



FROSCON 2023

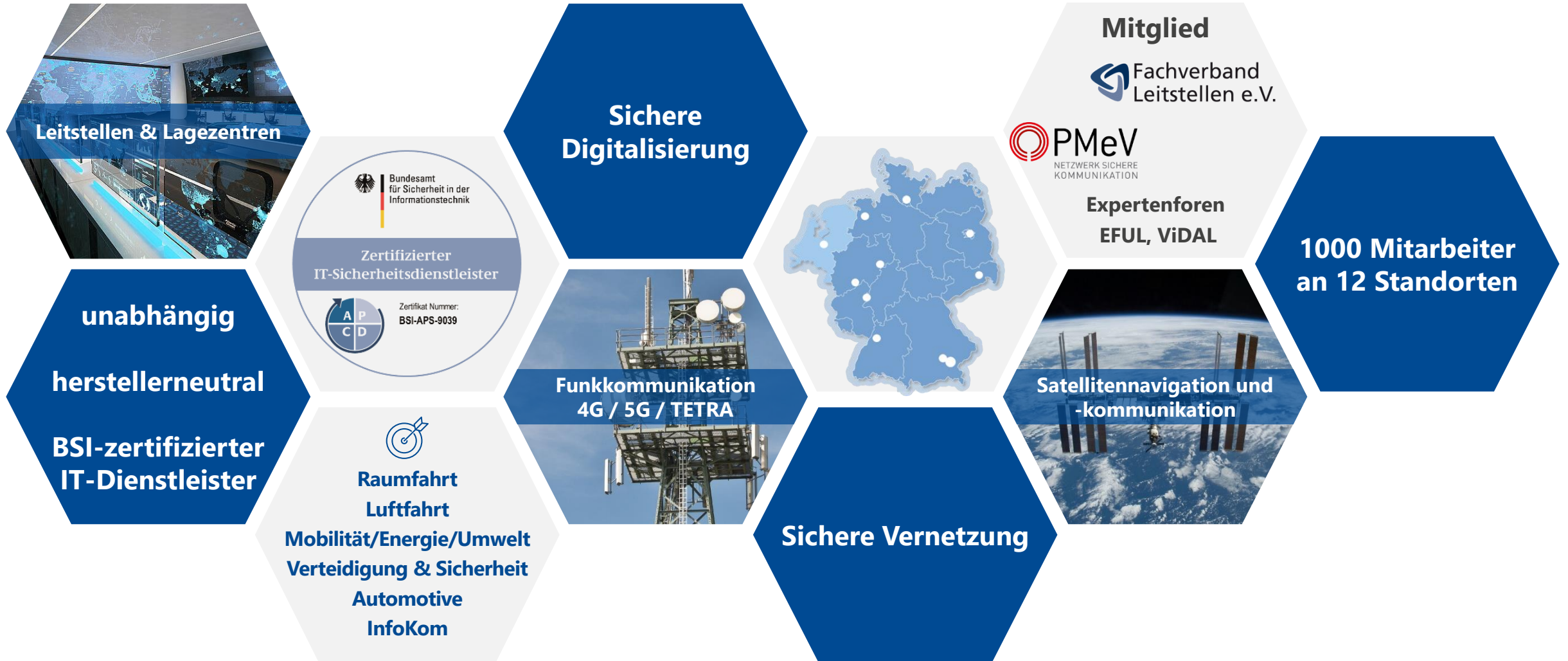
Vorhersage des Einsatzaufkommens in Rettungsleitstellen

Oliver Heinzl M.Sc., Achim Marikar M.Sc. Dipl.-Ing.(FH)

Agenda

- IABG
- Ausgangssituation & Motivation
- Verwendete Daten
- Umsetzung
- Vertrauen schaffen
- Do it yourself!
- Ausblick







1

Ausgangssituation & Motivation

Zuständig für



250.000 Einwohner

Erfasst



90.000 Ereignisse pro Jahr

Zusammenarbeit mit



- **22** freiwilligen Feuerwehren
- **162** Ortsteilfeuerwehren
- **5** Werksfeuerwehren
- **1** Universitätsklinikum
- **2** Krankenhäuser der Grundversorgung

Veranlasst



13.700 Gebietsabsicherungen* im Jahr

* Verlegefahrten der Rettungswagen um lokale Engpässe zu vermeiden

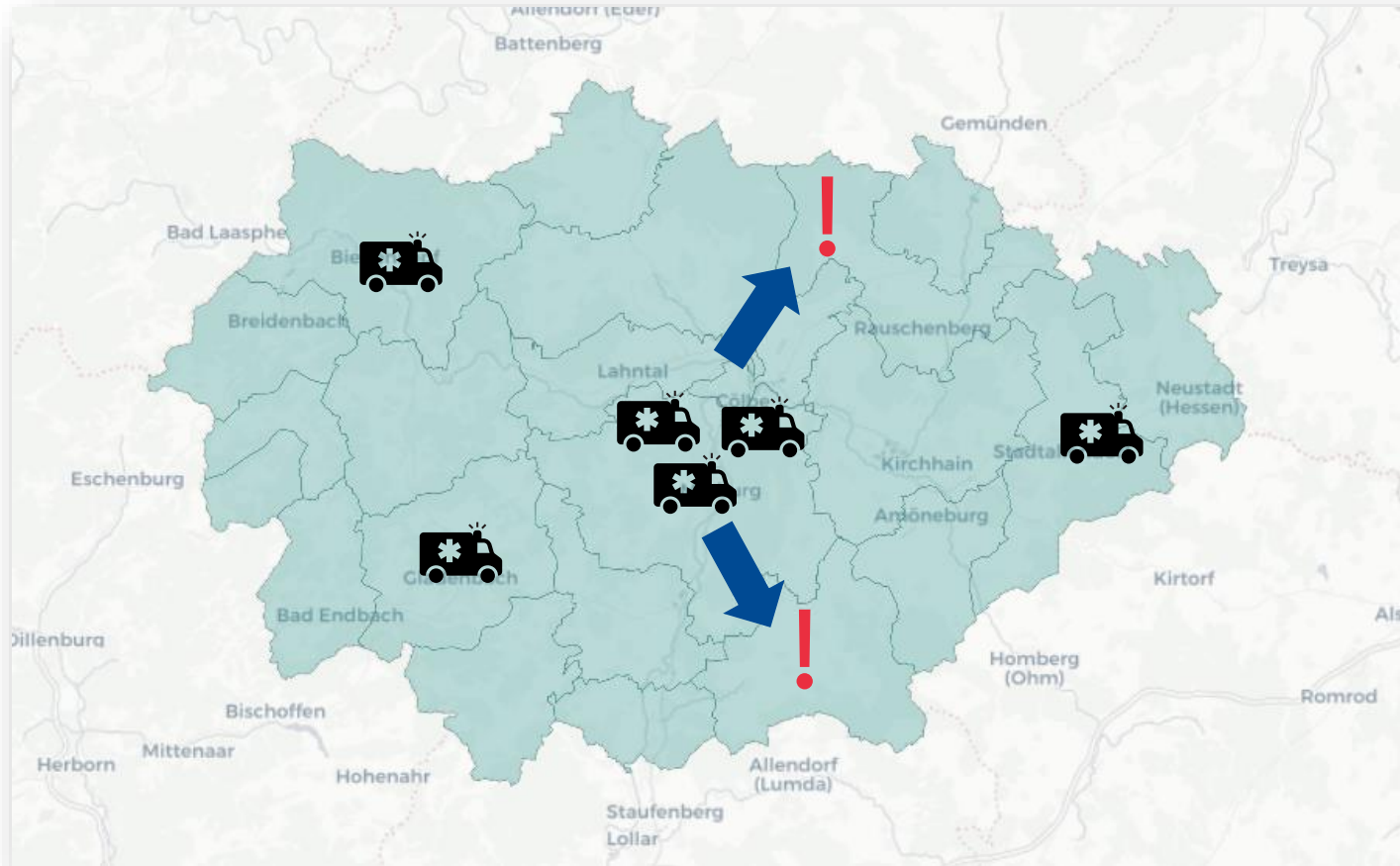
LANDKREIS



MARBURG
BIEDENKOPF

Besteht aus

22 Städten und Gemeinden



Möglichkeit 1 *Häufung der Ressourcen in Städtischen Gebieten*

Vorteil

- Schneller an den meisten Einsatzorten

Nachteil

- Ländliche Gebiete werden benachteiligt
- Hilfsfrist* wird evtl. nicht erfüllt
 - Fatale Folgen möglich

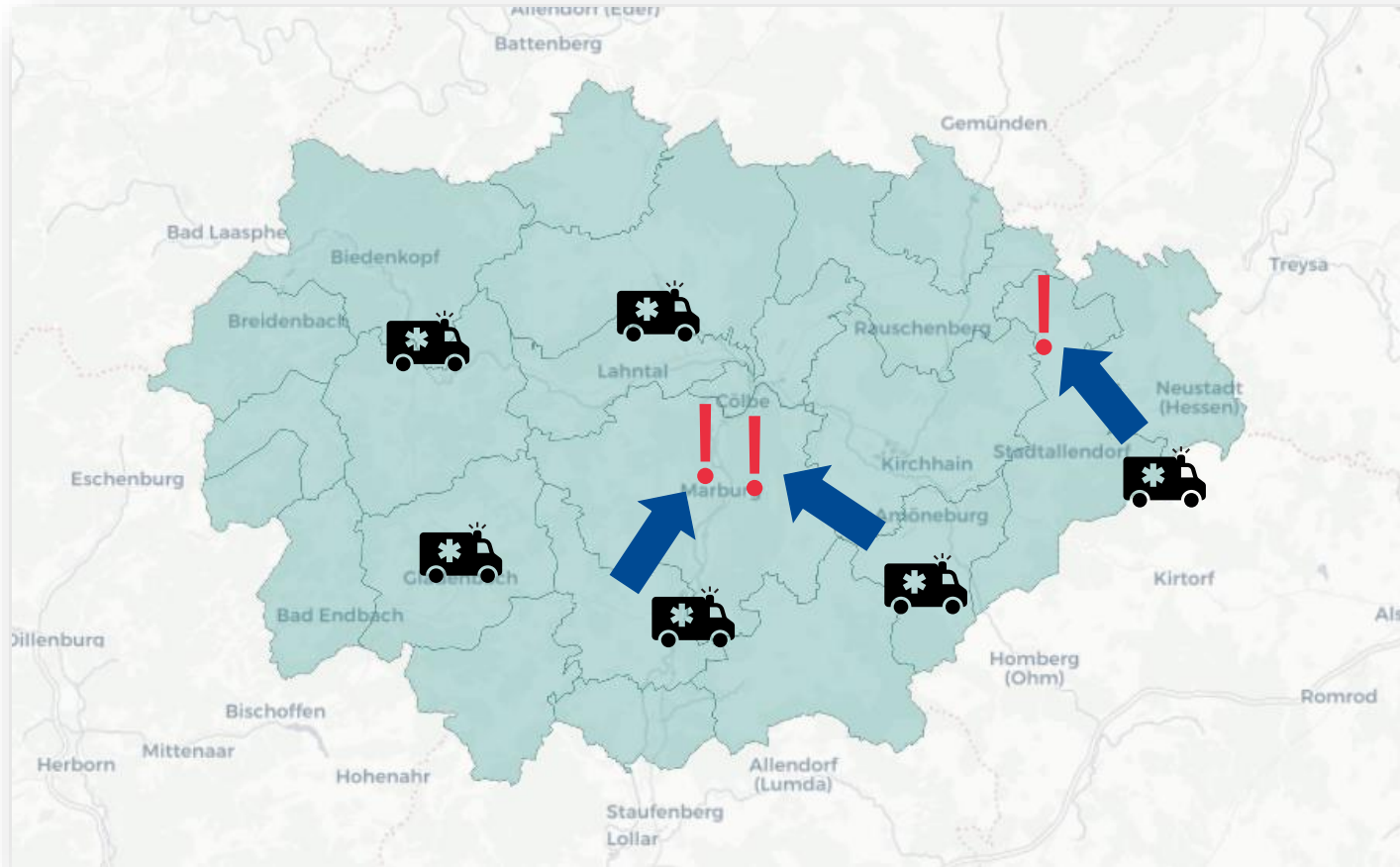
* Erreichen des Einsatzortes innerhalb von 10 Minuten in mindestens 90% der Fälle



Einsatzposition



Rettungswagen



Möglichkeit 2 *Verteilung der Ressourcen in der gesamten Region*

Vorteil

- Bessere Gebietsabsicherung

Nachteil

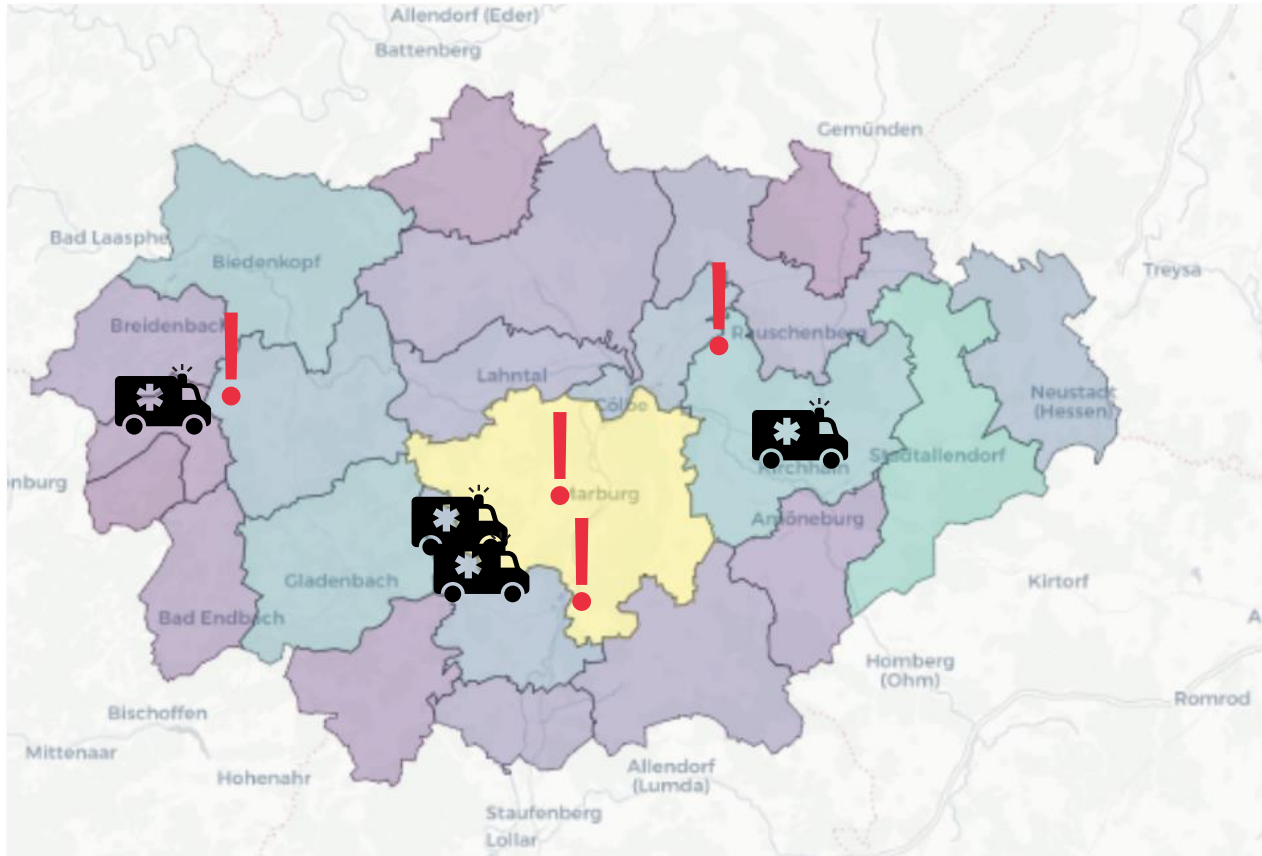
- Die mittlere Wartezeit erhöht sich (die meisten Einsätze erfolgen in den Städten)



Einsatzposition



Rettungswagen



Lösung mit Künstlicher Intelligenz
Verteilung der Ressourcen basierend auf einer berechneten Prognose

Prognose

- des Einsatzvolumens
- der örtlichen Verteilung

Ziele

- Einhaltung der Hilfsfrist
- Reduzierung der Verlegefahrten zur Gebietsabsicherung
- Einsparung von Ressourcen



Einsatzposition



Rettungswagen

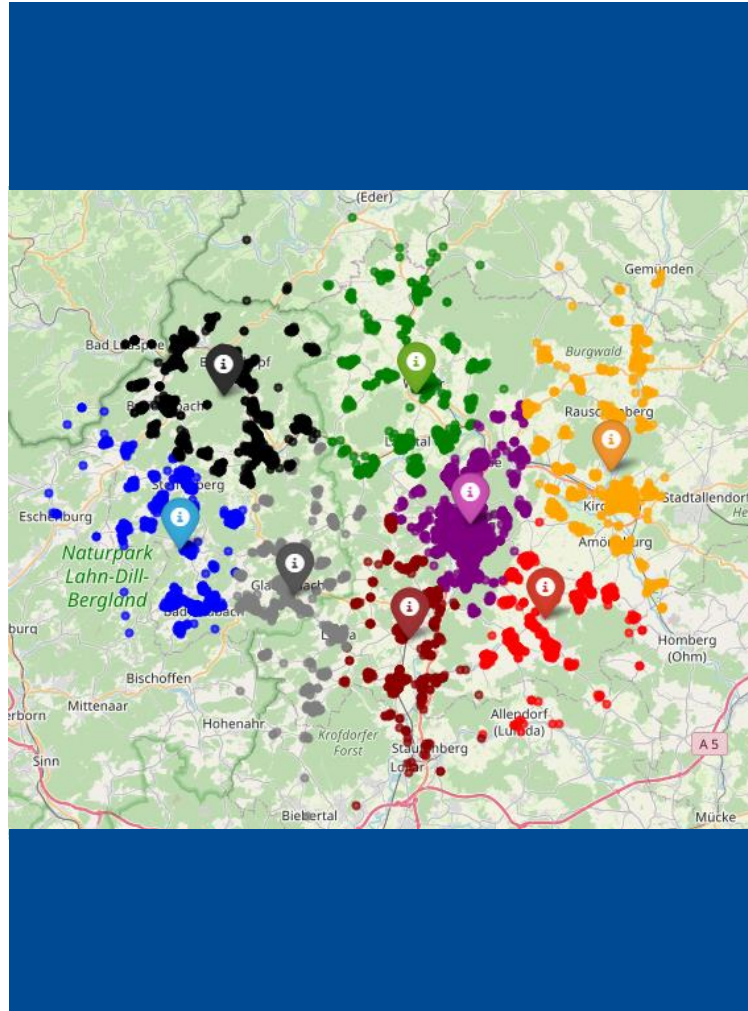
2

Verwendete Daten

KI-Modell verwendet einen umfangreichen Datensatz:

- **historische** Informationen über vergangene **Einsätze** aus dem gesamten Landkreis (Marburg-Biedenkopf)
- umfasst eine Vielzahl von Merkmalen:
 - Einsatzort
 - Zielort
 - Einsatzart
 - Meldebild
 - Einsatzmittel
 - Einsatzanfang
 - Einsatzende
 - Alter

1



Des Weiteren werden **Wetterdaten** aus dem Open-Data-Projekt des DWDs benutzt:

- Temperatur
- Druck
- Niederschlag
- Windstärke
- Warnungen

2





Pandas: ist eine leistungsstarke Bibliothek für Datenmanipulation. Es können Daten eingelesen, bereinigt, gefiltert und aggregiert werden.



NumPy: bietet leistungsstarke Array-Operationen und mathematische Funktionen, die für statistische Berechnungen notwendig sind.



Statsmodels: bietet eine Vielzahl von Funktionen für statistische Analysen, Hypothesentests, lineare Regression und Zeitreihenanalysen.



Matplotlib: ermöglicht die Visualisierung von Daten und statistischen Ergebnissen. Matplotlib ist vielseitig und kann nahezu alle Arten von Diagrammen erstellen.



GeoPandas: erweitert Pandas, um geografische Daten zu verarbeiten und bietet Funktionen für die räumliche Datenanalyse.



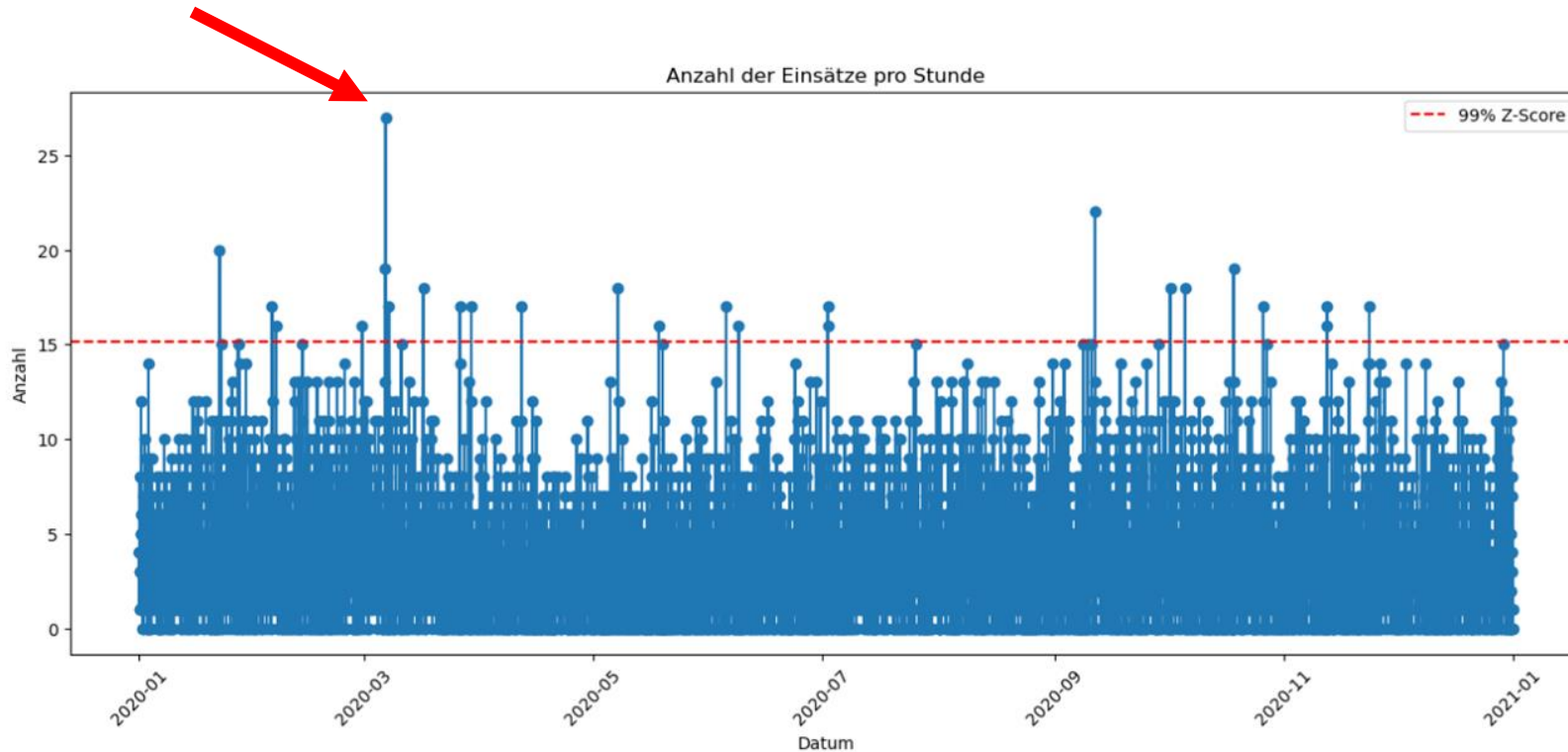
SciPy: bietet zusätzliche Funktionalitäten für Berechnungen. Es enthält statistische Funktionen, lineare Algebra, Interpolation und Optimierungsalgorithmen.



Scikit-learn: ist eine umfangreiche Bibliothek für Maschinelles Lernen. Sie bietet lineare Regression, Entscheidungsbäume, k-Nächste-Nachbarn und noch vieles mehr.



Seaborn: ähnlich wie Matplotlib ermöglicht diese, optisch ansprechende Visualisierungen zu erstellen, wurde aber auf die Erstellung statistischer Plots optimiert.



Vorgehensweise

1. Formatierung der Daten mit Pandas
2. Berechnung des Z-score mit Statsmodels oder Scipy.stats
Alternative: Verwendung des Clustering-Algorithmus aus dem Clustering-Modul von Scikit-learn
3. Visualisierung mit matplotlib

Bus kracht in Apotheke



Am Freitag kam es am Südbahnhof zu einem Großeinsatz von Polizei und Rettungskräften. Ein Stadtbus war in die Apotheke gefahren. 16 Menschen wurden verletzt. Lebensgefahr besteht laut Polizei aber nicht.



Nadine Weigel
06.03.2020, 21:05 Uhr



Oft ist eine Erklärung einer Anomalie über eine kurze Internetrecherche möglich.

3

Umsetzung

Erstellung des Prognose-Modells



Was ist Künstliche Intelligenz?



Künstliche Intelligenz ist die Eigenschaft eines IT-Systems, »menschenähnliche«, intelligente Verhaltensweisen zu zeigen.

Bitkom e.V. und Deutsches Forschungszentrum für künstliche Intelligenz

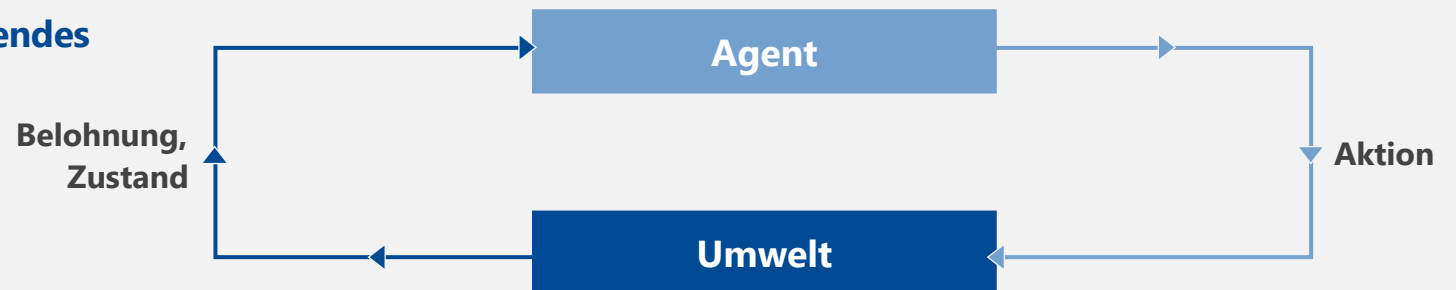
Überwachtes Lernen



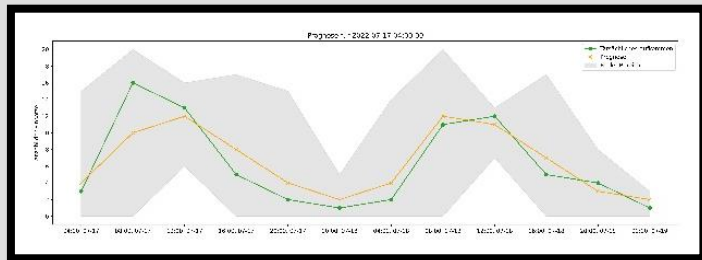
Unüberwachtes Lernen



Verstärkendes Lernen



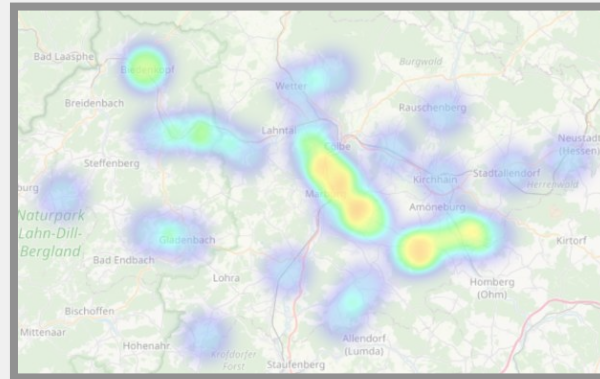
Volumenmodell



Vorhersage des Einsatzvolumens
(Überwachtes Lernen)

1

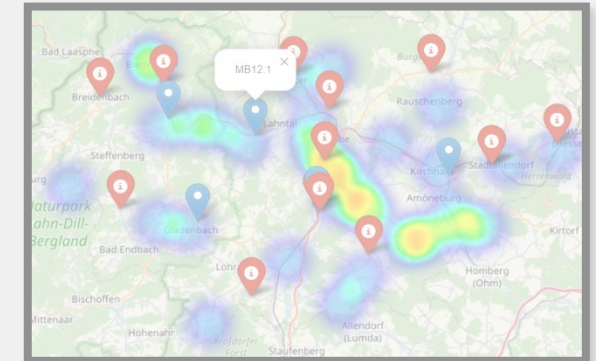
Verteilungsmodell



Vorhersage der örtlichen Verteilung der
Einsätze
(Unüberwachtes Lernen)

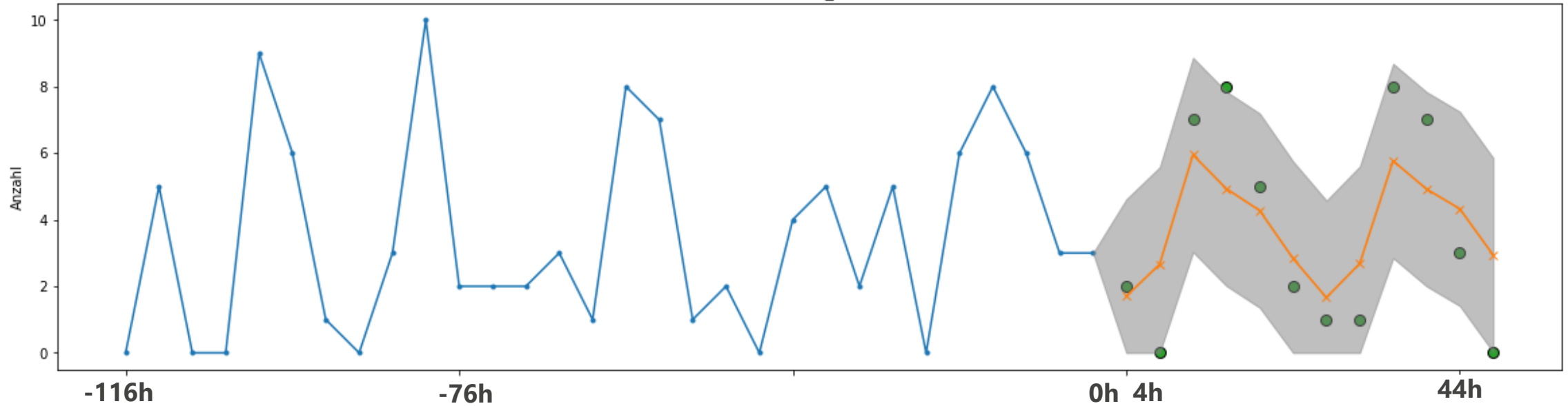
2

Allokationsmodell



Unterstützung der Allokation basierend
auf den Prognosen
(Verstärkendes Lernen/MIP)

3



Werte der letzten 7 Tage aus dem Einsatzleitsystem

Prognose für die nächsten 2 Tage

- Historie
- Tatsächliche Daten
- Prognose
- Konfidenzbereich



Wetterdaten & Warnmeldungen



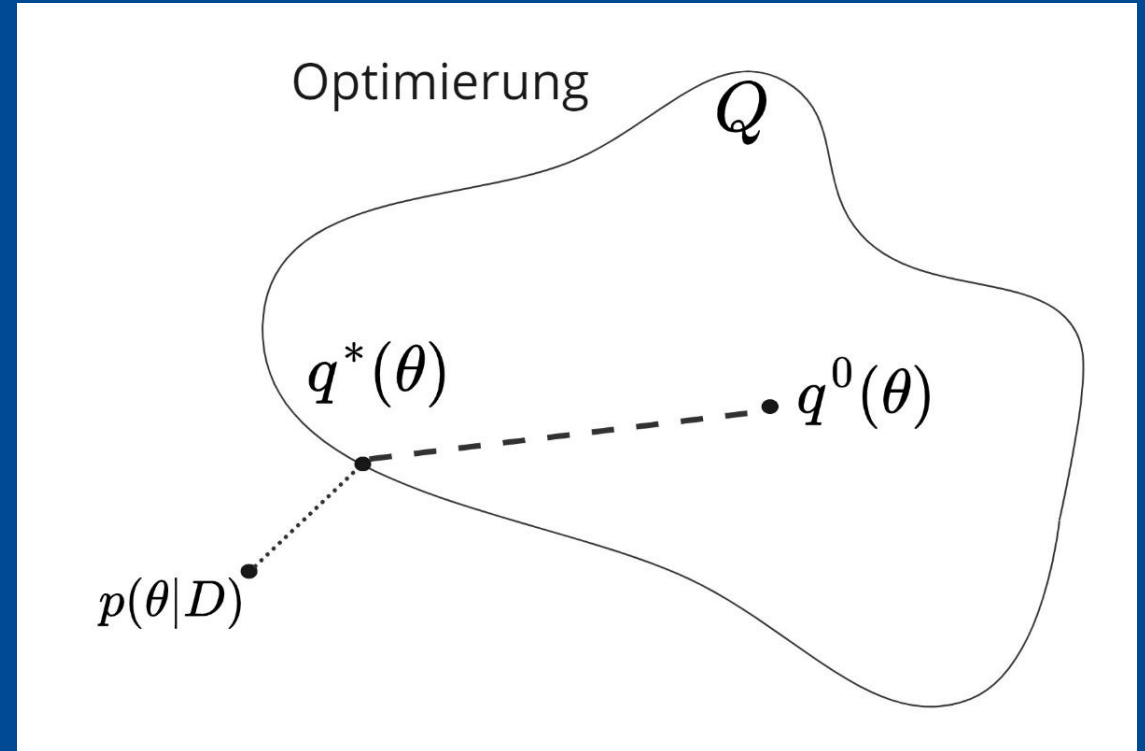
Prognose Zeitpunkt

Ziel des Volumenmodells ist es für ein neues gegebenes Input x^* die prädiktive Verteilung der Labels y^* zu erhalten:

$$p(y^*|x^*, D) = \int_{\Theta} p(y^*|x^*, \theta)p(\theta|D)d\theta$$

$p(\theta |D)$ ist nicht berechenbar, deshalb wird versucht eine einfachere approximierende Posterior-Verteilung $q(\theta)$ zu finden, in dem man die KL -Divergenz $KL(q(\theta)||p(\theta|D))$ minimiert:

$$\int q(\theta) \log(p(Y|X, \theta))d\theta - KL(q||p) \quad (1)$$



Es wird eine Familie von Verteilungen Q festgelegt. Das Ziel besteht darin, eine optimale Verteilung $q^*(\theta)$ zu finden, wobei die Optimalität als die kleinste mögliche KL -Divergenz zwischen $q^*(\theta)$ und $p(\theta)$ definiert ist.

Es ist davon auszugehen, dass die eigentlichen Werte durch eine bestimmte Funktion beschrieben werden können, diese aber dennoch durch Rauschen verfälscht werden. Deshalb wird angenommen:

$$p(y|x, \theta) \sim \mathcal{N}(y; \hat{y}(x, \theta), \frac{1}{\tau} I_F)$$

Hierbei steht nun $\hat{y} = \hat{y}(x, \theta)$ für die Funktion, welches von dem Modell beschrieben wird, mit Parametern θ .

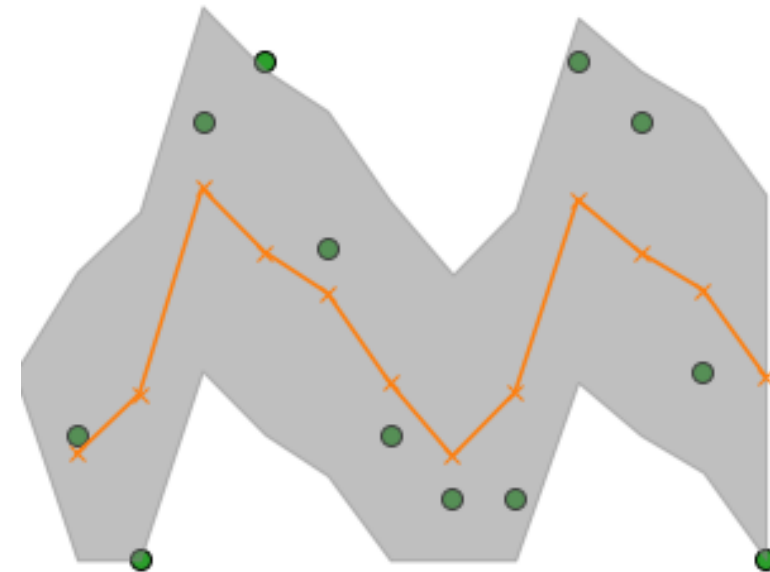
Wird beispielsweise nur ein einfaches Neuronales Netz mit nur einem Layer genommen, so hätte man:

$$\hat{y}(x, \theta) = \sigma(W \cdot x + b)$$

mit Aktivierungsfunktion σ , und mit den Parametern $\theta = (W, b)$.



```
tf.keras.layers.Dense(  
    units,  
    activation=None,  
    use_bias=True,  
    kernel_initializer='glorot_uniform',  
    bias_initializer='zeros',  
    kernel_regularizer=None,  
    bias_regularizer=None,  
    activity_regularizer=None,  
    kernel_constraint=None,  
    bias_constraint=None,  
    **kwargs  
)
```



Geht man davon aus, dass die Anzahl der Ereignisse unabhängig voneinander auftreten, so gilt:

$$\log p(Y|X, \theta) = \sum_i^N \log(p(y_i|x_i, \theta)) \propto -\frac{\tau}{2} \sum \|y_i - \hat{y}(x_i, \theta)\|_2^2$$

Basierend auf der Idee des Monte-Carlo-Dropouts werden die Parameter des Netzes θ durch

$$\theta = z \odot M$$

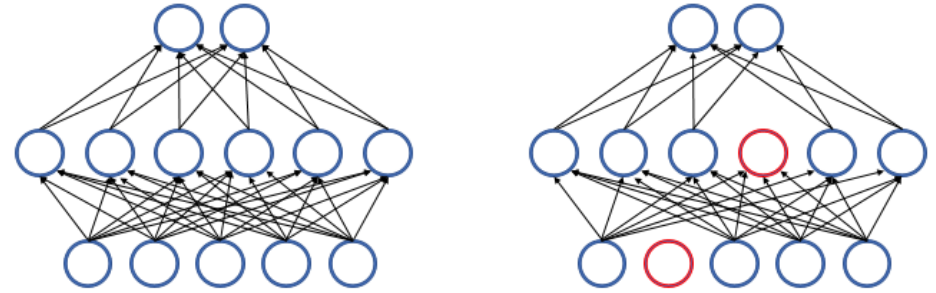
parametrisiert, wobei die Einträge $z \sim \text{Bernoulli}(p)$ sind.

Dadurch erhält man

$$\int q(\theta) \log(p(Y|X, \theta)) d\theta = \int q(\theta(z)) \log(p(Y|X, \theta(z))) dz \quad (2)$$

$$\approx \sum \log(p(y_i|x_i, \theta(\hat{z}_i))) \quad (3)$$

mit Realisationen \hat{z} .



Without Dropout

With Dropout



```
tf.keras.layers.Dropout(  
    rate, noise_shape=None, seed=None, **kwargs  
)
```

Optimierung III

Das Optimierungsproblem:

$$\min_M E_{z \sim \text{Bernoulli}(p)} \left[\frac{\tau}{2} \sum_{i=0}^N \|y_i - \hat{y}(x_i, z \odot M)\|_2^2 \right] + \frac{1}{2} \|M\|_2^2$$



```
class MeanSquaredError(Loss):  
  
    def call(self, y_true, y_pred):  
        return tf.reduce_mean(tf.math.square(y_pred - y_true), axis=-1)  
  
tf.keras.losses.Loss(  
    reduction=losses_utils.ReductionV2.AUTO, name=None  
)
```

```
 $m_0 \leftarrow 0$  (Initialize 1st moment vector)  
 $v_0 \leftarrow 0$  (Initialize 2nd moment vector)  
 $t \leftarrow 0$  (Initialize timestep)  
while  $\theta_t$  not converged do  
     $t \leftarrow t + 1$   
     $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$  (Get gradients w.r.t. stochastic objective at timestep  $t$ )  
     $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$  (Update biased first moment estimate)  
     $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$  (Update biased second raw moment estimate)  
     $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$  (Compute bias-corrected first moment estimate)  
     $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$  (Compute bias-corrected second raw moment estimate)  
     $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$  (Update parameters)
```



```
tf.keras.optimizers.Adam(  
    learning_rate=0.001,  
    beta_1=0.9,  
    beta_2=0.999,  
    epsilon=1e-07,  
    amsgrad=False,  
    weight_decay=None,  
    clipnorm=None,  
    clipvalue=None,  
    global_clipnorm=None,  
    use_ema=False,  
    ema_momentum=0.99,  
    ema_overwrite_frequency=None,  
    jit_compile=True,  
    name='Adam',  
    **kwargs  
)
```

Genauigkeit der Prognose

Basierend auf dem optimalen Gewicht M^* , erhält man die relevanten Daten.

Erwartungswert

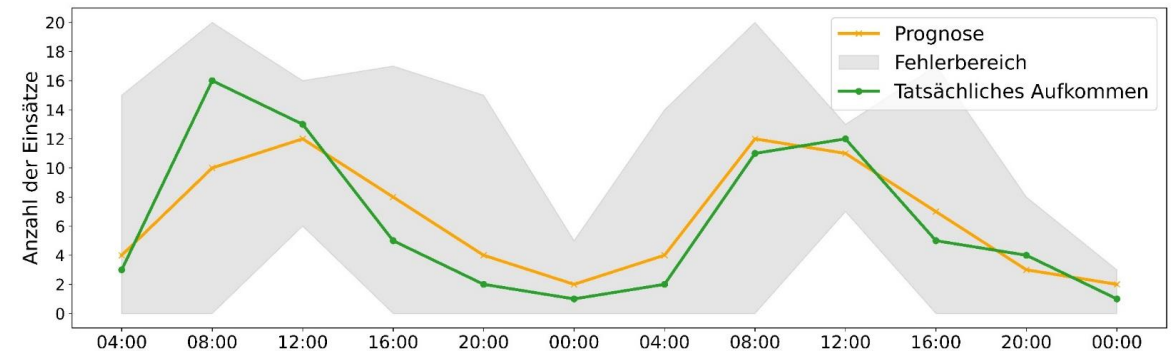
$$E_q(y^*) \approx \frac{1}{R} \sum_{r=0}^R \hat{y}(x^*, \hat{z}_r \odot M^*)$$

Varianz

$$E_q((y^*)^t \cdot y^*) \approx \frac{1}{\tau} Id + \frac{1}{R} \sum_{r=0}^R \hat{y}(x^*, \hat{z}_r \odot M^*)^T \hat{y}(x^*, \hat{z}_r \odot M^*)$$

Wobei $R \in \mathbb{N}$ eine feste Größe ist.

Das Model kann so angepasst werden, dass die Varianz auch optimiert wird und von dem Eingabewert abhängt. Dadurch gibt das Model zu jeder Prognose einen Konfidenzbereich an. Der Bereich schwankt abhängig davon, wie genau die Prognose berechnet werden konnte.





4

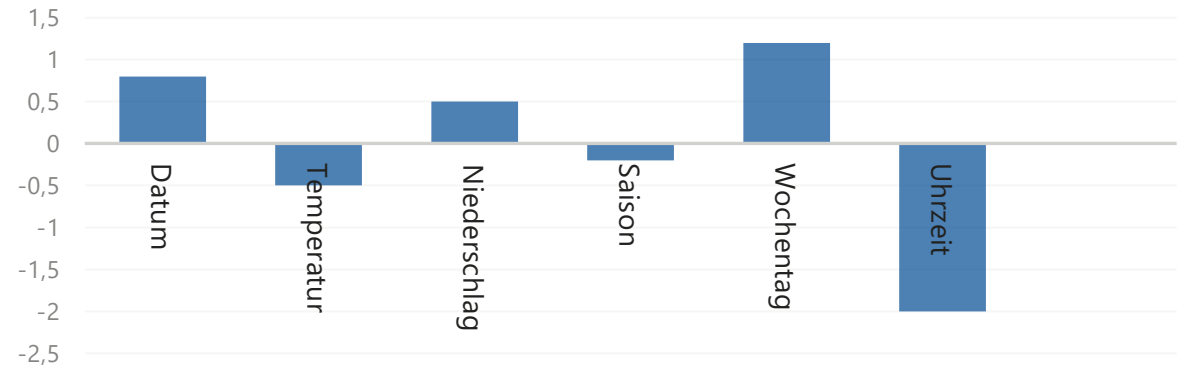
Vertrauen schaffen

Erklärbarkeit der Vorhersage gewährleisten

SHAP (SHapley Additive exPlanations):

- **Ziel:** Beiträge zur Vorhersage einer Zielgröße einzelner Features zu quantifizieren.
- **Methodik:** Aufteilung der Vorhersage des Modells in additive Beiträge, um den Einfluss jedes Features zu verstehen.
- Ermöglicht die Beantwortung wichtiger Fragen:
 - Wie trägt jedes Feature zur Vorhersage bei?
 - Welche Features haben den größten Einfluss auf eine Vorhersage?
 - Wie kann die Entscheidung des Modells für eine bestimmte Instanz erklärt werden?

Einfluss auf die Prognose zum 1. Juni 0:00 - 4:00





5

Do it yourself!

Seattle Open Data

Einsatzdaten frei verfügbar unter
<https://data.seattle.gov/Public-Safety/Seattle-Real-Time-Fire-911-Calls/kzjm-xkqj>



Address	Type	Datetime	Latitude	Longitude	Report Location
2208 15th Ave W	Auto Fire Alarm	2023 Aug 02 01:37:...	47.638695	-122.376149	POINT (-122.376149...
10715 8th Ave Ne	Aid Response	2023 Aug 02 01:35:...	47.706633	-122.32052	POINT (-122.32052 ...
2524 S Jackson St	Nurseline/AMR	2023 Aug 02 01:28:...	47.599265	-122.299586	POINT (-122.299586...
1607 42nd Ave E	Automatic Fire Alar...	2023 Aug 02 01:28:...	47.633391	-122.278538	POINT (-122.278538...
501 Rainier Ave S	Auto Fire Alarm	2023 Aug 02 01:26:...	47.598322	-122.3134	POINT (-122.3134 4...
Prefontaine Pl S / Y...	Aid Response	2023 Aug 02 01:17:...	47.601716	-122.330265	POINT (-122.330265...

Deutscher Wetterdienst

Wetterdaten frei verfügbar unter
<https://opendata.dwd.de>





6

Ausblick

Zukünftige Entwicklungen

Mitarbeit im Forschungsprojekt Aircis

- „Stärkung der Rettungskette unter Extremwettereinflüssen durch den Einsatz von Künstlicher Intelligenz am Beispiel Lausitz“
- Bisherige Erkenntnisse dienen als Basis für die Arbeit im Forschungsprojekt
- Weiterführen der Forschung mit vielen Partnern



Gefördert durch:



IABG mbH

Einsteinstraße 20
85521 Ottobrunn

Tel. +49 89 6088-0
Fax +49 89 6088-2220

info@iabg.de
www.iabg.de

IABG mbH, NL Bonn

Königswinterer Str. 552b
53227 Bonn



Oliver Heinzl

Data Analyst

heinzl@iabg.de
+49 89 6088 3365



Achim Marikar

Senior Consultant Critical
Communications

marikar@iabg.de
+49 228 91767 60



- [1] João Bento et al. "Timeshap: Explaining recurrent models through sequence perturbations". In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021, pp. 2565–2573.
- [2] Yarín Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Appendix. 2016. arXiv: 1506.02157 [stat.ML].
- [3] Yarín Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. 2016. arXiv: 1506. 02142 [stat.ML].
- [4] Christian Robert. "Machine Learning, a Probabilistic Perspective". In: CHANCE 27.2 (2014), pp. 62–63. doi: 10 . 1080 / 09332480 . 2014 . 914768. eprint: [https : / / doi . org / 10 . 1080 / 09332480 . 2014 . 914768](https://doi.org/10.1080/09332480.2014.914768). url: [https : //doi.org/10.1080/09332480.2014.914768](https://doi.org/10.1080/09332480.2014.914768).