Evaluation von Ausreißer-Behandlung und Auswirkungen von Umwelteinflüssen auf Feinstaub-Analysen

Aaron Braatz

Hamburg University of Applied Science
Faculty of Computer Science and Engineering
Department of Computer Science
Berliner Tor 7, 20099 Hamburg
aaron.braatz@haw-hamburg.de

Zusammenfassung. In dieser Arbeit wird der Einfluss von räumlichen und zeitlichen Ausreißern auf Kriging betrachtet und wie sich das Entfernen dieser Ausreißer positiv auf räumliche Interpolation von Feinstaubdaten auswirkt. Außerdem wird der Zusammenhang von Feinstaubmessungen zu Wetter, Verkehr und verschieden Tages- und Jahreszeiten evaluiert.

Schlüsselwörter: raumzeitliche Datenanalyse · Feinstaubanalyse · Kriging · Ausreißer Erkennung · Korrelation · Clustering

1 Einleitung

Die Analyse von Feinstaubdaten ist eine Teildisziplin der raumzeitlichen Datenanalyse (*engl. Spatio-temporal Data Analysis*; STDA). Feinstaubdaten werden mittels verschiedener Sensoren gemessen. Die Sensoren liefern sogenannte Rohdaten, die für die weiteren Analysen aufgearbeitet werden müssen. Zudem ist das Zusammenspiel mit anderen raumzeitlichen Daten relevant, sowie deren gegenseitiger Einfluss aufeinander.
Diese Arbeit ist Teil eines Projekts für STDA von Feinstaubdaten und baut auf [Braatz, 2021] auf. Darin wurde eine Pipeline für die Analyse von Feinstaubdaten vorgestellt wurde. In diesem Teil des Projekts wird die Toolchain praktisch getestet. Dabei wird in der Vorverarbeitung ein besonderer Fokus auf die Ausreißererkennung und -entfernung gelegt. Daraufhin wird die Auswirkung des Entfernens von Ausreißern auf die räumliche Interpolation von Feinstaubdaten evaluiert. Außerdem werden die vorverarbeiteten Daten genutzt, um Zusammenhänge zu anderen Umwelteinflüssen zu untersuchen (siehe Abbildung 1.0.1).
In Abschnitt 2 werden die Besonderheiten der Feinstaubdaten kurz zusammengefasst und die praktische Vorverarbeitung wird beschrieben. Abschnitt 3 beschreibt die Ausreißererkennung und evaluiert die Verbesserungen für das Kriging. Die Auswirkungen von Umwelteinflüssen werden in Abschnitt 4 behandelt. Zum Abschluss wird in Abschnitt 5 ein Fazit gezogen und ein Ausblick auf

zukünftige Arbeiten gegeben.
Im Anhang in Unterabschnitt 7.1 werden ferner Ergänzungen und Änderungen der Toolchain beschrieben, die sich in der praktischen Nutzung ergeben haben.

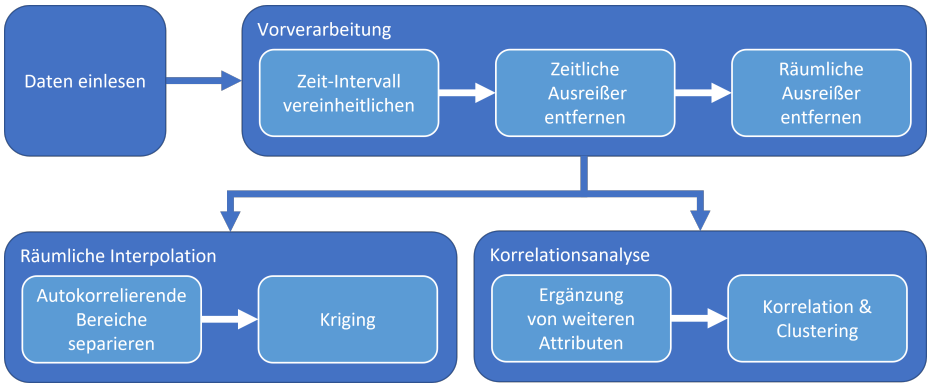

Abb. 1.0.1. Bestandteile dieser Arbeit (eigene Darstellung)

2 Feinstaub-Datensatz

Der untersuchte Feinstaub-Datensatz wird bereitgestellt von dem Projekt Sensor.Community[1] (ehem. Luftdaten.info). Dieser Datensatz wurde bereits in einer vorherigen Arbeit genauer untersucht [Braatz, 2021, S. 9-11]. Der genaue Aufbau des Datensatzes wird noch einmal im Anhang in Unterabschnitt 7.2 beschrieben. Die Erkenntnisse zu dem Datensatz sind:

- Über die Zeit kommen neue Messstationen dazu oder werden wieder entfernt (variable Anzahl an Messpunkten)
- Die Messungen liegen nicht in einer fest definierten Frequenz vor
- Teilweise werden unterschiedliche Datumsformate genutzt
- Einige Stationen haben Texte anstelle der Messwerte
- Es gibt extreme Messwerte, die auf Ausreißer schließen lassen
- Der Großteil (ca. 90 %) der Messwerte liegt unter 100 ppm
- Es gibt keine Informationen über die Umgebung der Sensoren (z.B. Indoor oder Outdoor)

2.1 Vorverarbeitung

Als erste Schritt in der Daten-Vorverarbeitung werden alle nicht benötigten Spalten entfernt (vergleiche Unterabschnitt 7.2). Damit bleiben folgende Spalten:

[1] `https://sensor.community/de/` (04.10.2021)

1. location - numerische Identifikation des Sensorknotens
2. lat - Breitengrad des Sensorknotens
3. lon - Längengrad des Sensorknotens
4. timestamp - Zeitpunkt der Messung
5. P1 - PM10 in ppm
6. P2 - PM2,5 in ppm

Die Spalte ,,timestamp" wird auf ein einheitliches Datumsformat in UTC Zeit überführt und alle nicht-numerischen Werte werden aus den Spalten ,,P1" und ,,P2" entfernt.
Messwerte in den Spalten ,,P1" und ,,P2" $>$ 500 ppm und $<$ 0 ppm werden ebenfalls entfernt. Da bereits ca. 90 % der Feinsatubdaten unter 100 ppm wird entschieden eine Obergrenze zu definieren. 500 ppm sind höher angesetzt, als Messungen an feinstaubreichen Tagen, wie an Silvester und werden daher entfernt.
Daraufhin werden die Messungen auf eine einheitliche Datenfrequenz von 10 Minuten gebracht (Eine Messung alle 10 Minuten). 10 Minuten werden gewählt, in Hinblick auf den Vergleich mit Wetterdaten, die in einem 10 Minuten-Takt abrufbar sind. Hierfür wird ein *zeitlich gewichteter Mittelwert* genutzt:

$$x_{10min} = \frac{1}{\sum t} \cdot \sum_{i=1}^{m} x_i \cdot t_i \tag{1}$$

dabei ist x_{10min} der zeitlich gewichtete Mittelwert für 10 Minuten, m die Anzahl der Messwerte innerhalb von 10 Minuten und t_i die Zeit für die der Messwerte x_i gültig ist (bis zum nächsten Messwert oder zur nächsten 10 Minuten-Grenze).
Sofern Daten in den Spalten ,,timestamp", ,,lat" oder ,,lon" fehlen, wird der dazugehörigen Datenpunkte entfernt, da der raumzeitlichen Zusammenhang nicht definiert werden kann.

3 Ausreißer-Behandlung

Die Ausreißer-Behandlung wird in zeitliche und räumliche Ausreißer unterteilt. In beiden Fällen wird eine robuste Variante des *Z-Score* genutzt um einen Wert als extrem, also als Ausreißer zu werten. Der *Z-Score* vergleicht dabei, ob ein Wert in einem Bereich deutlicher vom Mittelwert abweicht, als üblich. Die robuste Variante nutzt den Median und die mittlere Abweichung vom Median.

Z-Score:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{2}$$

robuster Z-Score:

$$y_i = \frac{x_i - \tilde{X}}{MAD} \tag{3}$$

dabei ist $\tilde{X}$ der Median von X und MAD der Median von $|x_i - \tilde{X}|$.

Alle erkannten Ausreißer werden entfernt und vorerst nicht synthetisch ersetzt. Das Ersetzen passiert implizit bei der räumlichen Interpolation. Um den Einfluss des Ausreißerentfernens zu überprüfen, wird der Sensor, beziehungsweise der Zeitpunkt mit den meisten Ausreißern gewählt. Dabei wird sich auf die PM10 (P1) Daten aus Januar 2018 in Köln beschränkt, um den Rechenaufwand zu minimieren.

3.1 Zeitliche Ausreißer

Bei der Erkennung zeitlicher Ausreißer wird eine Fensterfunktion mit einer Größe von 5 Zeitschritten (je 10 Minuten) genutzt. Durch das kleine Fenster sollen Ausreißer nur für einen zeitlich-lokalen Bereich identifiziert werden, um beispielsweise Sensorrauschen entgegen zu wirken. Bei den zeitlichen Ausreißern wird ein Schwellwert von 3 für den robusten *Z-Score* gewählt. Durch diesen Schwellenwert werden extremere Werte identifiziert, als bei dem üblichen Schwellenwert von 2. Der hier gewählte passive Ansatz soll den Fokus auf die räumliche Ausreißer-Erkennung lenken.
In Abbildung 3.1.1 ist zu sehen, dass größtenteils höhere Werte, im Vergleich zu den umliegenden Messungen, als Ausreißer klassifiziert wurden (Die Bezeichnung ,,Cologne Städte" stammt aus dem Datensatz[2] für das Filtern nach Orten). Bei etwa 2018-01-13 ist eine deutliches Maximum in den Messungen zu sehen. Dabei nehmen die umliegenden Werte gleichmäßig zu und ab. Das führt dazu, dass diese Werte nicht als Ausreißer klassifiziert werden.

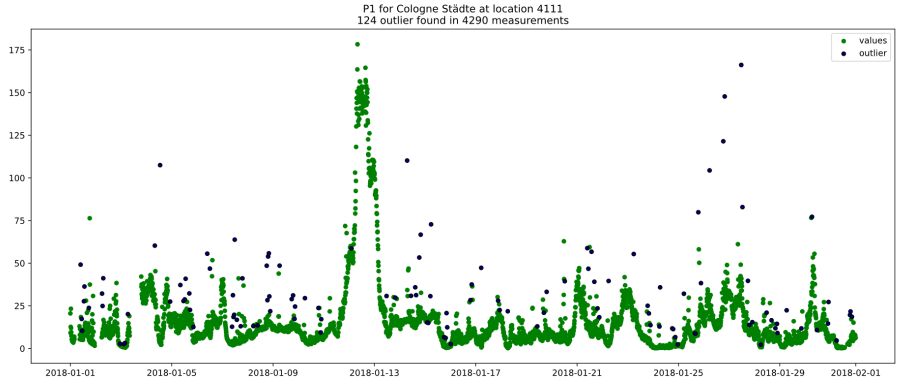

Abb. 3.1.1. Zeitliche Ausreißer-Erkennung bei Feinstaubmessungen einer Messstation (eigene Darstellung)

[2] `https://github.com/isellsoap/deutschlandGeoJSON` (04.10.2021)

3.2 Räumliche Ausreißer

In [Chen et al., 2008] wird eine robuste Methode für räumliche Ausreißer-Erkennung (engl. *spatial outlier detection*) vorgestellt. Für die Definition der Teilmenge, in der Ausreißer identifiziert werden sollen (ähnlich zu dem Fenster bei zeitlichen Ausreißern) werden die Messungen umliegender Station genutzt. In dieser Arbeit wird ein *Kernel*[Ker, 29.01.2021] genutzt, um naheliegende Station in der Nähe zu finden. Im Gegensatz zu *K-Nearest-Neighbours*[KNN, 29.01.2021] wird hierbei nicht immer eine feste Anzahl an Nachbarn betrachtet. In Tabelle 3.3 ist ein Vergleich der beiden Nachbarschafts-Funktionen zu sehen. Dabei ist zu erkennen, dass die *Kernel*-Funktion sichereren Ergebnissen führt. Die Gewichtungen der Nachbarn, welche über den *Kernel* bestimmt werden, werden nicht genutzt. Die Abbildung 3.2.1 zeigt einen Vergleich von Messungen mit und ohne Ausreißern. Rein optisch ist zu erkennen, dass die Messungen in dem rechten Teil ohne Ausreißer deutlich einheitlicher sind. Trotzdem sind noch lokale Ausprägungen sichtbar. Insgesamt werde etwa ein Fünftel der Messungen als Ausreißer klassifiziert. Das ist vermutlich auf den üblichen *Z-Score*-Schwellenwert von 2 zurückzuführen.

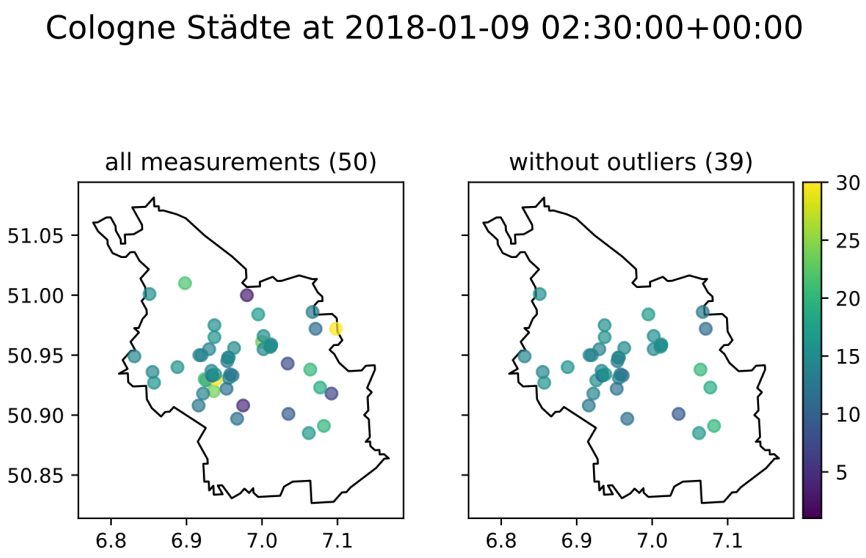

Abb. 3.2.1. Räumliche Ausreißer-Erkennung bei Feinstaubmessungen in Köln (eigene Darstellung)

3.3 Auswirkung auf die Autokorrelation

Im Folgenden werden die Auswirkungen der Ausreißer-Entfernung auf die Autokorrelation vorgestellt. Dafür gibt es drei unterschiedliche Indizes (*Morans'I*[Moran, 1950], *Geary's C*[Geary, 1954] und *Getis and Ord's G*[Ord and Getis, 1995]). Alle Indizes können Werte in einem Bereich von [-1, 1] annehmen. Ein Wert von ,,1" gibt eine positive Autokorrelation an, das heißt, dass nahegelegene Messungen gleich sind. Bei ,,-1" sind nahegelegene Messungen maximal unterschiedlich, also gibt es eine negative Autokorrelation. Bei ,,0" gibt es keine Aussage bezüglich ihrer Autokorrelation. Zusätzlich werden zwei unterschiedliche Funktionen zur Bestimmung der Nachbarschaft verglichen (*K-Nearest-Neighbours (KNN)*[KNN, 29.01.2021] und *Kernel*[Ker, 29.01.2021]). Die Unterschiede zwischen den Funktionen wurden bereits kurz in Unterabschnitt 3.2 vorgestellt. Die Indizes und Nachbarschafts-Funktionen werden in den Default-Einstellungen des Python-Moduls *pysal* genutzt. Das Modul gibt zusätzlich die Möglichkeit, den *p-Wert*(engl. *p-value*) zu simulieren und bietet so eine Schätzung der Signifikanz. Dabei werden Permutationen der Messungen erzeugt und überprüft, wie häufig der Autokorrelations-Index ähnlich oder stärker ausgeprägt ist. Bei einem kleinen *p-Wert* gibt es nur wenige Permutationen, die stärker oder ähnlich ausgeprägt sind, somit ist der berechnete Index signifikant (üblicherweise bei einem *p-Wert* von $\leq 0{,}05$; allerdings können auch höhere Signifikanz-Niveaus gewählt werden).
In Tabelle 3.3 ist zu erkennen, dass mit der Nachbarschafts-Funktion *KNN* tendenziell keine Autokorrelation bei *Moran's I* und *Getis and Ord's G* erkennbar ist, dafür aber umso stärker bei *Geary's C*. Allerdings ist anhand der simulierten *p-Werte* keine Signifikanz erkennbar. Mit der *Kernel*-Funktion ist bei allen Indizes eine positive Autokorrelation zu erkennen. Bei den Daten ohne Ausreißer ist das Signifikanz-Niveau nahe 0,05.

Autokorrelations-Index	Nachbarschafts-Funktion	Statistischer Wert	mit Ausreißern	ohne Ausreißer
Moran's I	KNN	Index	-0.09	0.01
		simulierter p-Wert	0.32	0.37
	Kernel	Index	0.15	0.26
		simulierter p-Wert	0.37	0.03
Geary's C	KNN	Index	0.95	0.97
		simulierter p-Wert	0.38	0.41
	Kernel	Index	0.47	0.69
		simulierter p-Wert	0.4	0.02
Getis and Ord's G	KNN	Index	0.04	0.05
		simulierter p-Wert	0.47	0.48
	Kernel	Index	0.47	0.56
		simulierter p-Wert	0.4	0.06

Tabelle 3.3.1. Autokorrelations-Vergleich für unterschiedliche Nachbarschafts-Funktionen, sowie mit und ohne Ausreißern

3.4 Verbesserung des Krigings

In diesem Unterabschnitt werden die Auswirkungen der Ausreißerentfernung auf die Kriging-Schätzung vorgestellt. In Tabelle 3.4 sind unterschiedliche Metriken dargestellt, welche in Unterabschnitt 7.3 genauer beschrieben sind. Die *Kriging Varianz* wurde bereits in [Braatz, 2019, S. 7] vorgestellt. *Variogramm* R^2 und *max Kriging Varianz* beschreiben, wie gut sich Modell an die Daten angepasst wurde. *MAE*(*Mean absolute error*) und *RMSE*(*Root mean squared error*) beschreiben, wie gut das Modell auf unbekannte Daten generalisiert. Dabei wurden *MAE* und *RMSE* 5-fach kreuzvalidiert. Die Metriken zeigen eine deutliche Verbesserung nach dem Entfernen der Ausreißer. *Variogramm* R^2 zeigt, dass das Variogramm etwa doppelt so gut die Variabilität in den Daten beschreibt, nachdem die Ausreißer entfernt wurden. Auch in den anderen Metriken ist zu sehen, dass die Unsicherheiten in dem Modell reduziert wurden. In Abbildung 3.4.1 und Abbildung 3.4.2 ist die Verbesserung graphisch dargestellt. Hierbei wurde jeweils ein 20x20-Raster mittels Kriging geschätzt. Es ist zu beachten, dass der Wertebereich der *Kriging Varianz* unterschiedlich ist.

Metrik	mit Ausreißern	ohne Ausreißer
Variogramm R^2	0.23	0.44
max Kriging Varianz	33.86	3.93
MAE	3.9	1.6
RMSE	30.05	4.59

Tabelle 3.4.1. Kriging Metrik Vergleich (MAE & RMSE sind 5-fach kreuzvalidiert)

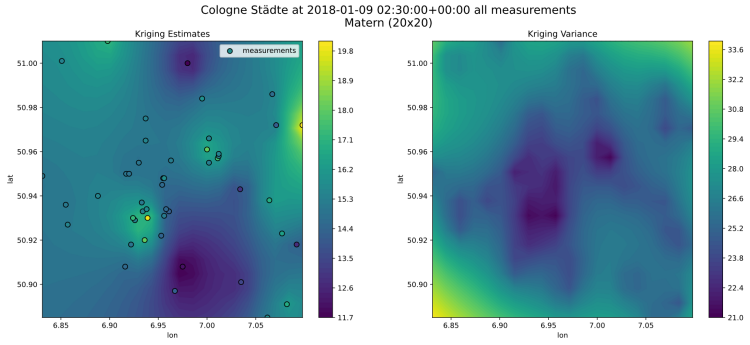

Abb. 3.4.1. Kriging Schätzung (links) und Kriging-Varianz(rechts) auf Basis aller Daten (eigene Darstellung)

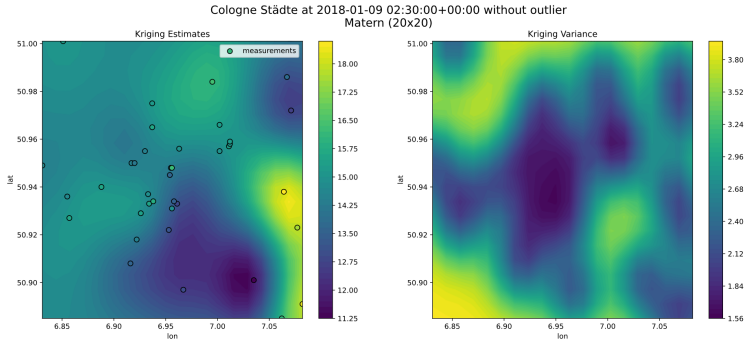

Abb. 3.4.2. Kriging Schätzung (links) und Kriging-Varianz(rechts) auf Basis der Daten ohne Ausreißer (eigene Darstellung)

4 Zusammenhang zu Umwelteinflüssen

In diesem Abschnitt wird der Zusammenhang von Umwelteinflüssen und Feinstaub untersucht. Die betrachteten Daten sind in Tabelle 4 aufgelistet.

Typ	generierte Attribute	Quelle
Zeit im Jahr	day_sin, day_cos	Zeitstempel
Zeit am Tag	hour_sin, hour_cos	Zeitstempel
Straßenverkehr	kfz_A, lkw_A, kfz_B, lkw_B	Bundesanstalt für Straßenwesen
Wochentag	weekday_1.0-weekday_7.0, day_info_w	Bundesanstalt für Straßenwesen
Ferien	day_info_u	Bundesanstalt für Straßenwesen
Feiertage	day_info_s	Bundesanstalt für Straßenwesen
Windgeschwindigkeit	windspeed	Deutscher Wetterdienst
Luftdruck	air_pressure	Deutscher Wetterdienst
Luftfeuchte	humidity	Deutscher Wetterdienst
Niederschlag	precipitation	Deutscher Wetterdienst
Temperatur	temperature	Deutscher Wetterdienst

Tabelle 4.0.1. Übersicht über genutzte Umwelteinflüsse, die dazu generierten Attribute und ihre Quellen

Für ,,Zeit im Jahr" und ,,Zeit am Tag" werden die saisonalen Eigenschaften durch *Sinus / Cosinus*-Repräsentationen beschrieben. So werden einerseits die Jahreszeiten (auf Basis der Tage im Jahr), als auch der Tag-Nacht-Zyklus (auf Basis der Stunden am Tag) dargestellt. Die Daten zum ,,Straßenverkehr"

der *Bundesanstalt für Straßenwesen*[3](BAST) beinhalten Zählerdaten von Bundestraßen und Autobahnen in Deutschland, aufgeteilt auf unterschiedliche Fahrzeugklassen und die Fahrtrichtung. In dieser Arbeit werden nur die KFZ und LKW Zahlen betrachtet. Die Fahrtrichtungen pro Zähler und Fahrzeugart werden aufsummiert. Für die grundsätzliche Überprüfung der Zusammenhänge wird der Mittelwert für jedes Bundesland gebildet. Dadurch ist die Zuordnung zu den einzelnen Feinstaub-Messstation relativ ungenau, dies kann aber in zukünftigen Versuchen präzisiert werden. Die Daten der BAST enthalten weiter Informationen ob ein Tag innerhalb der Woche, am Wochenende, an einem Feiertag oder zur Urlaubszeit ist. Diese Informationen wurden mittels *One-Hot-Encoding* repräsentiert. Die Wetterdaten wurden von dem DWD (Deutscher Wetterdienst)[4] bezogen. Durch das Python-Modul *wetterdienst* kann zu jeder Feinstaubstation die nächste Wetterstation zugeordnet werden und dementsprechend die ortsbezogenen Wetterdaten bezogen werden. Die historischen Wetterdaten sind nur in einem stündlichen Takt abrufbar. Daher wird für die Feinstaubdaten ein stündlicher Mittelwert gebildet.

4.1 Korrelation

Der lineare Zusammenhang zwischen den Attributen wird mit der *Pearson Korrelation* überprüft. Hierbei kann ein Korrelationswert in einem Bereich von [-1, 1] liegen. ,,1" beschreibt einen vollständig positiven linearen Zusammenhang der Attribute (siehe die Korrelation mit sich selbst auf der Hauptdiagonalen in Abbildung 4.1.1). ,,-1" beschreibt einen vollständig negativen linearen Zusammenhang und bei ,,0" gibt es keinen linearen Zusammenhang. In Abbildung 4.1.1 wird die Korrelationsmatrix dargestellt für die Daten von 2018 in Köln. Außerdem werden nur die Daten dargestellt, bei denen der absolute Korrelationswert über 0,1 liegt. Zu beachten ist, dass keine Aussage über die Signifikanz der Werte gegeben wird, da das Python-Modul dazu keine Informationen gibt.
Auffällig sind Temperatur, Luftfeuchtigkeit und Windgeschwindigkeit, die eine negative Korrelation aufweisen zu den Feinstaubmessungen. Dementsprechend sinkt der Feinstaubanteil in der Luft, wenn die Temperatur, Luftfeuchtigkeit und Windgeschwindigkeit steigt. Mögliche Erklärungen des Autors dafür sind die Verwirbelung durch den Wind und Verklumpung des Feinstaubs bei hoher Luftfeuchtigkeit. Allerdings ist der Sensor selbst auch empfindlich auf diese Wettereinflüsse, wie in [lub, 2017, S. 20] beschrieben. Weiter ist die Cosinus-Repräsentation des Jahres positiv korrelierend mit den Feinstaubdaten. Dementsprechend ist die Feinstaubbelastung in den kalten Monaten höher und in den warmen Monaten niedriger. Das kann vermutlich auf die starke negative Korrelation von Luftfeuchtigkeit und Temperatur gegenüber der Cosinus-Repräsentation zurückgeführt werden.

[3] `https://www.bast.de/BASt_2017/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Stundenwerte.html;jsessionid=DB241735D854A44E99B320CB662FC263.live21324?nn=1819490` (04.10.2021)

[4] `https://www.dwd.de/EN/service/copyright/copyright_artikel.html?nn=495490&lsbId=627548` (04.10.2021)

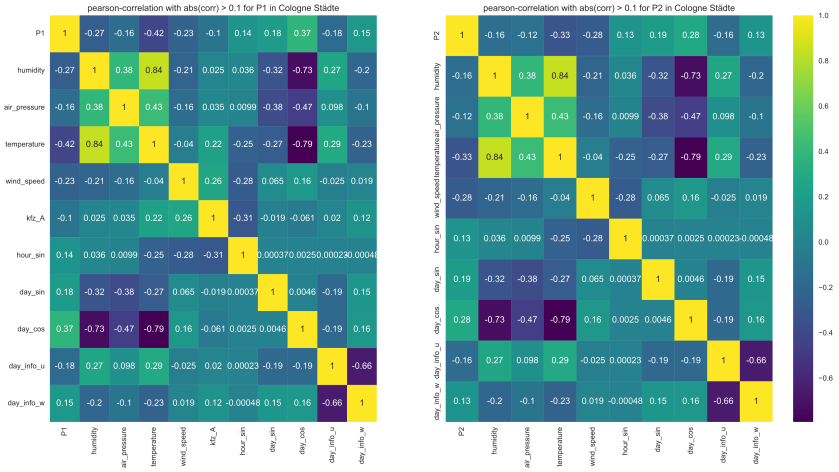

Abb. 4.1.1. Pearson Korrelation zu P1 und P2 für die Attribute mit einer absoluten Korrelation größer 0,1 (eigene Darstellung)

4.2 Clustering

Mithilfe von *Clustering* sollen übergreifende Zusammenhänge zwischen den Attributen bestimmt werden. Für die Nutzung der *Clustering*-Verfahren werden die Daten mittels MinMax-Skalierung auf einen Wertebereich von [0, 1] skaliert. Zuerst wird mit Hilfe der *Elbow-Method* eine mögliche Anzahl an *Clustern* identifiziert(siehe Abbildung 4.2.1. Der *Distortion Score* gibt ein Maß an, wie gut die *Cluster* die Daten abgrenzen. Bei geringer Anzahl k an *Clustern* sinkt der *Distortion Score* stark, bei einer hohen Anzahl geringfügig. An dem ,,Ellenbogen" im Graph gibt es ausreichend *Cluster*, um die Daten abzugrenzen, aber nicht zu viele, dass mögliche *Cluster* erneut geteilt werden.
In Abbildung 4.2.2 werden Cluster bezüglich PM10 (P1) relativ zu den anderen Attributen dargestellt. Die *Cluster* werden mittels *KMeans-Clustering*[Arthur and Vassilvitskii, 2007] definiert (abgesehen von $k = 10$ mit Default-Parametern). Allerdings lassen sich keine klaren Regeln erkennen, nach denen die *Cluster* gebildet wurden. Als eine Alternative zu *KMeans* wird auch *DBSCAN*[Ester et al., 1996] auf die Daten angewendet (mit Default-Parametern). Dabei muss keine Anzahl an *Clustern* angegeben werden, da diese intern bestimmt werden.
Angaben zu dem *DBSCAN*-Ergebnis:

- Geschätzte Anzahl an *Clustern*: 42
- Geschätzte Anzahl an Rausch-Datenpunkten: 4
- Silhouettenkoeffizient: 0.228

Die Anzahl der *Cluster* ist dementsprechend deutlich größer im Vergleich zu dem, was die *Elbow-Method* für *KMeans* ergeben hat. Der Silhouettenkoeffizi-

ent gibt an, wie gut die Cluster von einander differenzierbar sind. Nach [Struyf et al., 1996, S. 10] ist mit einem Koeffizienten $\leq 0{,}25$ keine wesentliche Struktur vorhanden. Dementsprechend ist mit *DBSCAN* kein Struktur in den Clustern vorhanden

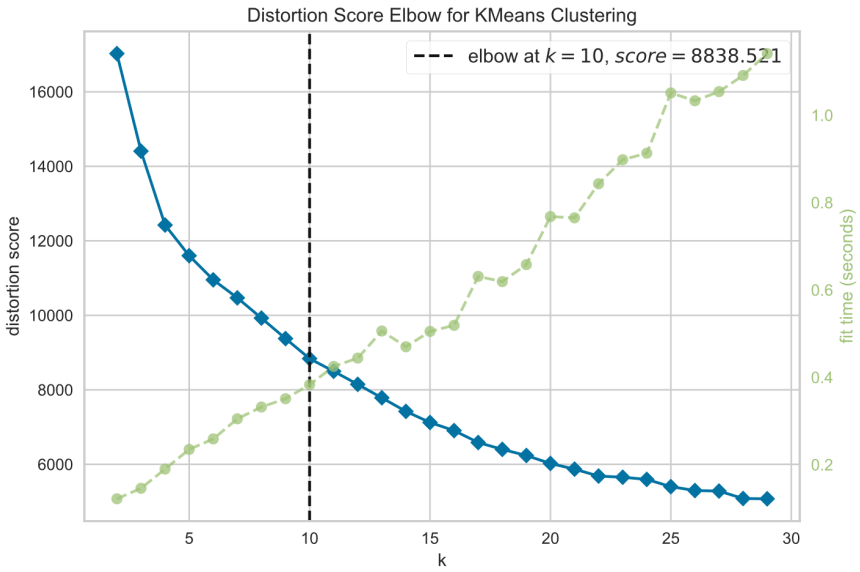

Abb. 4.2.1. Darstellung der *Elbow-Method* für KMeans Clustering. Demnach gibt es ein Optimum bei 10 Clustern (eigene Darstellung)

5 Fazit & Ausblick

In dieser Arbeit konnte grundsätzlich der positive Einfluss der Ausreißerentfernung auf das Interpolieren von Feinstaubdaten mittels Kriging gezeigt werden. Allerdings beschränkt sich diese Arbeit auf kleine Bereiche, während die Effektivität für größere Bereiche noch zu evaluieren ist. Außerdem wurden verhältnismäßig viele Datenpunkte als räumliche Ausreißer identifiziert. Das ist vermutlich auf den üblichen *Z-Score*-Schwellenwert von 2 zurückzuführen. Durch eine Anpassung des Schwellenwertes könnte dem entgegenwirkt werden. Auch könnten die zeitlichen Ausreißer noch durch synthetische Daten, wie zum Beispiel durch den lokalen Median, ergänzt werden. Optimalerweise würde eine Gewichtung der räumlichen und zeitlichen Faktoren genutzt werden um fehlende Daten zu ergänzen. [Zeng et al., 2014] nutzt dafür ein raumzeitliches Variogramm-Model.

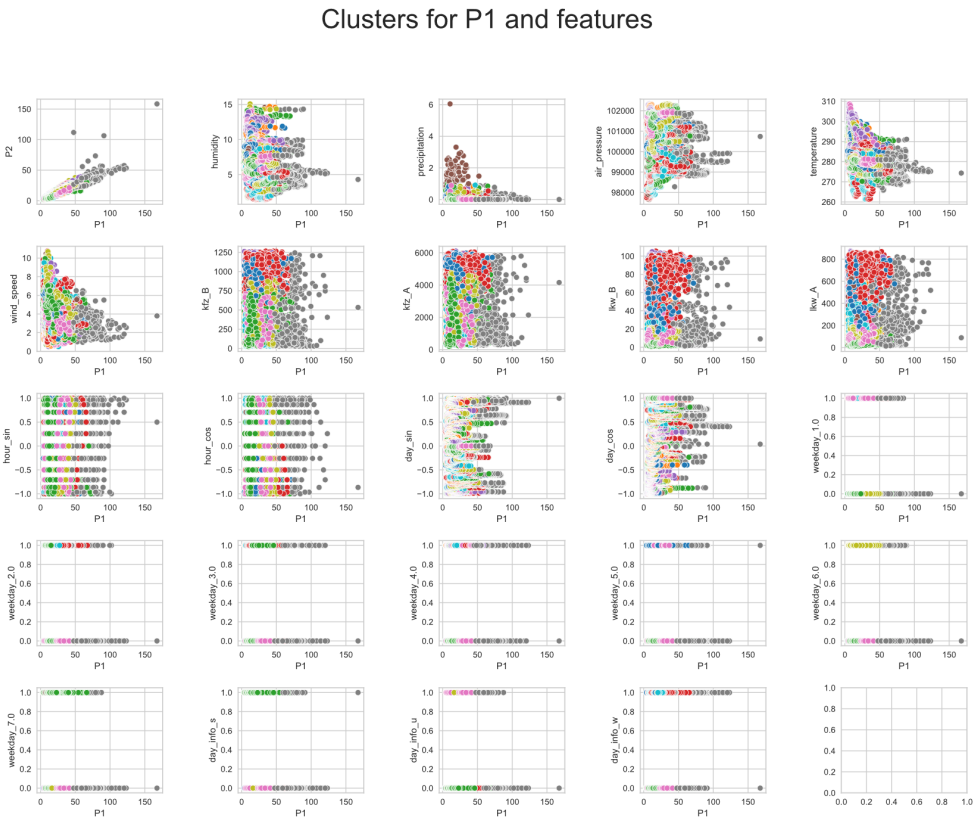

Abb. 4.2.2. KMeans Clustering für 10 Cluster (eigene Darstellung)

Weiter konnten die ersten Zusammenhänge zwischen Feinstaubdaten und anderen Umwelteinflüssen festgestellt werden. Insbesondere der Einfluss von Temperatur, Luftfeuchtigkeit und Windgeschwindigkeit war erkennbar. Die Signifikanz dieser Ergebnisse muss noch festgestellt werden.
Das *Clustering* mit Umwelteinflüssen konnte kein eindeutiges Ergebnis liefern. Es gilt weiterhin zu überprüfen, ob präzisere Einstellung der Distanz- und Dichte-Parameter die Modelle verbessern können. Auch die Interpretation der Cluster steht noch aus.
Bisher wurden einige Vereinfachung bezüglich der Umwelteinflüsse angenommen, wie zum Beispiel, die nur Bundesland-genaue Zuordnung der Verkehrszahlen zu den Feinstaubsensoren. In einer kommenden Arbeit wird diese Zuordnung noch verfeinert, um durch eine präzisere Zuordnung die lokalen Zusammenhänge besser abzubilden.
Diese Arbeit bildet die Basis für weitere Analysen, mit denen Ereignisse erkannt werden können, die zu einer Erhöhung der Feinstaubwerte führen, wie zum Beispiel Chemieunfälle oder Brände. Auch das Aufkommen von Staus könnte identifiziert werden.
In einer zukünftigen Arbeit soll die zeitliche Dimension genauer betrachtet werden, um Vorhersagen zu Feinstaubaufkommen zu machen und die Bewegung von Feinstaub nachzuvollziehen.

6 Danksagung

An dieser Stelle möchte ich mich für das Projekt selbst und die Bereitstellung der Daten bei Sensor.Community bedanken.

Literaturverzeichnis

(13.09.2021) Joblib: running python functions as pipeline jobs — joblib 1.1.0.dev0 documentation. URL https://joblib.readthedocs.io/en/latest/

(2017) Messungen mit dem feinstaubsensor sds011. URL https://pd.lubw.de/90536

(29.01.2021) Spatial weights — python spatial analysis library. URL https://splot.readthedocs.io/en/stable/users/tutorials/weights.html#kernel-weights

(29.01.2021) Spatial weights — python spatial analysis library. URL https://splot.readthedocs.io/en/stable/users/tutorials/weights.html#k-nearest-neighbor-weights

Arthur D, Vassilvitskii S (2007) K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, USA, SODA '07, p 1027–1035

Benjamin Gutzmann, Andreas Motl, Daniel Lassahn, Ilya Kamenshchikov, Max Bachmann, Michael Schrammel (2021) earthobservations/wetterdienst: Start migrating from dogpile.cache to filesystem_spec. https://doi.org/10.5281/ZENODO.5501071

Braatz A (2019) Raumzeitliches data-mining in dynamischen sensorsystemen: Raumzeitliches data-mining in dynamischen sensorsystemen. Bachelor's thesis, Hochschule für angewandte Wissenschaften Hamburg, URL https://reposit.haw-hamburg.de/handle/20.500.12738/9152?&locale=de

Braatz A (2021) Räumliche interpolation von feinstaub-sensordaten mit hilfe von kriging. URL https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2021-proj/braatz.pdf

Chen D, Lu CT, Kou Y, Chen F (2008) On detecting spatial outliers. GeoInformatica 12(4):455–475, https://doi.org/10.1007/s10707-007-0038-8, URL https://link.springer.com/article/10.1007/s10707-007-0038-8

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, KDD'96, p 226–231

Geary RC (1954) The contiguity ratio and statistical mapping. The Incorporated Statistician 5(3):115, https://doi.org/10.2307/2986645

Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall,

Chris Holdgraf, Filipe, François Leblanc (2020) geopandas/geopandas: v0.8.1. https://doi.org/10.5281/ZENODO.3946761
Mirko Mälicke, Egil Möller, Helge David Schneider, Sebastian Müller (2021) mmaelicke/scikit-gstat: A scipy flavoured geostatistical variogram analysis toolbox. https://doi.org/10.5281/ZENODO.4835779
Moran PAP (1950) Notes on continuous stochastic phenomena. Biometrika 37(1/2):17, https://doi.org/10.2307/2332142
Ord JK, Getis A (1995) Local spatial autocorrelation statistics: Distributional issues and an application. Geographical Analysis 27(4):286–306, https://doi.org/10.1111/j.1538-4632.1995.tb00912.x
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830
Rey SJ, Anselin L (2007) PySAL: A Python Library of Spatial Analytical Methods. The Review of Regional Studies 37(1):5–27
Struyf A, Hubert M, Rousseeuw P (1996) Clustering in an object-oriented environment. Journal of Statistical Software 1(4), https://doi.org/10.18637/jss.v001.i04
Zeng Z, Lei L, Hou S, Ru F, Guan X, Zhang B (2014) A regional gap-filling method based on spatiotemporal variogram model of co2 columns. IEEE Transactions on Geoscience and Remote Sensing 52(6):3594–3603, https://doi.org/10.1109/TGRS.2013.2273807

7 Anhang

7.1 Toolchain

In diesem Kapitel wird die beschriebene Toolchain aus [Braatz, 2021] korrigiert beziehungsweise auf Basis der Erfahrungen in der praktischen Umsetzung ergänzt.

SciKit GStat [Mirko Mälicke et al., 2021] wird nicht mehr genutzt, da die selbst implementierte Distanzfunktionen für geographischen Koordinaten von dem Variogramm nicht korrekt an das Kriging übergeben wird und die Parameter für die Distanzfunktion nicht konsistent zwischen den Bestandteilen sind.

GeoPandas [Kelsey Jordahl et al., 2020] wird nun nicht nur für die Visualisierung genutzt, sondern zusätzlich für die Zuordnung von Sensorstationen zu Städten und Bundesländern, sowie für die Filterung nach Städten und Bundesländern.

scikit-learn [Pedregosa et al., 2011] ist ein hoch-abstrahiertes Python-Modul für *Machine-Learning* und *Data-Processing*. In diesem Projekt wird es genutzt für Metriken zur Qualitätsbestimmung des Krigingschätzers.

pysal [Rey and Anselin, 2007] ist die *Python Spatial Analysis Library*, ein *open-source* Python-Module für die Entwicklung von High-Level-Anwendungen für die räumliche Analyse. *PySal* wird genutzt für die Autokorrelations-Indizes und Nachbar-Bestimmung für die räumliche Ausreißer-Bestimmung.

multiprocessing und joblib Das *multiprocessing*-Modul ermöglicht zusammen mit *joblib* [job, 13.09.2021] die Verteilung von Berechnung auf alle Prozessoren (Python ist grundsätzlich auf einen Prozessor beschränkt). Das Modul wird genutzt, um Berechnung, die für alle Zeitstempel oder Sensoren durchgeführt werden müssen, zu beschleunigen. Hauptsächlich betrifft das die Vorverarbeitung und Ausreißer-Erkennung.

wetterdienst [Benjamin Gutzmann et al., 2021] ist eine Service-Bibliothek für den Zugriff auf Wetterdaten. Darüber werden die Wetterdaten des DWD (Deutscher Wetterdienst)[5] abgerufen. Zusätzlich wird die nächste Wetterstationen zu einer Station an einem Koordinatenpunkt ausgegeben. So können die räumlich nächsten Wetterdaten einer Station zugeordnet werden.

7.2 Feinstaub-Datensatz

Der Datensatz enthält 12 Spalten:

1. sensor_id - numerische Identifikation für den Sensor
2. sensor_type - Typ des Sensors (hier nur 'SDS011')
3. location - numerische Identifikation des Sensorknotens
4. lat - Breitengrad des Sensorknotens mit drei Nachkommastellen
5. lon - Längengrad des Sensorknotens mit drei Nachkommastellen
6. timestamp - Zeitpunkt der Messung; größtenteils in ISO8601, aber auch in Unixzeit in Millisekunden
7. P1 - PM10 in ppm (Parts per million), teilweise aber auch als Text (z.B. 'PM10', '['PM10']', '%vla2%')
8. durP1 - Überrest des PDD42NS; wird nicht betrachtet
9. ratioP1 - Überrest des PDD42NS; wird nicht betrachtet
10. P2 - PM2,5 in ppm (Parts per million), teilweise aber auch als Text (z.B. 'PM25', '['PM25']', '%vla2%')
11. durP2 - Überrest des PDD42NS; wird nicht betrachtet
12. ratioP2 - Überrest des PDD42NS; wird nicht betrachtet

7.3 Kriging Metriken

Metriken zur Validierung des Kriging-Schätzers:
Der mittlere absolute Fehler (*engl. mean absolute error*; MAE):

$$MAE = \frac{1}{l} \cdot \sum_{j=1}^{l} |\hat{z}(s_j) - z^*(s_j)| \tag{4}$$

[5] `https://www.dwd.de/EN/service/copyright/copyright_artikel.html?nn=495490&lsbId=627548` (04.10.2021)

dabei sind $\hat{z}(s_j)$ die geschätzten Werte und $z^*(s_j)$ die Beobachtungen an Validierungspunkten.

Die Wurzel aus dem mittleren, quadrierten Fehler (*engl. root mean square error*; RMSE):

$$RMSE = \sqrt{\frac{1}{l} \cdot \sum_{j=1}^{l} [\hat{z}(s_j) - z^*(s_j)]^2} \tag{5}$$

dabei ist l die Anzahl der Validierungspunkte.

Das Bestimmtheitsmaß R^2 gibt den Anteil der Variabilität in den Beobachtungen an, die durch das Vorhersagemodell beschrieben wird:

$$R^2 = \frac{\sum (\hat{z} - \bar{z})^2}{\sum (z^* - \bar{z})^2} \tag{6}$$

dabei ist $\bar{z}$ der Mittelwert über alle Beobachtungen