_prec_he | total_sun_dur | public_holiday        |   |
| 2  | 12         | 0      | 5.20833333  | 0.2           | 0.68888889    | no                    |   |
| 3  | 32         | 2.9    | 3.84583333  | 1.4           | 2.2           | no                    |   |
| 4  | 140        | 0      | 4.67083333  | 0.3           | 0             | no                    |   |
| 5  | 126.9      | 0      | 3.57083333  | 0             | 0.62222222    | no                    |   |
| 6  | 279.5      | 0      | -0.34583333 | 0             | 4.53333333    | no                    |   |
| 7  | 209        | 4      | -1.09166667 | 0             | 0             | no                    |   |
| 8  | 0          | 0      | 0.275       | 0.2           | 0             | no                    |   |
| 9  | 83.7       | 0      | 0.15416667  | 0             | 0.28888889    | no                    |   |
| 10 | 116.4      | 0      | -3.10416667 | 0             | 6.86666667    | no                    |   |
| 11 | 257.1      | 0      | -1.52083333 | 0             | 0             | no                    |   |
| 12 | 439.2      | 0      | -0.89166667 | 1.4           | 0             | no                    |   |
| 13 | 1088.55    | 0      | 0.75833333  | 0.4           | 0             | no                    |   |
| 14 | 450.9      | 0      | 4.9375      | 0.8           | 0.68888889    | no                    |   |
| 15 | 0          | 0      | 6.82916667  | 0.5           | 0             | Erster Weihnachtstag  |   |
| 16 | 0          | 0      | 6.6625      | 0.1           | 0             | Zweiter Weihnachtstag |   |
| 17 | 45         | 0      | 4.20833333  | 0             | 0             | no                    |   |
| 18 | 196        | 54.9   | 4.1375      | 0             | 0             | no                    |   |
| 19 | 196        | 0      | -0.08333333 | 0             | 8.44444444    | no                    |   |
| 20 | 196        | 3      | -3.6125     | 0             | 4.64444444    | no                    |   |
| 21 | 196        | 0      | -6.77916667 | 0             | 8.8           | no                    |   |
| 22 | 0          | 0      | -6.67916667 | 0             | 0             | Neujahr               |   |
| 23 | 190.5      | 2.5    | -1.50416667 | 2.4           | 0             | no                    |   |
| 24 | 27         | 5      | -0.47916667 | 0.8           | 0             | no                    |   |
| 25 | 227.6      | 17.5   | 1.075       | 1.5           | 0.2           | no                    |   |
| 26 | 117.91     | 29.9   | -3.9125     | 0.5           | 3.42222222    | no                    |   |
| 27 | 0          | 0      | -12.9416667 | 0             | 9.46666667    | Heilige Drei Könige   |   |
| 28 | 68.1       | 3.9    | -10.1083333 | 1.3           | 0.2           | no                    |   |
| 29 | 0          | 0      | -2.79583333 | 0.8           | 0             | no                    |   |
| 30 | 177.7      | 2.5    | -0.80416667 | 0             | 0             | no                    |   |
| 31 | 105.5      | 3.4    | -0.9625     | 0             | 5             | no                    |   |
| 32 | 81         | 8      | -0.62916667 | 3.1           | 0.22222222    | no                    |   |
| 33 | 299        | 9.9    | 3.2875      | 4.2           | 0             | no                    |   |
| 34 | 295.2      | 2.8    | 2.1125      | 3.3           | 0             | no                    |   |

4. Monat und Wochentag aus dem Datum extrahieren

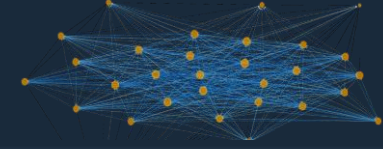
5. Die Spalten public\_holiday, Monat und Wochentag durch Onehot-Encoding aufbereiten

6. Alle Daten spaltenweise minmax-skalisieren (auf 0 bis 1)

7. Sequenzen aus den Daten bilden

- Für das Training von Modellen brauchen wir mehrere statistische Daten
- Über eine Sequenz (Länge = 15) wird die Vergangenheit betrachtet um daraus die Zukunft (Länge = 5) zu prognostizieren
- Problem: Blumen werden vor den Feiertage verkauft
- Trick: Feiertage sind lange im voraus bekannt, wir können zukünftige Werte übergeben

# Datenaufbereitung (Externe Merkmale)



|    | B          | C      | D           | E             | F             | G                     | H |
|----|------------|--------|-------------|---------------|---------------|-----------------------|---|
| 1  | CutFlowers | PotOwn | mean_temp   | total_prec_he | total_sun_dur | public_holiday        |   |
| 2  | 12         | 0      | 5.20833333  | 0.2           | 0.68888889    | no                    |   |
| 3  | 32         | 2.9    | 3.84583333  | 1.4           | 2.2           | no                    |   |
| 4  | 140        | 0      | 4.67083333  | 0.3           | 0             | no                    |   |
| 5  | 126.9      | 0      | 3.57083333  | 0             | 0.62222222    | no                    |   |
| 6  | 279.5      | 0      | -0.34583333 | 0             | 4.53333333    | no                    |   |
| 7  | 209        | 4      | -1.09166667 | 0             | 0             | no                    |   |
| 8  | 0          | 0      | 0.275       | 0.2           | 0             | no                    |   |
| 9  | 83.7       | 0      | 0.15416667  | 0             | 0.28888889    | no                    |   |
| 10 | 116.4      | 0      | -3.10416667 | 0             | 6.86666667    | no                    |   |
| 11 | 257.1      | 0      | -1.52083333 | 0             | 0             | no                    |   |
| 12 | 439.2      | 0      | -0.89166667 | 1.4           | 0             | no                    |   |
| 13 | 1088.55    | 0      | 0.75833333  | 0.4           | 0             | no                    |   |
| 14 | 450.9      | 0      | 4.9375      | 0.8           | 0.68888889    | no                    |   |
| 15 | 0          | 0      | 6.82916667  | 0.5           | 0             | Erster Weihnachtstag  |   |
| 16 | 0          | 0      | 6.6625      | 0.1           | 0             | Zweiter Weihnachtstag |   |
| 17 | 45         | 0      | 4.20833333  | 0             | 0             | no                    |   |
| 18 | 196        | 54.9   | 4.1375      | 0             | 0             | no                    |   |
| 19 | 105        | 0      | -0.08333333 | 0             | 8.44444444    | no                    |   |
| 20 | 77.7       | 3      | -3.6125     | 0             | 4.64444444    | no                    |   |
| 21 | 186.6      | 0      | -6.77916667 | 0             | 8.8           | no                    |   |
| 22 | 0          | 0      | -6.67916667 | 0             | 0             | Neujahr               |   |
| 23 | 190.5      | 2.5    | -1.50416667 | 2.4           | 0             | no                    |   |
| 24 | 27         | 5      | -0.47916667 | 0.8           | 0             | no                    |   |
| 25 | 227.6      | 17.5   | 1.075       | 1.5           | 0.2           | no                    |   |
| 26 | 117.91     | 29.9   | -3.9125     | 0.5           | 3.42222222    | no                    |   |
| 27 | 0          | 0      | -12.9416667 | 0             | 9.46666667    | Heilige Drei Könige   |   |
| 28 | 68.1       | 3.9    | -10.1083333 | 1.3           | 0.2           | no                    |   |
| 29 | 0          | 0      | -2.79583333 | 0.8           | 0             | no                    |   |
| 30 | 177.7      | 2.5    | -0.80416667 | 0             | 0             | no                    |   |
| 31 | 105.5      | 3.4    | -0.9625     | 0             | 5             | no                    |   |
| 32 | 81         | 8      | -0.62916667 | 3.1           | 0.22222222    | no                    |   |
| 33 | 299        | 9.9    | 3.2875      | 4.2           | 0             | no                    |   |
| 34 | 295.2      | 2.8    | 2.1125      | 3.3           | 0             | no                    |   |

4. Monat und Wochentag aus dem Datum extrahieren

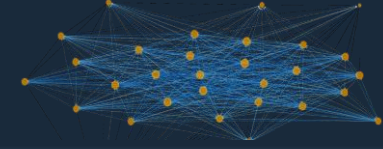
5. Die Spalten public\_holiday, Monat und Wochentag durch Onehot-Encoding aufbereiten

6. Alle Daten spaltenweise minmax-skalisieren (auf 0 bis 1)

7. Sequenzen aus den Daten bilden

- Für das Training von Modellen brauchen wir mehrere statistische Daten
- Über eine Sequenz (Länge = 15) wird die Vergangenheit betrachtet um daraus die Zukunft (Länge = 5) zu prognostizieren
- Problem: Blumen werden vor den Feiertage verkauft
- Trick: Feiertage sind lange im voraus bekannt, wir können zukünftige Werte übergeben

# Datenaufbereitung (Externe Merkmale)



|    | B          | C      | D           | E             | F             | G                     | H |
|----|------------|--------|-------------|---------------|---------------|-----------------------|---|
| 1  | CutFlowers | PotOwn | mean_temp   | total_prec_he | total_sun_dur | public_holiday        |   |
| 2  | 12         | 0      | 5.20833333  | 0.2           | 0.68888889    | no                    |   |
| 3  | 32         | 2.9    | 3.84583333  | 1.4           | 2.2           | no                    |   |
| 4  | 140        | 0      | 4.67083333  | 0.3           | 0             | no                    |   |
| 5  | 126.9      | 0      | 3.57083333  | 0             | 0.62222222    | no                    |   |
| 6  | 279.5      | 0      | -0.34583333 | 0             | 4.53333333    | no                    |   |
| 7  | 209        | 4      | -1.09166667 | 0             | 0             | no                    |   |
| 8  | 0          | 0      | 0.275       | 0.2           | 0             | no                    |   |
| 9  | 83.7       | 0      | 0.15416667  | 0             | 0.28888889    | no                    |   |
| 10 | 116.4      | 0      | -3.10416667 | 0             | 6.86666667    | no                    |   |
| 11 | 257.1      | 0      | -1.52083333 | 0             | 0             | no                    |   |
| 12 | 439.2      | 0      | -0.89166667 | 1.4           | 0             | no                    |   |
| 13 | 1088.55    | 0      | 0.75833333  | 0.4           | 0             | no                    |   |
| 14 | 450.9      | 0      | 4.9375      | 0.8           | 0.68888889    | no                    |   |
| 15 | 0          | 0      | 6.82916667  | 0.5           | 0             | Erster Weihnachtstag  |   |
| 16 | 0          | 0      | 6.6625      | 0.1           | 0             | Zweiter Weihnachtstag |   |
| 17 | 45         | 0      | 4.20833333  | 0             | 0             | no                    |   |
| 18 | 196        | 54.9   | 4.1375      | 0             | 0             | no                    |   |
| 19 | 105        | 0      | -0.08333333 | 0             | 8.44444444    | no                    |   |
| 20 | 77.7       | 3      | -3.6125     | 0             | 4.64444444    | no                    |   |
| 21 | 186.6      | 0      | -6.77916667 | 0             | 8.8           | no                    |   |
| 22 | 0          | 0      | -6.67916667 | 0             | 0             | Neujahr               |   |
| 23 | 190.5      | 2.5    | -1.50416667 | 2.4           | 0             | no                    |   |
| 24 | 27         | 5      | -0.47916667 | 0.8           | 0             | no                    |   |
| 25 | 227.6      | 17.5   | 1.075       | 1.5           | 0.2           | no                    |   |
| 26 | 117.91     | 29.9   | -3.9125     | 0.5           | 3.42222222    | no                    |   |
| 27 | 0          | 0      | -12.9416667 | 0             | 9.46666667    | Heilige Drei Könige   |   |
| 28 | 68.1       | 3.9    | -10.1083333 | 1.3           | 0.2           | no                    |   |
| 29 | 0          | 0      | -2.79583333 | 0.8           | 0             | no                    |   |
| 30 | 177.7      | 2.5    | -0.80416667 | 0             | 0             | no                    |   |
| 31 | 105.5      | 3.4    | -0.9625     | 0             | 5             | no                    |   |
| 32 | 81         | 8      | -0.62916667 | 3.1           | 0.22222222    | no                    |   |
| 33 | 299        | 9.9    | 3.2875      | 4.2           | 0             | no                    |   |
| 34 | 295.2      | 2.8    | 2.1125      | 3.3           | 0             | no                    |   |

4. Monat und Wochentag aus dem Datum extrahieren

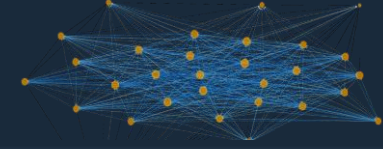
5. Die Spalten public\_holiday, Monat und Wochentag durch Onehot-Encoding aufbereiten

6. Alle Daten spaltenweise minmax-skalisieren (auf 0 bis 1)

7. Sequenzen aus den Daten bilden

- Für das Training von Modellen brauchen wir mehrere statistische Daten
- Über eine Sequenz (Länge = 15) wird die Vergangenheit betrachtet um daraus die Zukunft (Länge = 5) zu prognostizieren
- Problem: Blumen werden vor den Feiertage verkauft
- Trick: Feiertage sind lange im voraus bekannt, wir können zukünftige Werte übergeben

# Datenaufbereitung (Externe Merkmale)



|    | B          | C      | D           | E             | F             | G                     | H |
|----|------------|--------|-------------|---------------|---------------|-----------------------|---|
| 1  | CutFlowers | PotOwn | mean_temp   | total_prec_he | total_sun_dur | public_holiday        |   |
| 2  | 12         | 0      | 5.20833333  | 0.2           | 0.68888889    | no                    |   |
| 3  | 32         | 2.9    | 3.84583333  | 1.4           | 2.2           | no                    |   |
| 4  | 140        | 0      | 4.67083333  | 0.3           | 0             | no                    |   |
| 5  | 126.9      | 0      | 3.57083333  | 0             | 0.62222222    | no                    |   |
| 6  | 279.5      | 0      | -0.34583333 | 0             | 4.53333333    | no                    |   |
| 7  | 209        | 4      | -1.09166667 | 0             | 0             | no                    |   |
| 8  | 0          | 0      | 0.275       | 0.2           | 0             | no                    |   |
| 9  | 83.7       | 0      | 0.15416667  | 0             | 0.28888889    | no                    |   |
| 10 | 116.4      | 0      | -3.10416667 | 0             | 6.86666667    | no                    |   |
| 11 | 257.1      | 0      | -1.52083333 | 0             | 0             | no                    |   |
| 12 | 439.2      | 0      | -0.89166667 | 1.4           | 0             | no                    |   |
| 13 | 1088.55    | 0      | 0.75833333  | 0.4           | 0             | no                    |   |
| 14 | 450.9      | 0      | 4.9375      | 0.8           | 0.68888889    | no                    |   |
| 15 | 0          | 0      | 6.82916667  | 0.5           | 0             | Erster Weihnachtstag  |   |
| 16 | 0          | 0      | 6.6625      | 0.1           | 0             | Zweiter Weihnachtstag |   |
| 17 | 45         | 0      | 4.20833333  | 0             | 0             | no                    |   |
| 18 | 196        | 54.9   | 4.1375      | 0             | 0             | no                    |   |
| 19 | 105        | 0      | -0.08333333 | 0             | 8.44444444    | no                    |   |
| 20 | 77.7       | 3      | -3.6125     | 0             | 4.64444444    | no                    |   |
| 21 | 186.6      | 0      | -6.77916667 | 0             | 8.8           | no                    |   |
| 22 | 0          | 0      | -6.67916667 | 0             | 0             | Neujahr               |   |
| 23 | 190.5      | 2.5    | -1.50416667 | 2.4           | 0             | no                    |   |
| 24 | 27         | 5      | -0.47916667 | 0.8           | 0             | no                    |   |
| 25 | 227.6      | 17.5   | 1.075       | 1.5           | 0.2           | no                    |   |
| 26 | 117.91     | 29.9   | -3.9125     | 0.5           | 3.42222222    | no                    |   |
| 27 | 0          | 0      | -12.9416667 | 0             | 9.46666667    | Heilige Drei Könige   |   |
| 28 | 68.1       | 3.9    | -10.1083333 | 1.3           | 0.2           | no                    |   |
| 29 | 0          | 0      | -2.79583333 | 0.8           | 0             | no                    |   |
| 30 | 177.7      | 2.5    | -0.80416667 | 0             | 0             | no                    |   |
| 31 | 105.5      | 3.4    | -0.9625     | 0             | 5             | no                    |   |
| 32 | 81         | 8      | -0.62916667 | 3.1           | 0.22222222    | no                    |   |
| 33 | 299        | 9.9    | 3.2875      | 4.2           | 0             | no                    |   |
| 34 | 295.2      | 2.8    | 2.1125      | 3.3           | 0             | no                    |   |

4. Monat und Wochentag aus dem Datum extrahieren

5. Die Spalten public\_holiday, Monat und Wochentag durch Onehot-Encoding aufbereiten

6. Alle Daten spaltenweise minmax-skalisieren (auf 0 bis 1)

7. Sequenzen aus den Daten bilden

- Für das Training von Modellen brauchen wir mehrere statistische Daten
- Über eine Sequenz (Länge = 15) wird die Vergangenheit betrachtet um daraus die Zukunft (Länge = 5) zu prognostizieren
- Problem: Blumen werden vor den Feiertage verkauft
- Trick: Feiertage sind lange im voraus bekannt, wir können zukünftige Werte übergeben

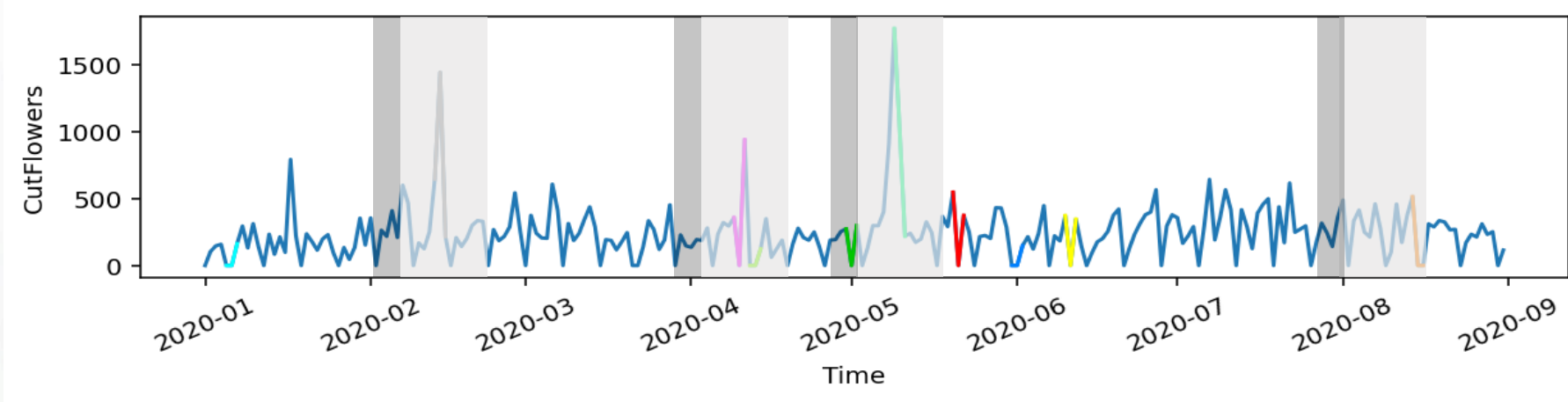
# Auswahl von Testdaten



Jede Software muss vor ihrer Anwendung getestet werden. KI-Modelle neigen dazu, ihre Trainingsdaten gut zu prognostizieren auf neuen Daten aber zu scheitern

Wir wählen aus dem Jahr 2020 jeweils zwei Wochen um den Valentinstag, Ostern, Muttertag und Anfang August (unten hell grau markiert) und erhalten damit  $4 \times 14 = 56$  Sequenzen von denen eine Hälfte zur Validierung (Hyperparameter-Tuning) und die andere Hälfte zum Testen verwendet wird

Bei der Anwendung des Modells prognostizieren wir die Zukunft (die nächsten 5 Tage) auf Basis der Vergangenheit (der letzten 15 Tage). Die Anwendungsdaten sind daher um die Prognoselänge verschieden zu den Trainingsdaten, daher entfernen wir vor der Testphase 5 Trainingssequenzen (dunkelgrau markiert)



# Regression vs. Klassifikation



Klassifikation

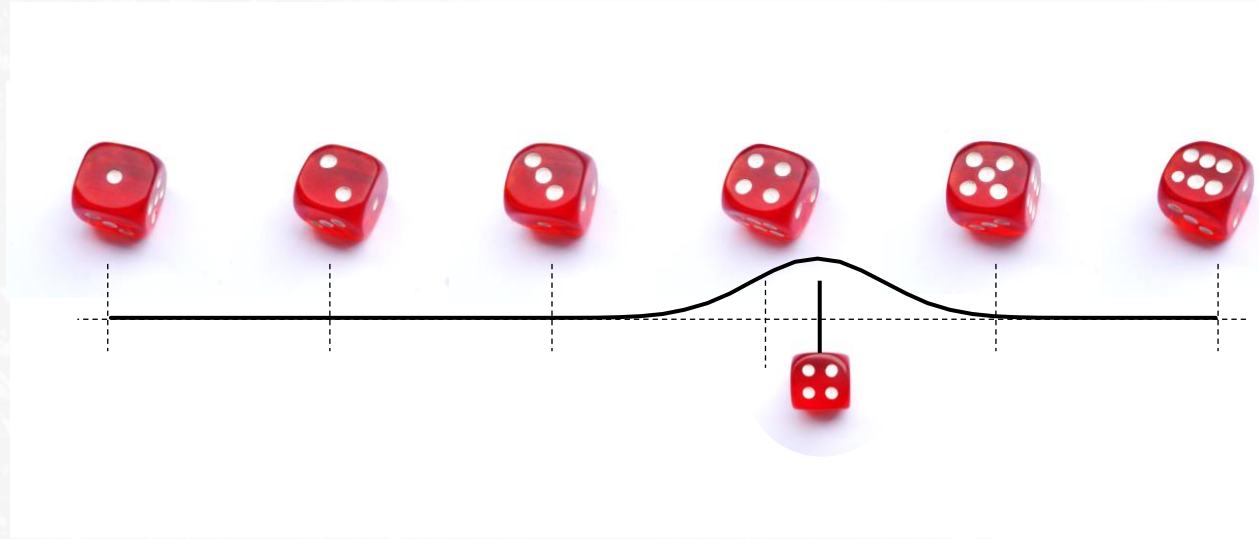


Labels sind diskret und unsortiert

Metriken:

- Accuracy, Optimum: 1
- Recall, Optimum:
- Precision, Optimum: 1

Regression



Labels sind sortiert

Metriken:

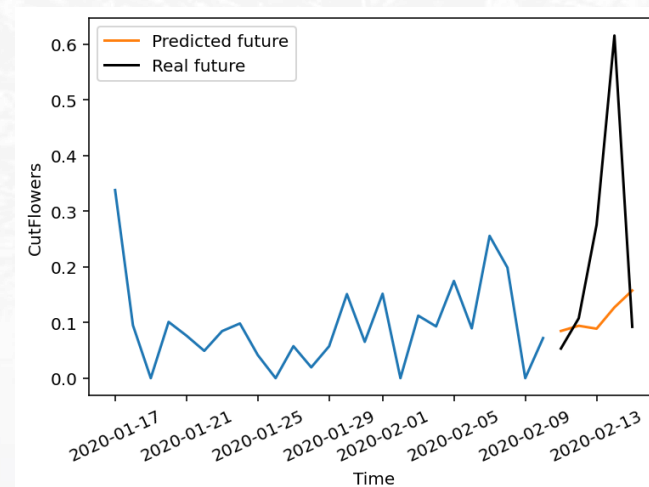
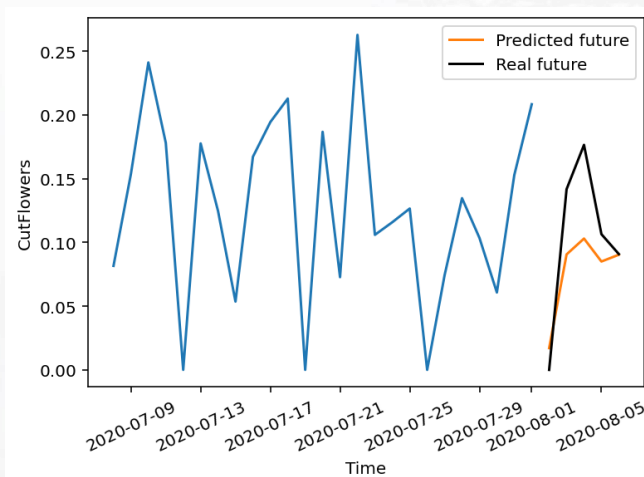
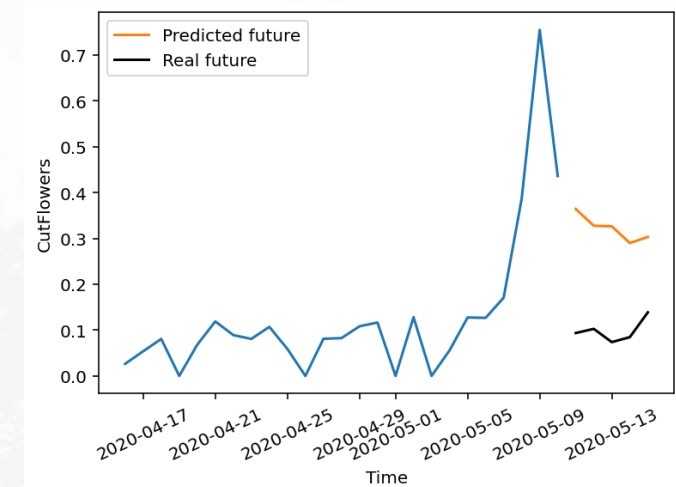
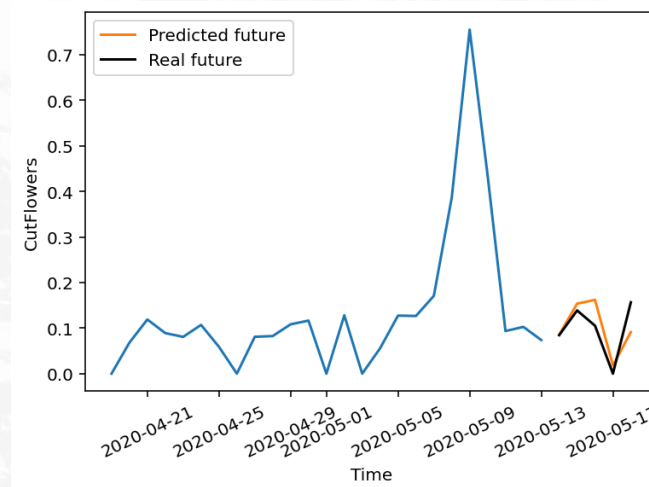
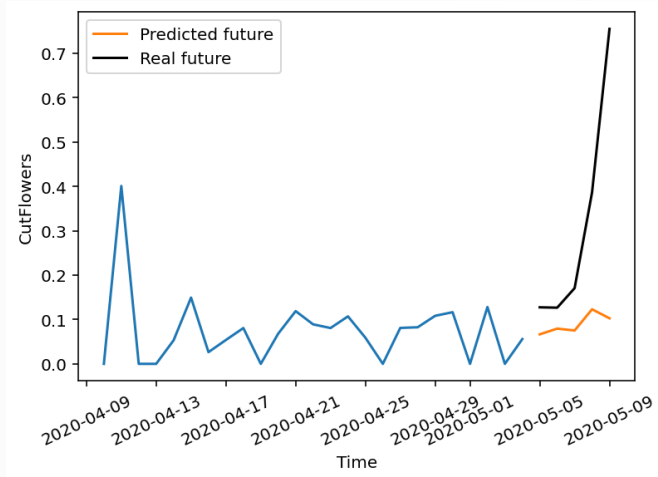
- Mittlerer Absoluter Fehler (MAE), Optimum: 0
- Mittlerer Quadrierter Fehler (MSE), Optimum: 0
- Bestimmtheitsmaß ( $R^2$ -Score), Optimum: 1



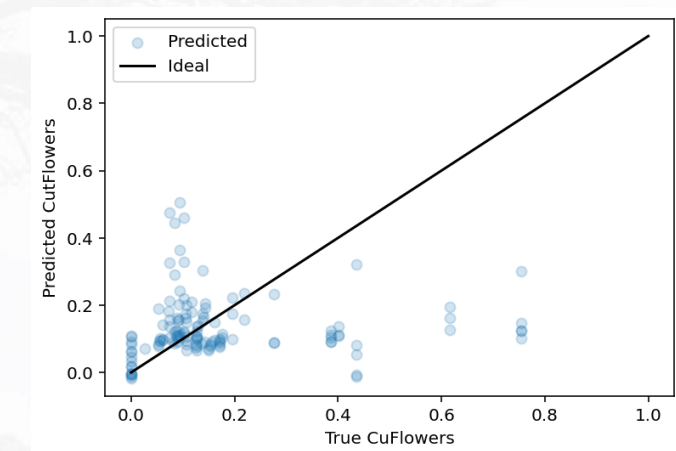
```
from nixtla import NixtlaClient
nixtla_client = NixtlaClient(api_key=api_key)    # Definition und
nixtla_client.validate_api_key()                # Authentifizierung eines api_keys

y_pred = nixtla_client.forecast(
    df=data_past,                               # Merkmale inklusive CutFlowers in der Vergangenheit
    X_df=x_future,                             # Merkmale (die nächsten Feiertage) in der Zukunft
    h=forecast_length,                         # Die Prognoselänge
    freq='D',                                  # Das Intervall der Zeitpunkte (täglich)
    time_col='Date',                           # Zeitspalte
    target_col='CutFlowers',                   # Vorherzusagende Zielspalte
    hist_exog_list=['mean_temp', 'total_prec_height_mm', 'total_sun_dur_h'], # Merkmale in data_past die
                                                    # nicht x_future bekannt sind
)
```

# TimeGPT Ergebnisse



Bestimmtheitsmaß: -0.279  
Mittlerer Absoluter Fehler: 0.124





```
from autogluon.timeseries import TimeSeriesPredictor, TimeSeriesDataFrame

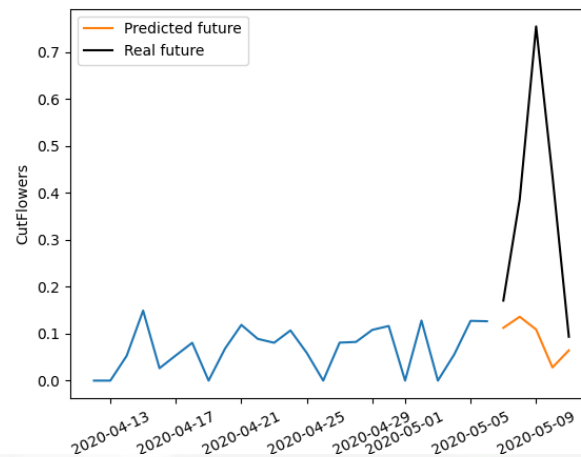
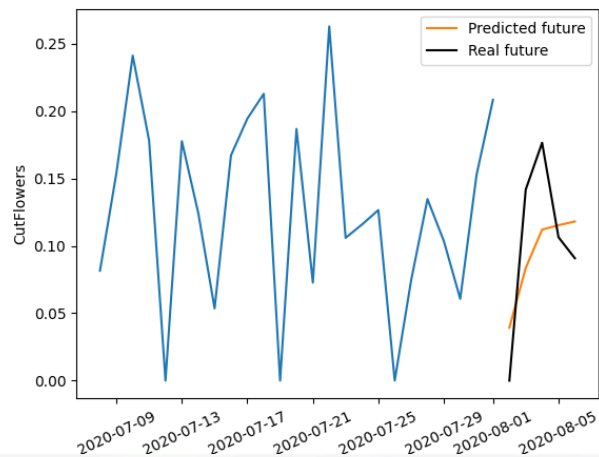
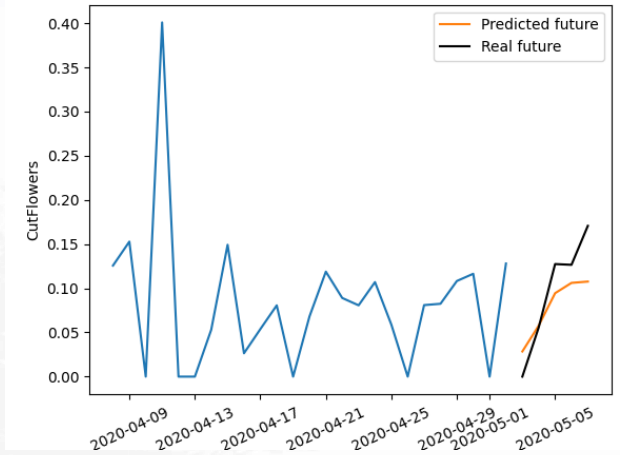
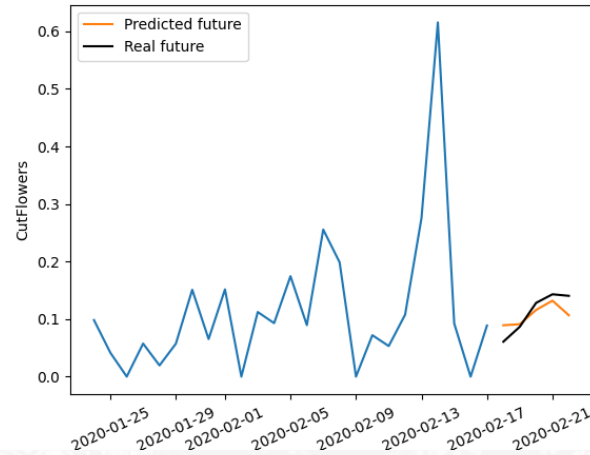
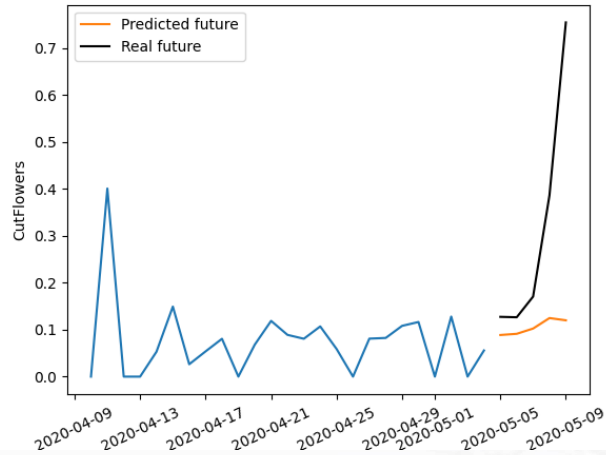
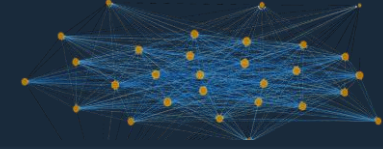
data=TimeSeriesDataFrame(data)          # TimeSeriesDataFrame mit mehreren Sequenzen und Timestamps

predictor = TimeSeriesPredictor(        # Definition eines Predictors
    prediction_length=forecast_length,  # Die Prognoselänge
    target="CutFlowers"                 # Vorherzusagende Zielspalte
)

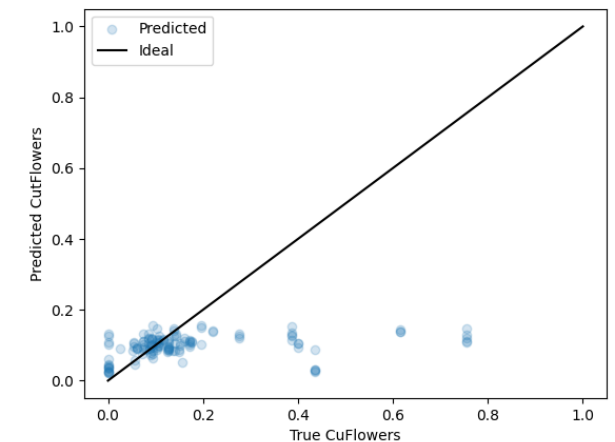
predictor.fit(                          # Ein Chronos-Modell auf die Daten neu fitten
    data,
    hyperparameters={"Chronos": {"model_path": "amazon/chronos-bolt-base"}, },
)

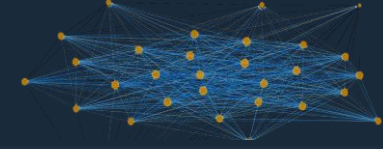
y_pred = predictor.predict(test_data)   # Das gefittete Modell auf neue Testdaten anwenden
```

# Chronos Ergebnisse



Bestimmtheitsmaß: -0.089  
Mittlerer Absoluter Fehler: 0.099





```
!git clone https://github.com/DataDog/toto.git /kaggle/working/toto  
!pip requirements.txt
```

*# Git Repository Klonen*  
*# Nötige Bibliotheken installieren*

```
from dataset import MaskedTimeseries  
from inference.forecaster import TotoForecaster  
from model.toto import Toto
```

```
toto = Toto.from_pretrained('Datadog/Toto-Open-Base-1.0')  
toto.compile()  
forecaster = TotoForecaster(toto.model)
```

*# Ein vortrainiertes Modell wählen*  
*# Das Modell kompilieren*  
*# Einen Forecaster definieren*

```
inputs = MaskedTimeseries(  
    series=data,  
    time_interval_seconds=60*60*24,  
)
```

*# Die Input-Daten vorbereiten*  
*# Das Zeitintervall der Daten angeben*

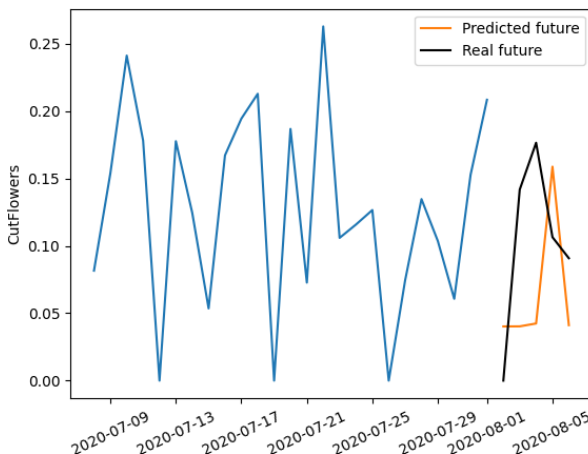
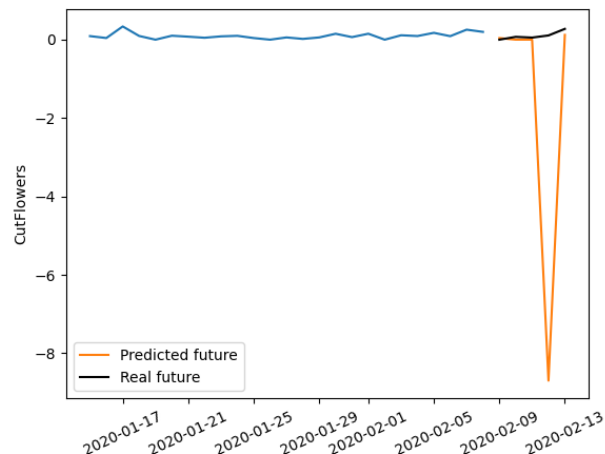
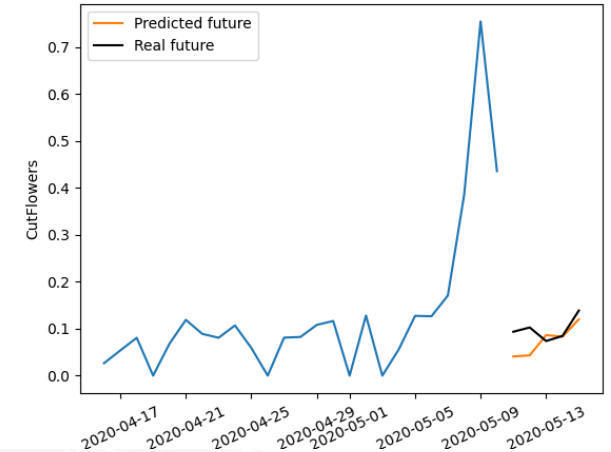
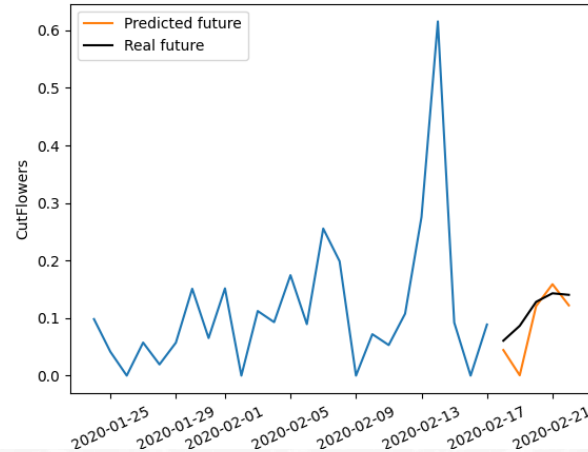
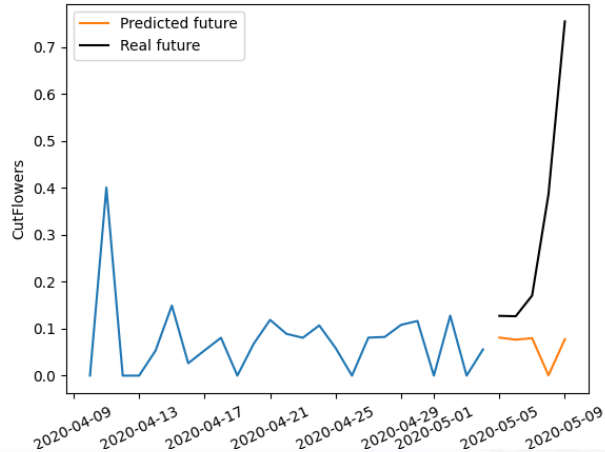
```
forecast = forecaster.forecast(  
    inputs,  
    prediction_length=5,  
)
```

*# Eine Prediction durchführen*  
*# Die Inputdaten übergeben*  
*# Die Prognoselänge definieren*

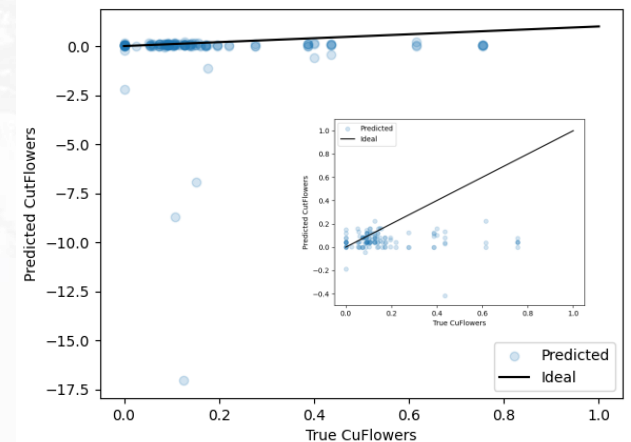
```
y_pred = forecast.mean
```

*# Mittelwerte der Vorhersage bilden*

# Toto Ergebnisse



Bestimmtheitsmaß: -108  
Mittlerer Absoluter Fehler: 0.416



# ML vs. LSTM vs. CNN vs. Transformer Modelle



## Machine Learning Modelle

Traditionelle Machine Learning Modelle wie lineare Regression, Entscheidungsbäume oder Random Forests sind einfach zu implementieren und benötigen oft weniger Rechenressourcen. Sie sind gut geeignet für kleinere Datensätze und bieten oft eine schnelle Lösung für einfache Probleme. Allerdings können sie Schwierigkeiten haben, komplexe Muster in Zeitreihendaten zu erfassen, insbesondere wenn es um langfristige Abhängigkeiten geht. Ihre Leistung kann stark von der Merkmalsauswahl abhängen, was zusätzliche Vorverarbeitung erfordert.

## LSTM-Modelle

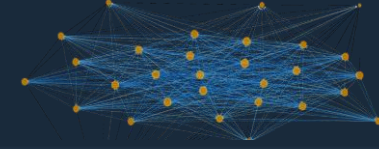
Long Short-Term Memory (LSTM) Modelle sind eine spezielle Art von rekurrenten neuronalen Netzen, die entwickelt wurden, um langfristige Abhängigkeiten in Sequenzdaten zu erfassen. Sie sind besonders effektiv für Zeitreihenprognosen, da sie in der Lage sind, Informationen über längere Zeiträume zu speichern. Der Nachteil ist, dass LSTMs rechenintensiv sind und eine größere Menge an Daten benötigen, um gut zu generalisieren. Zudem kann das Training lange dauern und sie sind anfällig für Überanpassung, wenn die Daten nicht ausreichend diversifiziert sind.

## 1D-Convolution-Modelle

1D-Convolutional Neural Networks (CNNs) sind darauf ausgelegt, lokale Muster in sequenziellen Daten zu erkennen. Sie sind effizient in der Verarbeitung von Zeitreihendaten, da sie durch die Verwendung von Faltungsschichten die Anzahl der zu lernenden Parameter reduzieren. Dies führt zu schnelleren Trainingszeiten und einer besseren Generalisierung. Ein Nachteil ist, dass sie möglicherweise nicht so gut wie LSTMs in der Lage sind, langfristige Abhängigkeiten zu erfassen, da sie sich auf lokale Informationen konzentrieren.

## Transformer-Modelle

Transformer-Modelle haben sich als äußerst leistungsfähig für verschiedene Aufgaben der natürlichen Sprachverarbeitung erwiesen und finden zunehmend Anwendung in der Zeitreihenprognose. Sie nutzen Selfattention, um Beziehungen zwischen allen Zeitpunkten in der Eingabesequenz zu erfassen, was sie besonders gut für komplexe Zeitreihen macht. Der Nachteil ist, dass sie sehr ressourcenintensiv sind und große Mengen an Daten benötigen, um effektiv zu arbeiten. Zudem kann die Implementierung komplex sein, was eine steilere Lernkurve für Entwickler bedeutet.



```
import tf_keras as keras
import tensorflow_probability as tfp

model = keras.models.Sequential()
model.add(keras.layers.Input(x_train.shape[1:]))
model.add(keras.layers.LSTM(100,return_sequences=False))
model.add(keras.layers.Dense(100,activation='tanh'))
model.add(keras.layers.Dense(forecast_length*2,activation='linear'))
model.add(tfp.layers.DistributionLambda(lambda t:
    tfp.distributions.Normal(loc=t[:, :forecast_length],
    scale=1e-3 + tf.nn.softplus(t[:, forecast_length:]))))

def negative_logarithmic_likelihood(y, y_pred):
    return -y_pred.log_prob(y)

model.compile(loss= negative_logarithmic_likelihood, optimizer=optimizer,metrics=[ 'r2_score'])
model.fit(x_train, y_train, batch_size=20, epochs=3000,validation_data=(x_val, y_val))

y_pred = model(x_test)
```

*# Ein neues Modell mit Keras erzeugen*  
*# Eine Input-Schicht hinzufügen*  
*# Die Zeitpunkte mit LSTM sequentiell lesen*  
*# Die 100 neuen Merkmale verarbeiten*  
*# Die Vorhersage produzieren*  
*# Die Prediction als Gauß-Kurve modellieren*  
*# Die erste Hälfte für Erwartungswerte*  
*# Die zweite Hälfte für Standardabweichungen*

*# Den negativen logarithmischen Likelihoodwert minimieren*

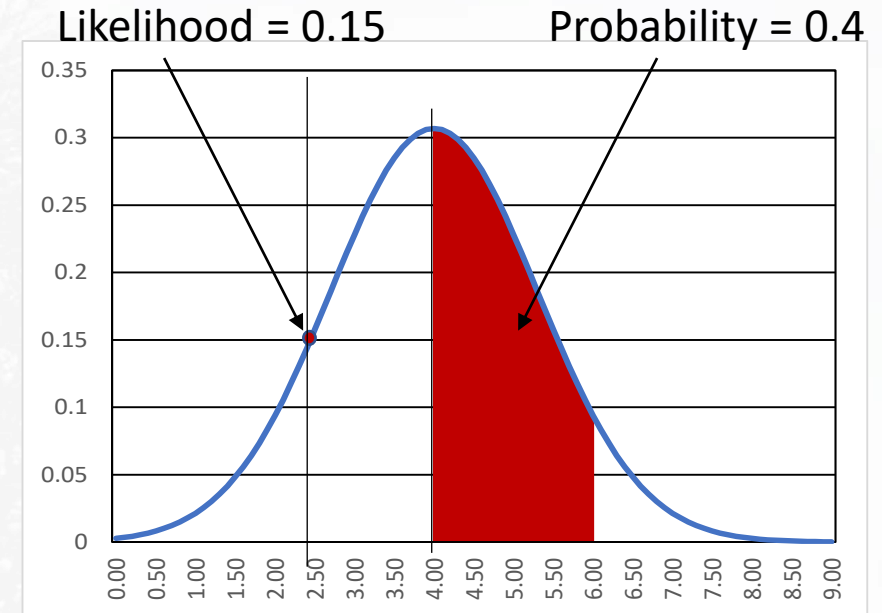
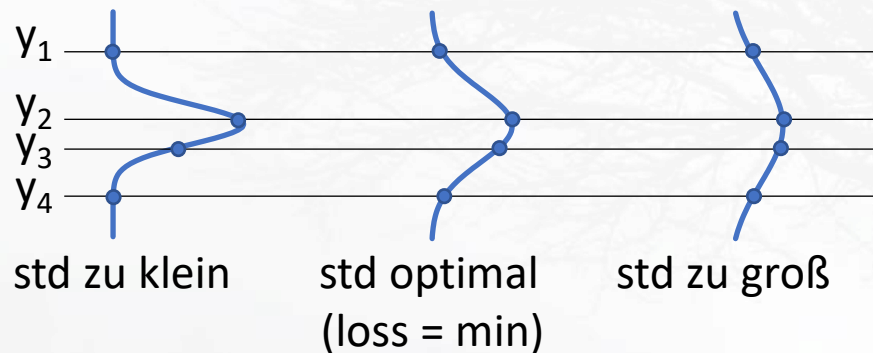
*# Eine Prediction der Erwartungswerte und Standardabweichungen*

# Bayes'sche neuronale Netze



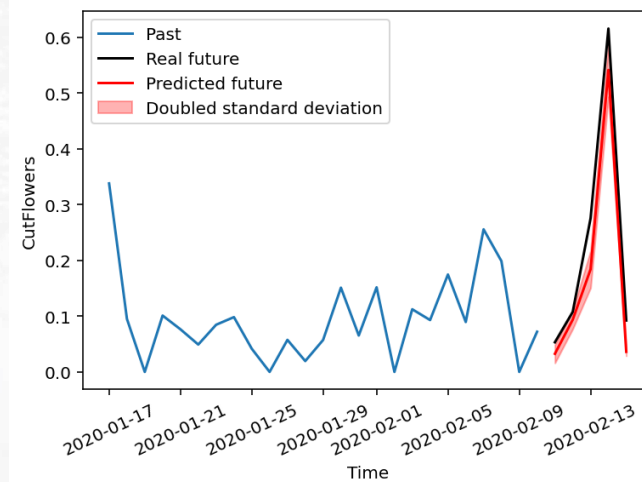
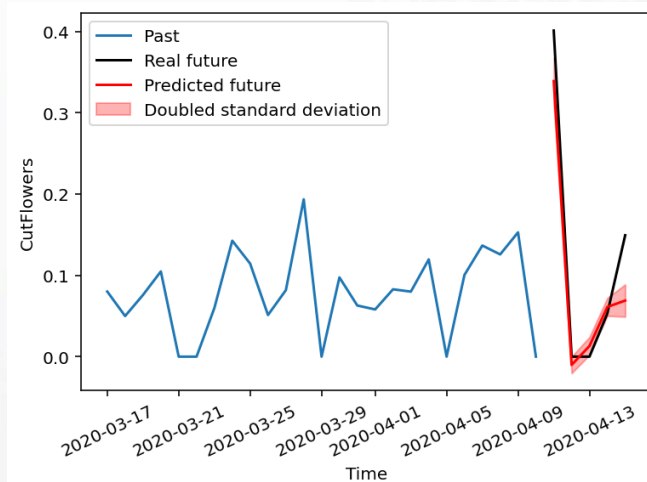
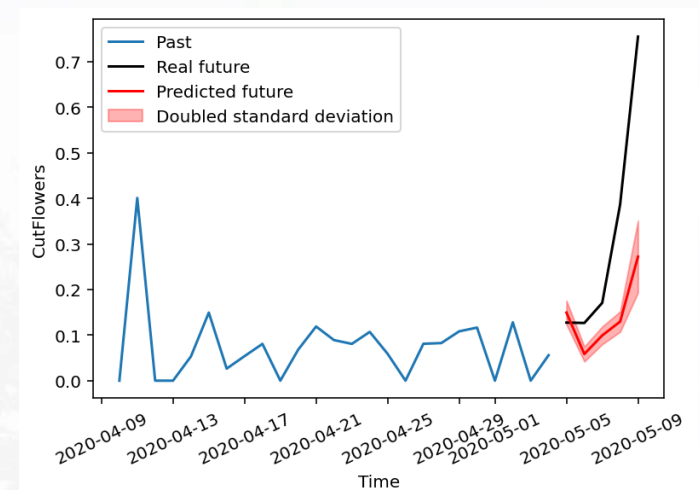
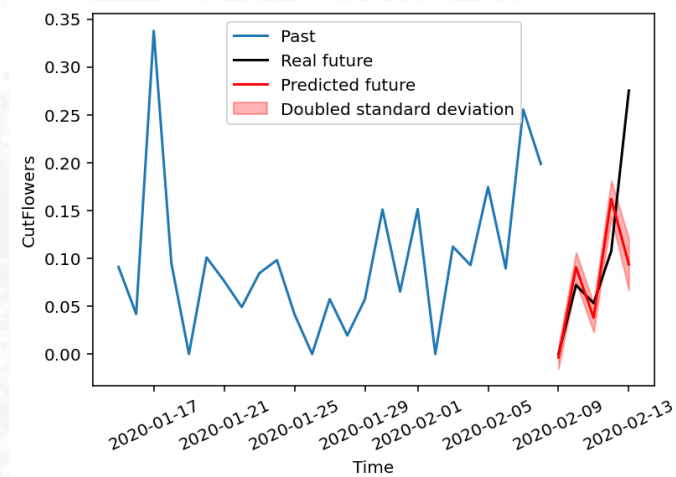
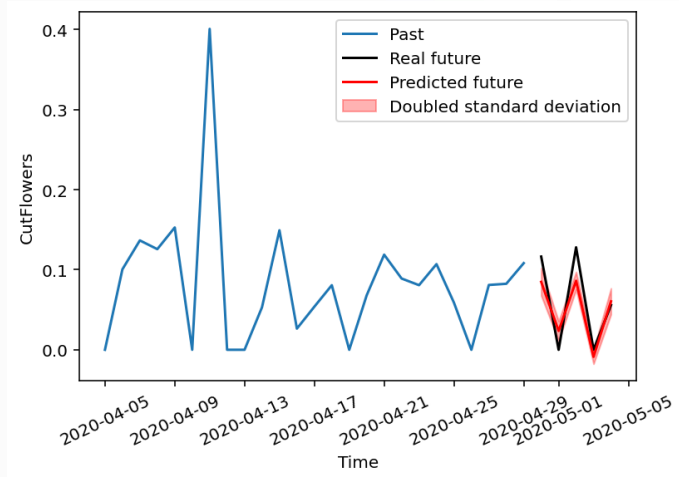
Anstelle von genau definierten Parametern in neuronalen Netzen werden Verteilungsfunktionen erlernt

Wir verwenden z.B. Normalverteilungen am Output der Modelle für die Mittelwerte und Standardabweichungen erlernt werden

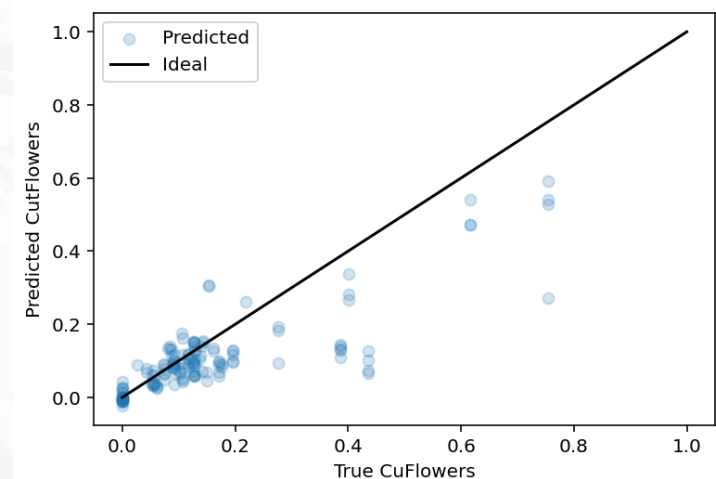


Die Minimierung des negativen logarithmischen Likelihood-Losses bewirkt, dass sich die Verteilungsfunktion bestmöglich an die gegebenen Labels anpasst

# LSTM Ergebnisse



Bestimmtheitsmaß: 0.582  
Mittlerer Absoluter Fehler: 0.062



# Feature Importance



Wir vermuten eine große Abhängigkeit der verkauften Blumen von den Merkmalen, insbesondere Feiertagen

Wir können diese Abhängigkeit betrachten

Künstliche neuronale Netze sind symbolisch differenzierbar, das brauchen wir für die Optimierung

$$loss = \frac{1}{n} \sum (y_{train} - y_{pred}(p))^2 \quad p^* = p - lr \frac{dloss}{dp}$$

Beim Training neuronaler Netze werden Modellparameter  $p$  mithilfe einer Lernrate  $lr$  und Gradienten  $dloss/dp$  kontinuierlich verbessert

Für die Feature Importance bilden wir stattdessen den Gradienten von  $y_{pred}$  als Funktion von  $x_{train}$

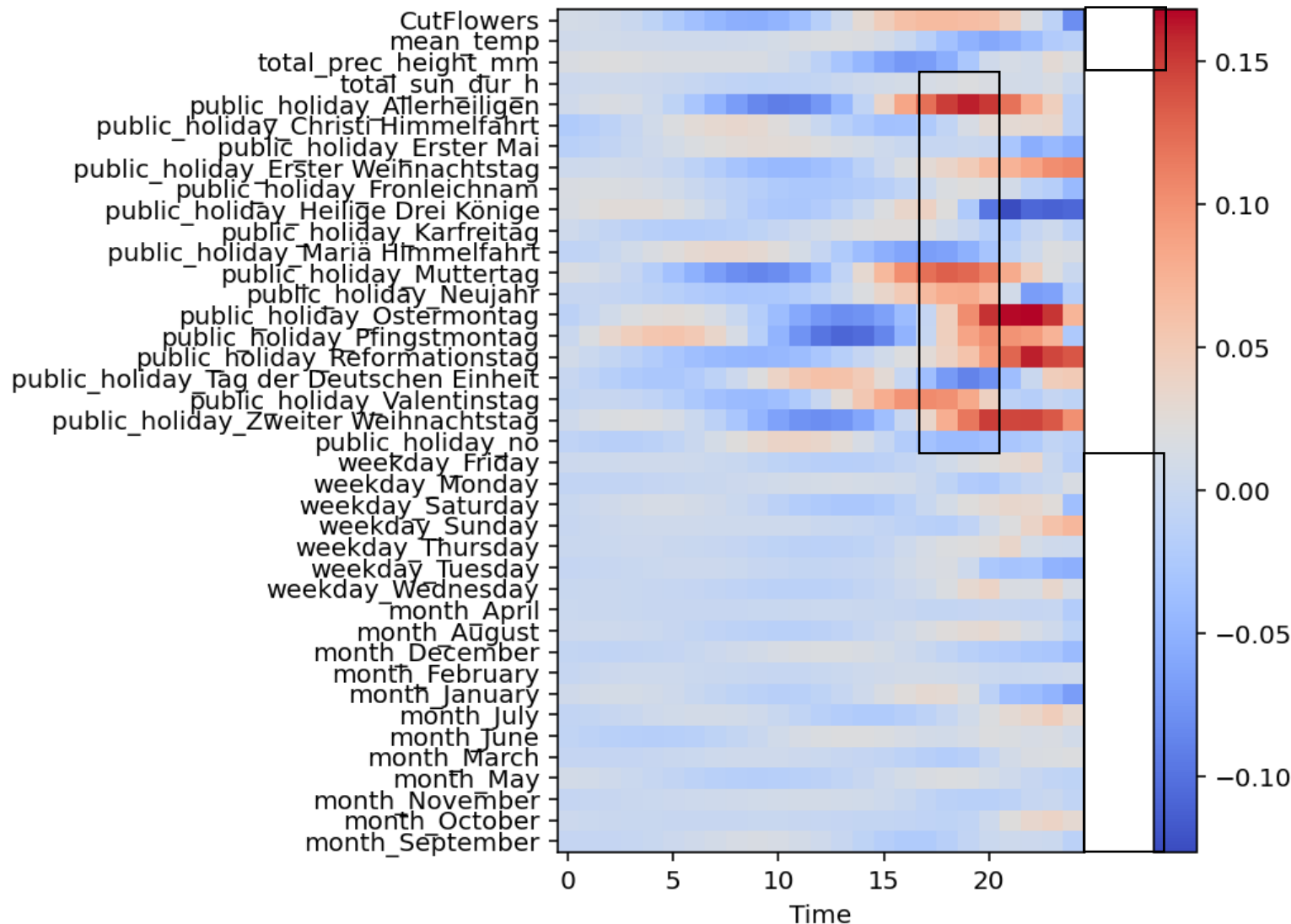
$$y_{pred} = model(x_{train}) \quad \text{Feature Importance} = \frac{dy_{pred}}{dx_{train}}$$

`x_train = tf.Variable(x_train)` *# Wir müssen x\_train in eine TensorFlow-Variable konvertieren*

`with tf.GradientTape() as tape:` *# Wir nutzen in TensorFlow die GradientTape-Funktion um automatisch*  
`y_pred = model(x_train).mean()` *# Ableitungen der gegebenen Gleichung zu berechnen*

`grads = tape.gradient(y_pred, x_train)` *# Die Ableitung von y\_pred nach x\_train*

# Gradient $dy_{\text{pred}} / dx_{\text{train}}$



- Merkmal vergrößern  
→ mehr Blumen verkaufen
- Merkmal vergrößern  
→ weniger Blumen verkaufen
- Prognosebereich



- Foundation Models werden zunehmend verwendet um ohne weiteres Training komplexe Daten auszuwerten
- Für die Zeitreihenprognose ist die Auswahl an Foundation Models noch sehr überschaubar
- Wir haben TimeGPT, Chronos und Toto mit individuellen LSTM-Modellen verglichen
- Die individuellen LSTM-Modelle konnten insbesondere Feiertage deutlich besser berücksichtigen und erzielten bessere Prognosen
- Generative KI kann (noch) nicht alle Aufgaben im Data Science lösen

Feedback:



<https://www.umfrageonline.com/s/xcrsxzi>

[www.linkedin.com/in/bernd-ebenhoch](https://www.linkedin.com/in/bernd-ebenhoch)