

BACHELOR ARBEIT

# Schlüsselfaktoren zur Steigerung des Photovoltaik-Ausbaus in Deutschland

Warum die Solar-Förderung in Deutschland neu gedacht werden muss

zur Erlangung des Grades Bachelor of Science

Eingereicht von: Conny Brintzinger  
Köpenicker Allee 9  
10318 Berlin  
[conny@nexus-six.de](mailto:conny@nexus-six.de)

Matrikelnummer: 190 044  
Fachsemester: 6  
Studiengang: Data Science & Business Analytics  
Datum Abgabe: 25.04.2024  
Gutachter: Prof. Dr. Marcel Hebing

# Inhaltsverzeichnis

Danksagung .....	1
Eigenständigkeitserklärung .....	2
Abstract .....	3
Abbildungs-Verzeichnis .....	6
Abkürzungsverzeichnis .....	9
1. Einleitung und Forschungsstand .....	10
1.1. Deutschland auf dem Weg zur Klimaneutralität .....	10
1.2. Forschungslücke, konkrete Forschungsfragen und Zielsetzung .....	11
2. Daten und Methoden .....	12
2.1. Methodisches Vorgehen .....	12
2.2. Erhebung, Beschreibung und Bereinigung der Daten.....	13
2.2.1 Marktstammdaten.....	13
2.2.2 Sozioökonomische Daten.....	14
2.2.3 Weitere Datenerhebung mit Bezug zum Solarausbau.....	15
2.2.4 Zeitreihen-Datenerhebung .....	15
2.3. Räumliche Zuordnung der Daten.....	17
2.4. Sicherung der bereinigten Daten als Data-Warehouse.....	18
2.5. Datenintegration .....	18
2.6. Transformationen.....	19
2.7. Entwicklungsumgebung .....	19
2.8. Analysemethoden .....	20
3. Ergebnisse .....	23
3.1. Sozioökonomische Einflussfaktoren .....	23
3.2. Zeitreihen-Analysen und SARIMAX-Modellierung .....	24

4. Diskussion und Handlungsempfehlung .....	31
5. Literaturverzeichnis .....	35
6. Anhang .....	42
6.1. Weitere Auswertungen und Analyse-Ergebnisse.....	42
6.2. README-Datei.....	<b>Fehler! Textmarke nicht definiert.</b>
6.4. Requirements-Dateien .....	<b>Fehler! Textmarke nicht definiert.</b>
6.5. Jupyter Notebooks (PDF) .....	<b>Fehler! Textmarke nicht definiert.</b>

## Danksagung

Ich möchte die Gelegenheit nutzen, um den Dozenten der Digital Business University meinen aufrichtigen Dank auszusprechen. Euer Data Science & Business Analytics Studiengang hat mich nicht nur fachlich, sondern auch persönlich bereichert. Besonders möchte ich Professor Prof. Dr. Marcel Hebing meine Anerkennung aussprechen. Durch seine inspirierende Lehre hat er es geschafft, die Welt der Daten und Algorithmen mit Leben zu füllen, meine Neugier zu wecken und mich ermutigt, Zahlen kritisch zu hinterfragen.

Ein herzliches Dankeschön geht an meine geschätzten Kollegen bei EVIDEN, die mit interessanten Gesprächen über Solarenergie und wertvollen praktischen Tipps zum Erfolg meiner Bachelorarbeit beigetragen haben.

Zu guter Letzt möchte ich meinem Partner Daniel von Herzen danken. Mit deiner Motivation und unermüdlichen Unterstützung warst du während des gesamten Studiums sowie bei der Erstellung dieser Arbeit mein Fels in der Brandung.

## Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Alle sinngemäß und wörtlich übernommenen Textstellen aus fremden Quellen wurden kenntlich gemacht.

Berlin, den 25.04.2024



Conny Brintzinger

## Abstract

Der Solarausbau in Deutschland spielt eine entscheidende Rolle bei der Erreichung des im novellierten Klimaschutzgesetz 2021 verankerten Ziels der Klimaneutralität bis zum Jahr 2045.

Die vorliegende Arbeit untersucht den Zusammenhang sozioökonomischer Einflussfaktoren auf die Anzahl installierter Photovoltaik-Anlagen. Darüber hinaus wird die historische Entwicklung des Solarausbaus in Bezug zu temporalen Ereignissen und Einflussfaktoren betrachtet mit dem Ziel, die entscheidenden Dynamiken und Determinanten herauszuarbeiten, mögliche Problemfelder für den zukünftigen Solarausbau aufzuzeigen und Handlungsempfehlungen abzuleiten.

Solaranlagen sind heute effizient, preisgünstig und wirtschaftlich, was den Solarausbau fördert. Mit sinkender Einspeisevergütung und steigenden Strompreisen haben sich jedoch die Anreize für den Einbau einer Solaranlage über die letzten Jahre verändert und einen Paradigmenwechsel von Volleinspeisung hin zur Teil einspeisung vollzogen. Diesen zeitbezogenen Veränderungen und maßgeblichen Einflussfaktoren, wie Eigentumsquote oder Bevölkerungsdichte werden aktuell zu wenig berücksichtigt, was zu einer ineffektiven Förderpraxis beiträgt. Es wird daher eine Methodik zur Erforschung potenzieller Einflussfaktoren vorgestellt, auf deren Basis zeitgemäße Solar-Fördermodelle entwickelt werden können, die in einem volatilen Umfeld flexibel und kosteneffizient agieren.

# Tabellen-Verzeichnis

<b>Tabelle 1:</b> SARIMAX Modelle und inkludierte Prädiktoren.....	22
<b>Tabelle 2:</b> Durchschnittliche installierte Brutto-Leistung nach Bundesland in absteigender Rangfolge. ....	49
<b>Tabelle 3:</b> TOP 15 Postleitzahlengebiete mit dem höchsten prozentualer Zuwachs an PV-Anlagen zwischen 2018 und 2023. ....	50
<b>Tabelle 4:</b> Prozentualer Zuwachs an PV-Anlagen in Berlin zwischen 2018 und 2023, sortiert nach Lage der installierten Anlagen. ....	51
<b>Tabelle 5:</b> BOTTOM 15 Postleitzahlengebiete mit dem niedrigsten prozentualen Zuwachs an PV-Anlagen zwischen 2018 und 2023. ....	52
<b>Tabelle 6:</b> Top-20 Gemeinden mit der höchsten Anzahl PV-Anlagen pro 10.000 Einwohner. ....	53
<b>Tabelle 7:</b> BOTTOM-20 Gemeinden mit der niedrigsten Anzahl PV-Anlagen pro 10.000 Einwohner. ....	54
<b>Tabelle 8:</b> Top-20 Gemeinden nach installierter Bruttolleistung (kWh) pro 10.000 Einwohner. ....	55
<b>Tabelle 9:</b> Bottom-20 Gemeinden nach installierter Bruttolleistung (kWh) pro 10.000 Einwohner. ....	56
<b>Tabelle 10:</b> Ranking installierte Bruttolleistung in kWh nach Bundesländern im Zeitraum von 2005 bis 2023. ....	70
<b>Tabelle 11:</b> Ranking installierte Bruttolleistung in kWh pro 10.000 Einwohner nach Bundesländern im Zeitraum von 2005 bis 2023. ....	72
<b>Tabelle 12:</b> Ergebnisse der OLS-Regression (Anzahl_PV_Anlagen ~ Eigentumsquote + Bevoelkerungsdichte) .....	74
<b>Tabelle 13:</b> Ergebnisse der OLS-Regression (Anzahl_PV_Anlagen ~ Strom_Verbraucherpreis + Einspeiseverguetung_kleine_PV_Anlagen).....	75
<b>Tabelle 14:</b> Feature Importance der sozioökonomischen Einflussfaktoren in Bezug auf die Anzahl installierter PV-Anlagen. ....	77
<b>Tabelle 15:</b> Vergleich der Sarimax-Modelle anhand RMSE, MAPE und MAE. 84	
<b>Tabelle 16:</b> Summary Sarimax Modell 1.....	85

<b>Tabelle 17:</b> Summary Sarimax Modell 2 .....	86
<b>Tabelle 18:</b> Summary Sarimax Modell 3 .....	87
<b>Tabelle 19:</b> Summary Sarimax Modell 4 .....	88
<b>Tabelle 20:</b> Ergebnisse der OLS-Regression (GesamtBruttoleistung ~ Strompreis + Einspeiseverguetung + Fukushima + Solardeckel) .....	89

# Abbildungs-Verzeichnis

<b>Abbildung 1:</b> Entwicklung der jährlich installierten PV-Bruttoleistung gegenüber Strompreis und Einspeisungsvergütung .....	25
<b>Abbildung 2:</b> Anzahl installierter PV-Anlagen nach Voll- und Teileinspeisung seit 2000 .....	26
<b>Abbildung 3:</b> Historischer Überblick über den Solarausbau in Deutschland von 2000 - 2023 .....	27
<b>Abbildung 4:</b> Zeitreihen-Zerlegung der monatlich installierten Gesamt-Bruttoleistung in Trend, Saisonalität und Residuen.....	29
<b>Abbildung 5:</b> Vergleich der Vorhersage-Genauigkeit der vier Sarimax-Modelle .....	30
<b>Abbildung 6:</b> Anzahl der installierten PV-Anlagen pro Betreiber.....	42
<b>Abbildung 7:</b> Histogramm zur Verteilung der Anzahl PV-Anlagen pro Betreiber.	43
<b>Abbildung 8:</b> Anzahl der installierten PV-Anlagen nach Bevölkerungsdichte.	44
<b>Abbildung 9:</b> Anzahl der installierten PV-Anlagen nach Eigenheimquote.....	45
<b>Abbildung 10:</b> Anteil von Photovoltaik an der Brutto-Stromerzeugung in Deutschland .....	46
<b>Abbildung 11:</b> Installierte Gesamtleistung aller PV-Anlagen nach Monat der Inbetriebnahme .....	47
<b>Abbildung 12:</b> Installierte Gesamtleistung aller PV-Anlagen nach Monat der Registrierung .....	47
<b>Abbildung 13:</b> Anzahl der in Betrieb genommenen PV-Anlagen pro Jahr nach Bundesland .....	48
<b>Abbildung 14:</b> Durchschnittliche installierte Bruttoleistung pro Jahr und Bundesland .....	48
<b>Abbildung 15:</b> Entwicklung der durchschnittlichen Leistung pro installiertem PV-Modul von 2005 bis 2023. ....	57
<b>Abbildung 16:</b> Verteilung der Nutzungsbereiche aller installierten PV-Anlagen von 2005 bis 2023. ....	58

<b>Abbildung 17:</b> Verteilung der Lage aller installierten PV-Anlagen von 2005 bis 2023 .....	59
<b>Abbildung 18:</b> Entwicklung der Hauptausrichtung der installierten PV-Anlagen von 2005 bis 2023.....	60
<b>Abbildung 19:</b> Entwicklung der Einspeisungsarten Voll- und Teileinspeisung für zwischen 2005 und 2023 installierte PV-Anlagen.....	61
<b>Abbildung 20:</b> Entwicklung der Leistungsbegrenzung der installierten PV-Anlagen von 2005 bis 2023.....	62
<b>Abbildung 21:</b> Entwicklung nach Installation von PV-Anlagen mit Stromspeichern und mit oder ohne gemeinsamen Wechselrichter von 2005 bis 2023. ....	63
<b>Abbildung 22:</b> Installierte Bruttolleistung in kWh nach Postleitzahlengebiet im Zeitraum 2005 bis 2008.....	64
<b>Abbildung 23:</b> Installierte Bruttolleistung in kWh nach Postleitzahlengebiet im Zeitraum 2009 bis 2012.....	65
<b>Abbildung 24:</b> Installierte Bruttolleistung in kWh nach Postleitzahlengebiet im Zeitraum 2013 bis 2018.....	66
<b>Abbildung 25:</b> Installierte Bruttolleistung in kWh nach Postleitzahlengebiet im Zeitraum 2019 bis 2023.....	67
<b>Abbildung 26:</b> Installierte Bruttolleistung in kWh nach Postleitzahlengebiet im Gesamt-Zeitraum 2005 bis 2023.....	68
<b>Abbildung 27:</b> Installierte Bruttolleistung in kWh nach Bundesländern im Gesamt-Zeitraum 2005 bis 2023.....	69
<b>Abbildung 28:</b> Installierte Bruttolleistung pro 10.000 Einwohner in kWh nach Bundesländern im Gesamt-Zeitraum 2005 bis 2023. ....	71
<b>Abbildung 29:</b> Korrelationsmatrix aller numerischen Variablen ab einer Korrelation von 0,4. ....	73
<b>Abbildung 30:</b> Entscheidungsbaum zum Einfluss sozioökonomische Einflussfaktoren auf die Anzahl installierter Solaranlagen in einer Region.....	76
<b>Abbildung 31:</b> Korrelationsmatrix sozioökonomische Einflussfaktoren.....	78
<b>Abbildung 32:</b> Korrelationsmatrix Kosten, Vergütungen und Ereignisse. ....	80

<b>Abbildung 33:</b> Visualisierung der Zeitreihen GesamtBruttoleistung, Anzahl_Volleinspeisung und Anzahl_Teileinspeisung.....	81
<b>Abbildung 34:</b> Visualisierung der Zeitreihen Einspeiseverguetung, Strompreis und PV_Kosten_Euro_pro_kWp.....	82
<b>Abbildung 35:</b> Visualisierung der Zeitreihen Fukushima, Ukrainekrieg und Solardeckel.....	83
<b>Abbildung 36:</b> Vergleich der Sarimax-Modelle anhand Vorhersagen auf die Testdaten. ....	84

## Abkürzungsverzeichnis

AGS-5	5-stelliger Amtlicher Gemeindeschlüssel
AGS- 8	8-stelliger Amtlicher Gemeindeschlüssel
AIC	Akaike Information Criterion
ARS	12-stelliger Amtlicher Regionalschlüssel
DWH	Data-Warehouse
EEG	Erneuerbare-Energien-Gesetz
EU	Europäische Union
GWh	Gigawattstunde (1 GWh entsprechen 1.000.000 kWh)
km <sup>2</sup>	Quadratkilometer
kWh	Kilowattstunde
kWp	Kilowattpeak
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MWh	Megawattstunde (1 MWh entsprechen 1.000 kWh)
OLS	Ordinary Least Squares
PLZ	Postleitzahl
PV	Photovoltaik
R <sup>2</sup>	Bestimmtheitsmaß, Gütemaß der linearen Regression
RMSE	Root Mean Squared Error
Sarimax	Seasonal Autoregressive Integrated Moving Average Exogenous

# 1. Einleitung und Forschungsstand

## 1.1. Deutschland auf dem Weg zur Klimaneutralität

Der Klimawandel wird heute als existenzielle Bedrohung wahrgenommen, der man vereint als Europäische Union mit ehrgeizigen Zielsetzungen zur Minimierung der Treibhausgas-Emissionen und Klimaschutzmaßnahmen begegnet. Im „Übereinkommen von Paris“ haben sich die EU-Staaten verpflichtet, bis 2050 klimaneutral zu werden, um die Folgen einer drohenden globalen Erwärmung von über 1,5°C über dem vorindustriellen Niveau zu vermeiden. Der wirtschaftliche und gesellschaftliche Wandel soll in gerechter, sozial ausgewogener, fairer und kosteneffizienter Weise erreicht werden (Bundesministerium für Wirtschaft und Klimaschutz, 2023).

Deutschland will bereits 2045 Klimaneutralität erreichen, so das ambitionierte Ziel, das im novellierten Klimaschutzgesetz 2021 (Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit, 2021) festgehalten wurde. Seitdem wird um den besten Weg und das richtige Maß gerungen. Diese tiefgreifende Transformation des gesamten Energieversorgungssystems in weniger als 25 Jahren zu realisieren, stellt einen enormen Kraftakt dar und ist an die Erreichung von Zwischenzielen geknüpft.

Bei der Dekarbonisierung der Energiewirtschaft und der Elektrifizierung des Energieverbrauchs kommt dem Ausbau der Photovoltaik eine zentrale Rolle zu (Ammon, Bruns, & Semerow, 2019). Zur Erreichung der Ziele müssen bis Ende 2030 pro Monat PV-Anlagen mit einer Leistung von 1,6 GWh zugebaut werden. Der Ausbau soll nicht linear erfolgen, sondern im Zeitverlauf ansteigen, wie sich aus den im EEG formulierten Zwischenzielen sowie der kürzlich veröffentlichten Photovoltaik-Strategie der Bundesregierung ergibt. Der Plan sieht vor, dass im Jahr 2023 9 GWh dazu kommen sowie in den Jahren 2024 und 2025 13 bzw. 18 GWh. Von 2026 an sollen jährlich 22 GWh zugebaut werden (Bundesministerium für Wirtschaft und Klimaschutz, 2023).

Doch sind die gesteckten Ziele realistisch und erreichbar? Welche Faktoren haben den Photovoltaik-Ausbau in der Vergangenheit beeinflusst und welche Schlüsse für die Zukunft können daraus abgeleitet werden?

## 1.2. Forschungslücke, konkrete Forschungsfragen und Zielsetzung

Mit der Abkehr von fossilen Energieträgern und zunehmender Elektrifizierung wird der Strombedarf in den nächsten Jahren ansteigen- insbesondere in den Bereichen Mobilität und Wärme (Bundesverband Solarwirtschaft e.V., 2023). Zahlreiche Studien zeigen die Potenziale im Ausbau der erneuerbaren Energien auf, belegen deren Machbarkeit und modellieren unterschiedliche Pfade zur Erreichung der Klimaneutralität bis 2045. Gleichzeitig wird auf Faktoren hingewiesen, die den PV-Ausbau in den nächsten Jahren behindern und damit die gesteckten Ziele gefährden könnten (Wirth, 2024; Ariadne-Konsortium, 2021; Quaschning, et al., 2021).

Je nach Intension und Adressat der Veröffentlichung variiert die Argumentation über mögliche Hemmnisse beim PV-Ausbau und wie diesen zu begegnen ist. Es liegt jedoch keine vollständige Analyse vor, die alle Aspekte der Solarausbau-Historie erfasst. Ziel der Arbeit ist es daher, die historischen Daten zum Solarausbau zu analysieren, die entscheidenden Entwicklungen und Einflussfaktoren zu ermitteln und mögliche Problemfelder in der Zukunft aufzuzeigen.

### Forschungsfragen:

**Forschungsfrage 1:** Welcher Kausalzusammenhang besteht zwischen sozi-ökonomischen Einflussfaktoren und der Anzahl errichteter PV-Anlagen in Deutschland?

**Forschungsfrage 2:** Welche temporalen Einflüsse erklären den Entwicklungsverlauf des PV-Ausbaus in Deutschland zwischen 2000 und 2023?

## 2. Daten und Methoden

### 2.1. Methodisches Vorgehen

#### **Teil 1: Sozioökonomische Einflussfaktoren**

Um sich den Forschungsfragen anzunähern, werden zunächst sozioökonomische Faktoren, wie Einkommensdaten, Bildungs- und Altersstruktur, Arbeitslosenquote, Kriminalitätsraten, Bevölkerungsdichte und -entwicklung, Eigenheimquote, Bundestagswahlergebnisse und Migrationshintergrund als potenzielle Einflussfaktoren für die Errichtung einer PV-Anlage bewertet. Es werden Tests auf Korrelation, ein Entscheidungsbaum sowie eine Feature Importance Analyse angestoßen, um die Bedeutung der Variablen auf die Zielgröße *Anzahl installierte PV-Anlagen* zu erklären und zu quantifizieren.

#### **Teil 2: Zeitreihen-Analysen**

Im zweiten Schritt wird der Solar-Ausbau in Deutschland in seinem zeitlichen Verlauf betrachtet. Eine umfassende explorative Datenanalyse des Marktstamm-datenregisters untersucht, wie sich die Anzahl an PV-Anlagen und die installierte Leistung zwischen 2000 und 2023 entwickelt haben mit Fokus auf Trends, saisonale Effekte und regionale Unterschiede. Durch Plotten der technischen Merkmale, wie Einspeisungsart, Leistungsbegrenzung, Lage oder Nutzungsbereiche im Zeitverlauf werden Veränderungen struktureller Art sichtbar gemacht. Eine Zeitreihen-Zerlegung betrachtet die Entwicklung der installierten Bruttoleistung getrennt nach Trend, saisonaler Komponente und Residuen.

Auf Basis der gewonnenen Erkenntnisse werden Strukturbrüche identifiziert und mittels Literaturrecherche in einen zeitlichen Zusammenhang mit Ereignissen, Preisentwicklungen und politischen Entscheidungen gesetzt, die als Ursachen für die beobachteten Veränderungen in Betracht kommen. Die Relevanz dieser potenziellen Einflussfaktoren wird durch Modellierung als multivariate Zeitreihenanalyse und Ausgabe der Fehler-Metriken evaluiert.

## 2.2. Erhebung, Beschreibung und Bereinigung der Daten

Im folgenden Abschnitt wird vorgestellt, welche Daten erhoben werden, welche Datenquellen genutzt werden und wie die Datenqualität zu beurteilen ist.

### 2.2.1 Marktstammdaten

Als zentrale Datenquelle dient das **Marktstammdatenregister** (Bundesnetzagentur, 2023), in dem konstante Daten wie Standort, Inbetriebnahme-Datum, Leistung und technische Kennzahlen von Anlagen zur Strom- und Gaserzeugung erfasst werden. Für den ersten Teil der Analysen wird der Stand aller bis zum 01.07.2023 in Betrieb genommenen Solar-Einheiten betrachtet, insgesamt 3.180.113 registrierte PV-Anlagen. In 66 Spalten sind umfassende Informationen zu Inbetriebnahme-Datum, Standort, Ausrichtung, Leistung und Einspeisungsart der Solaranlagen hinterlegt. Für jede Anlage besteht unabhängig von der Größe eine Registrierungspflicht, was eine hohe Datenqualität impliziert. Eine aktuelle Studie zeigt jedoch, dass viele Anlagenregistrierungen im Kleinstanlagenbereich (sogenannte steckerfertige Erzeugungsanlagen, Plug-In- oder Balkon-PV-Anlagen) fehlerhaft sind in Hinblick auf Lage, Art der Einheit und Einspeisungsart. Es wird zudem eine Dunkelziffer nicht gemeldeter Anlagen vermutet (EUPD Research, 2023). Seit 2021 haben Kleinstanlagen an Bedeutung gewonnen und bilden nach Anlagen auf Dächern und Gebäuden die zweithäufigste Art neu installierter PV-Anlagen (siehe Anhang, Abbildung 2-16). Insofern ist ab diesem Zeitpunkt von nicht näher quantifizierbaren Mängeln in der Datenqualität für neu installierte Kleinstanlagen auszugehen.

Für die Analysen nicht relevante Spalten werden aus dem Datensatz entfernt. Zudem werden 7.078 stillgelegte PV-Anlagen, 2 Anlagen mit fehlenden Standort-Informationen sowie 23 Anlagen, die sich außerhalb Deutschlands befinden, gelöscht. Fehlende Werte werden entsprechend der inhärenten Charakteristika der Variablen ersetzt, ungültige Postleitzahlen berichtigt und fehlende Inbetriebnahme-Daten mit dem Registrierungsdatum aufgefüllt.

Die einzelnen Schritte werden im Notebook *01\_ETL\_DWH\_Marktstammdaten* ausgeführt und in der *Readme-Datei* detailliert dokumentiert. Der bereinigte Datensatz *marktstammdaten* enthält 3.173.010 Zeilen und 26 Spalten.

Zur Erstellung der Zeitreihen-Analysen wird in Notebook *08\_Time\_Series\_Simulation* eine aktualisierte Version des Markstammdatenregisters verwendet, die Werte von 01/2000 bis 12/2023 enthält und ebenfalls die oben beschriebenen Schritte zur Bereinigung der Daten durchläuft. Der bereinigte Datensatz *marktstammdaten\_2023* enthält 3.722.267 Zeilen und lediglich fünf für die Zeitreihenanalysen relevante Spalten (*Inbetriebnahmedatum*, *AnlagenbetreiberMastrNummer*, *Bruttoleistung*, *Einspeisungsart* und *Lage*).

Die Tabellen *marktstammdaten* und *marktstammdaten\_2023* werden im Data-Warehouse gesichert.

## 2.2.2 Sozioökonomische Daten

Im nächsten Schritt, der in Notebook *02\_ETL\_DWH\_Sozioökonomische\_Daten* erfolgt, werden **sozioökonomische Daten** aus unterschiedlichen Open-Data-Portalen zusammengestellt, wie **Altersstruktur** (Zensus Datenbank, 2011), **Arbeitslosenquote** (GovData, 2018), **Ausländeranteil** (Deutschlandatlas, 2023) und **Migrationshintergrund** (Zensus Datenbank, 2011), **Bevölkerungsdichte und -entwicklung** (Statistisches Bundesamt, Destatis.de, 2021; Bundesamt für Kartographie und Geodäsie, 2020), **Bildungs- und Familienstand** (Zensus Datenbank, 2011; Deutschlandatlas, 2023), **Bundeswahlergebnisse 2021** (Statistisches Bundesamt, 2021), **Eigenheimquote** (Zensus Datenbank, 2011), **Einkommensentwicklung** (Statistische Ämter des Bundes und der Länder, Statistikportal.de, 2022), **Bruttoinlandsprodukt** (Deutschlandatlas, 2023), **Gini-Index** für Einkommensungleichheit (Statistische Ämter des Bundes und der Länder, Statistikportal.de/de, 2022) und **Kriminalitätsdelikte** (Deutschlandatlas, 2023).

Die Datenqualität der sozioökonomischen Daten ist für die vorgesehene Verwendung teilweise als ungenügend zu bewerten, da die Quellen häufig veraltet sind, fehlende Werte beinhalten oder die Datenverfügbarkeit auf granularer Ebene nicht ausreicht. Der Zensus-Datenbank entnommene Daten zu Altersstrukturen, Migrationshintergrund, Eigenheimquote und Familienstand stammen aus dem Jahr 2011 und sind somit nicht aktuell. Die Arbeitslosenquote und der Gini-Index sind lediglich auf Bundesländer-Ebene verfügbar. Insofern kann keine konsistente Datenbasis erzeugt werden, sowohl in Hinblick auf Zeitpunkt als auch Region der Datenerhebung.

### 2.2.3 Weitere Datenerhebung mit Bezug zum Solarausbau

In Notebook *03\_ETL\_DWH\_Solar\_Wetter\_Daten* werden weitere Daten zu **Sonnenstunden** (Deutscher Wetterdienst, 2023), **E-Mobilität** und **Anteil erneuerbarer Heizungen bei Neubauten** (Deutschlandatlas, 2023) zusammengetragen.

Um jeder Solaranlage eine Anzahl Sonnenstunden zuzuweisen, wird dem Centroid des Postleitzahlengebietes (Fadhel, 2023) mittels Haversine Distance Equation (Ahmed, 2020) Längen- und Breitengrad der jeweils nächstgelegenen Wetterstation zugeordnet. Die Daten für E-Mobilität und erneuerbare Heizungen beinhalten relative Werte bezogen auf Einwohnerzahl und Region in 2021/2022.

Es gibt keine fehlenden Werte. Die Datenquellen sind vertrauenswürdig.

### 2.2.4 Zeitreihen-Datenerhebung

Für Zeitreihenanalysen werden im Notebook *03\_ETL\_DWH\_Solar\_Wetter\_Daten* Daten zur Entwicklung von **Strompreis** (BMWK, 2022; BDEW, 2023), **Ein speisevergütung** (Solarenergie Förderverein, 2017; Bundesnetzagentur, 2024), **EEG-Umlage** (REWAG, 2022; Weitekamp, 2022), **PV-Kosten** (Vöpel, 2024; Our World in Data, 2023) und **potenziell relevanten Ereignissen** zwischen 2000 und 2023, wie der Nuklear-Katastrophe von **Fukushima** (LpB BW, 2023), dem **Solardeckel** (Leuschner, Energie-chronik\_1, 2018) oder der durch den **Ukraine-**

**Krieg** ausgelösten Energiekrise (Grimm, Groß, Marxsen, & Schwarz, Wirtschaftsdienst.eu, 2023) zusammengetragen.

Die Daten zu **Einspeisevergütung, Strom- und PV-Kosten** stehen nicht in öffentlichen Download-Portalen zur Verfügung. Sie werden daher aus unterschiedlichen Quellen zusammengetragen und auf Kohärenz geprüft.

Der sogenannte **Solardeckel** wurde im Juni 2012 von der Bundesregierung beschlossen und sah ein Auslaufen der Solar-Förderung und Stopp der Einspeisevergütung bei Erreichen einer installierten Gesamt-Bruttoleistung von 52.000 MWh vor. Da im August 2020 mit Erreichung der Marke zu rechnen war, wurde der Solardeckel im Juni 2020 in einer Neufassung des EEG gestrichen (Leuschner, energie-chronik\_3, 2020).

Die am 11.03.2011 durch einen Tsunami verursachte Reaktorkatastrophe im japanischen **Fukushima** löste den Ausstieg Deutschlands aus der Nutzung von Kernenergie aus. Deutsche Atomkraftwerke wurden Stresstests unterzogen und ältere Reaktoren abgeschaltet (BMUV, 2023). Damit rückten erneuerbare Energien, wie Wind und Solar stärker in den Fokus. Bereits einen Monat nach der Atomkatastrophe verkündete die damalige Kanzlerin Angela Merkel die Energiewende, die Deutschland schnellstmöglich ins Zeitalter der regenerativen Energien bringen sollte (Landeszentrale für politische Bildung BW, 2011). Ende 2011 gerieten die Katastrophe und deren Auswirkungen aus dem medialen Fokus.

Der Überfall Russlands auf die Ukraine im Februar 2022 begründet den **Ukrianeckrieg**, welcher zu Sanktionen seitens der EU gegenüber Russland führt. Russland antwortet darauf mit Lieferverknappungen für Erdgas, was weltweit Verwerfungen an den Energiemarkten auslöst (Hosp, Höltchi, Keusch, & Schürpf, 2023). Zudem sorgt eine Hitzewelle in Europa dafür, dass Atom- und Wasserkraftwerke ausfallen. Die Versorgungs-Lücken werden teilweise durch die Verstromung von Gas gefüllt (Bundesnetzagentur, 2023). Gas- und Strompreise steigen innerhalb weniger Monate um ein Vielfaches (Sackmann, 2022).

Bis Ende 2022 beschließt die Bundesregierung drei Entlastungspakete, die eine Strom- und Gaspreisbremse und diverse Härtefallregelungen enthalten, und

entschärft damit die Lage der Verbraucher und Unternehmen (Grimm, Groß, Marxsen, & Schwarz, Wirtschaftsdienst, 2023). Daher wird der Einflussfaktor *Ukrainekrieg* von Februar bis Dezember 2022 modelliert.

Ereignisse, wie der Ukrainekrieg, der Solardeckel oder Fukushima werden je nach Relevanz des Einflusses als 0 (kein Einfluss) oder 1 (Einfluss) kodiert. Alle Zeitreihen-Daten werden monatlich für den Zeitraum 01/2000 bis 12/2023 erfasst.

## 2.3. Räumliche Zuordnung der Daten

Die erhobenen sozioökonomischen Daten werden anhand der Region mit den Marktstammdaten verknüpft. Während den PV-Anlagen im Markstammdatenregister *Postleitzahlen* und der 8-stellige amtliche Gemeindeschlüssel AGS-8 zugeordnet sind, werden für statistische Daten meist das Bundesland, der 12-stellige Amtliche Regionschlüssel (ARS), der 8-stellige Amtliche Gemeindeschlüssel (AGS) oder die verkürzte 5-stellige Form des AGS verwendet (Statistisches Bundesamt, DeStatis, 2020).

Der Markstammdaten-Tabelle wird daher eine Spalte AGS-5 für den 5-stelligen AGS hinzugefügt, um auch auf dieser Ebene Daten verknüpfen zu können. Zudem wird jede neu erstellte Tabelle testweise gejoint, um auf fehlende Werte zu prüfen und diese gegebenenfalls zu behandeln.

Im Markstammdatenregister sind 8182 individuelle Postleitzahlen enthalten, während das Postleitzahlen-Register nur 8170 gültige Postleitzahlen listet (Schwochow, 2023). Es werden 198 Zeilen standardisiert, indem ungültige PLZ händisch durch die korrekten Werte ersetzt werden.

Die unterschiedlich hohe Granularität der Daten auf räumlicher Ebene beeinflusst Aussagekraft und Generalisierbarkeit der Ergebnisse. Während AGS-8 11.107 Ausprägungen aufweist, gibt es nur 8160 *Postleitzahlen* und 403 individuelle

Werte für AGS-05. Zudem weisen *PLZ* und *AGS-08* räumliche Überschneidungen auf, sodass eine wechselseitige Zuordnung nicht eindeutig und somit fehlerbehaftet ist.

## 2.4. Sicherung der bereinigten Daten als Data-Warehouse

Zur Datensicherung wird eine SQLite-Datenbank genutzt, da sie keinen separaten Serverprozess, Konfiguration oder Installation erfordert sowie eine effiziente Verwaltung und Portabilität ermöglicht (SQLite, 2023).

Die Datenbank *dwh* ist im *output*-Ordner gespeichert und beinhaltet alle erhobenen und bereinigten Daten in 26 Tabellen.

Detaillierte Informationen zum Aufbau des Data-Warehouse (DWH), den enthaltenen Daten und deren Relationen sowie zur Datenbereinigung sind in der *Readme-Datei* zu finden.

## 2.5. Datenintegration

Das Data-Warehouse *dwh* bildet die Datenbasis für die folgenden Analysen und erlaubt es, die jeweils benötigten Informationen per SQL-Abfrage zu selektieren und zusammenzuführen. Zusätzlich wird eine csv-Datei *marktstd\_full.csv* im *output*-Ordner abgelegt, die neben den Markstammdaten alle erhobenen und den PV-Standorten zugeordneten sozioökonomischen, demografischen und Wetter-Daten enthält. Sie bildet die Datenbasis für die Feature-Importance-Analyse und den Entscheidungsbaum.

Erzeugte Tabellen, die zu einem späteren Zeitpunkt oder einem anderen Notebook Verwendung finden, werden effizient als Pickle-Dateien (Karvouniaris, 2024) im *main*-Ordner gespeichert.

## 2.6. Transformationen

### Daten-Vorbereitung Decision Tree und Feature Importance

Die Vorbereitung der Daten umfasst eine Unterteilung der Variablen in numerisch und kategorial sowie die Festlegung der Zielvariablen *Anzahl\_PV\_Anlagen*. In einer Machine Learning Pipeline werden numerische Variablen mittels Sklearn StandardScaler (Brownlee, Machinelearningmastery, 2020) standardisiert, während kategorialen Daten mittels OneHotEncoder (Brownlee , Machinelearningmastery, 2020) ein Zahlenwert zugeordnet wird. Die so transformierten Daten werden dem DecisionTreeClassifier zugeführt.

### Daten-Vorbereitung Zeitreihen-Analysen / SARIMAX-Modellierung

Die Zeitreihen-Daten weisen unterschiedliche Einheiten und Maßstäbe auf, was zu Fehlern bei der Gewichtung der Variablen und damit zu verzerrten Ergebnissen führen kann (Brownlee, Machinelearningmastery, 2020). Daher werden die Daten mittels Sklearn MinMaxScaler normalisiert, um die Varianz der Eingabe-Werte auf den Wertebereich zwischen 0 und 1 zu begrenzen. Auf weitere Transformations-Schritte wird zugunsten der Erklärbarkeit der Modelle verzichtet.

## 2.7. Entwicklungsumgebung

Zur Bereinigung der Daten, Erstellung des DWH und Durchführung der Analysen wird eine im Anaconda-Navigator-Environment eingerichtete Python-Umgebung verwendet. JupyterLab Version 3.6.3 dient als interaktive Entwicklungsumgebung für Python 3.11.4 (Python, 2023) zur Erstellung dokumentierter Notebooks.

Zur Reproduzierbarkeit der Environments wurden Anforderungsdateien erstellt, die eine vollständige Dokumentation der installierten Pakete und jeweiligen Version umfassen (Webster, et al., 2024) und eine einfache Wiederherstellung der

Umgebungen via pip-Befehl erlaubt. Die Requirements-Dateien sind im *ReadMe-Ordner* abrufbar.

Sofern Python-Code anderer Entwickler übernommen wird, erfolgt eine Quellenangabe im Notebook. Bei Nutzung von Chat-GPT wird im Notebook auf das im Anhang beigefügte ChatGPT-Protokoll verwiesen.

## 2.8. Analysemethoden

### Teil 1: Untersuchung der sozioökonomischen Einflussfaktoren

Um den Einfluss demografischer und sozioökonomischer Faktoren auf die Anzahl installierter PV-Anlagen pro Gemeinde näher zu bestimmen, wird der Datensatz *marktstd\_full.csv* in das Notebook *06\_Entscheidungsbaum.ipynb* eingelesen, transformiert und dem Sklearn DecisionTreeClassifier zugeführt. Entscheidungsbäume (auch: Decision Trees) sind überwachte Lernmethoden, die zur Klassifizierung und Regression verwendet werden. Ziel ist die Erstellung eines Modells, das die Zielvariable durch das Erlernen einfacher, aus den Features abgeleiteter Entscheidungsregeln, bestmöglich vorhersagt (Geeksforgeeks, 2023).

Die Ergebnisse werden als gut interpretierbarer Ereignisbaum-Plot dargestellt. Zur Evaluation des Modells wird der R<sup>2</sup> Score ermittelt. Auf Basis des Decision Trees wird eine Feature Importance Analyse angestoßen, um die einflussreichsten Features zu ermitteln und deren Impact zu quantifizieren.

### Teil 2: Zeitreihen-Analysen / SARIMAX-Modellierung

Für die in Notebook *09\_Time\_Series\_FINAL.ipynb* folgenden Zeitreihen-Analysen wird die monatlich installierte *GesamtBruttoleistung* als Zielvariable festgelegt und deren Verlauf zwischen 2000 und 2023 mittels Time Series Decomposition (Statsmodels, 2024) untersucht. Durch das Visualisieren der Zeitreihe und seiner Komponenten Trend, Saisonalität und Residuen werden Entwicklungen,

Muster und Strukturbrüche im Zeitverlauf sichtbar gemacht (Peixeiro, 2022, S. 128-129).

Um den Einfluss der im Data Frame *timeseries\_df* (288 Datenpunkte) erfassten Prädiktoren (*Anzahl\_Volleinspeisung*, *Anzahl\_Teileinspeisung*, *Einspeisevergütung*, *Strompreis*, *PV\_Kosten\_Euro\_pro\_kWp*, *Fukushima*, *Ukrainekrieg* und *Solardeckel*) auf die Zielgröße *GesamtBruttoleistung* zu evaluieren, werden vier SARIMAX-Modelle (Statsmodels (Statsmodels, 2024)) erstellt, die sich durch die inkludierten Variablen unterscheiden. Die Wahl des Modell-Typs wird durch Anforderungen hinsichtlich Interpretierbarkeit, Erfassung exogener Prädiktoren und Inklusion der saisonalen Komponente determiniert.

Das zugrundeliegende ARMA-Modell ist in erster Ordnung mit den autoregressiven Komponenten (AR) und in zweiter Ordnung mit dem gleitenden Durchschnitt (MA) verbunden. ARMA (p, q) verarbeitet die vorherigen Werte bis zu p Perioden zurück, aber auch die Residuen von bis zu q Verzögerungen (Mehandzhiyski, 2023).

Mittels Dickey-Fuller-Test (Peixeiro, 2022, S. 135) wird auf Stationarität der Features geprüft. Diese Voraussetzung ist nicht gegeben, wird jedoch für die meisten Variablen durch einmaliges Differenzieren erreicht. Für den *Strompreis* ist eine zweite Differenzierung notwendig. Um den Grad der Differenzierung zu integrieren, wird das Modell um den Parameter d=2 zu ARIMA (p, d, q) erweitert (Peixeiro, 2022, S. 115-117).

Die den Daten inhärente Saisonalität, die sich aus der Zeitreihen-Zerlegung ergibt, wird durch eine Erweiterung des Modells zu SARIMA (p, d, q)(P,D,Q,s) realisiert. Die neu hinzugefügten Parameter P, D und Q stellen saisonale Counterparts von p, d und q dar und s beschreibt die Frequenz, also die Anzahl der Beobachtungen pro Zyklus. Die Daten weisen ein jährlich wiederkehrendes periodisches Muster mit Spitzen im Juni und Dezember auf (siehe Anhang, Abbildung 8). Daher wird das Parameter s=12 festgelegt (Peixeiro, 2022, S. 126-127). Die Erweiterung zu SARIMAX ermöglicht die Aufnahme exogener Variablen, welche nicht Teil der Zeitreihe sind, aber die Zielvariable beeinflussen in das Modell (Peixeiro, 2022, S. 141).

Um die optimale Kombination der Parameter p, q, P und Q für jedes der Modelle zu bestimmen, wird jeweils ein Hyperparameter-Tuning auf dem Trainingsdatensatz *train\_data* (274 Datenpunkte) durchgeführt und anhand der Testdaten *test\_data* (14 Datenpunkte) validiert.

Folgende Sarimax-Modelle werden anhand ihrer Statistiken und Fehler-Metriken verglichen und bewertet:

<b>Modell 1</b>	Inkludiert alle Variablen (Anzahl_Volleinspeisung, Anzahl_Teileinspeisung, Einspeisevergütung, Strompreis, PV_Kosten_Euro_pro_kWp, Fukushima, Ukrainekrieg, Solardeckel)
<b>Modell 2</b>	Exkludiert die Variablen Anzahl_Volleinspeisung und Anzahl_Teileinspeisung.
<b>Modell 3</b>	Exkludiert die Ereignis-Variablen (Fukushima, Ukrainekrieg), sowie Anzahl_Teileinspeisung und Anzahl_Volleinspeisung
<b>Modell 4</b>	Inkludiert ausschließlich die wirtschaftlichen Faktoren Einspeisevergütung, Strompreis und PV_Kosten_Euro_pro_kWp

Tabelle 1: SARIMAX Modelle und inkludierte Prädiktoren

Zur Evaluation der Modelle wird der über Statsmodels abrufbare Model-Summary herangezogen. Das Akaike Information Criterion (AIC) schätzt die Qualität eines Modells, indem es die relative Menge der durch die Modell-Anpassung verlorenen Informationen quantifiziert. Je niedriger der AIC, desto besser das Modell. Die Auswahl nach dem AIC ermöglicht es, ein Gleichgewicht zwischen der Komplexität eines Modells und seiner Anpassungsfähigkeit an die Daten zu finden (Peixeiro, 2022, S. 97-99). Als weitere zentrale Metrik wird der Ljung-Box-Test betrachtet. Er gibt Auskunft darüber, ob es eine signifikante Autokorrelation in den Residuen gibt. Liegt der Wert bei maximal 0.05, sprechen wir von einer geringen Autokorrelation und somit guten Anpassung des Modells (Hyndman, 2014).

Die Vorhersagen der Modelle für die Testdaten (November 2022 bis Dezember 2023) werden mit den tatsächlichen Werten abgeglichen, indem sie grafisch gegenübergestellt werden.

Zur abschließenden Bewertung werden Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) und Mean Absolute Error (MAE) der Modelle gegenübergestellt. Der RMSE ist ein Maß dafür, wie weit die Residuen gestreut sind, beziehungsweise wie konzentriert die Daten um die Linie der besten Anpassung sind (Statisticshowto\_RMSE, 2024). Der MAPE misst die Genauigkeit eines Prognosesystems in Prozent (Statisticshowto\_MAPE, 2024), während der MAE die Differenz zwischen dem gemessenen Wert und dem "wahren" Wert berechnet (Statisticshowto\_MAE, 2024).

## 3. Ergebnisse

### 3.1. Sozioökonomische Einflussfaktoren

Zunächst werden die Ergebnisse der **OLS-Regressionsanalysen** betrachtet. Potenzielle Einflussfaktoren werden auf deren Einfluss auf die Zielvariable *Anzahl\_PV\_Anlagen* untersucht mit dem Ergebnis, dass keine der betrachteten Größen für sich genommen einen hohen Impact auf die Anzahl der installierten PV-Anlagen in der jeweiligen Region hat.

Die höchsten R<sup>2</sup> Werte mit jeweils über 5% und signifikanten p-Werten kleiner 0,05 erzielten folgende Eingangsvariablen:

- *Anteil fertiggestellter Wohnungen mit primär erneuerbarer Heizenergie* ( $R^2 = 0.099$ )
- *Eigenheimquote* ( $R^2 = 0.080$ )
- *"Grüne"-Wähler bei der Bundestagswahl 2021* ( $R^2 = 0.070$ )
- *Bevölkerungsdichte* ( $R^2 = 0.070$ )
- *Durchschnittliches verfügbares Einkommen* ( $R^2 = 0.062$ )

Durch Kombination unterschiedlicher Prädiktoren können deutlich bessere Resultate erzielt werden. Die *Bevölkerungsdichte* liefert zusammen mit der *Eigenheimquote* ein  $R^2$  von 0,39. Das heißt, die Variablen erklären die Variation der Zielvariable *Anzahl\_PV\_Anlagen* zu 39% (siehe Anhang, Tabelle 12). Die Eingangsvariablen *Strompreis* und *Einspeisevergütung* erzielen ein  $R^2$  von 0,34 (siehe Anhang, Tabelle 13).

Auch im **Entscheidungsbaum** (siehe Anhang, Abbildung 31) und der anschließenden **Feature Importance-Analyse** (siehe Anhang, Tabelle 14) werden der *Bevölkerungsdichte* und der *Eigentumsquote* hohe Bedeutung für die Anzahl der installierten Solaranlagen pro Gemeinde (ALG-8) beigemessen. Die höchste Dichte an Solaranlagen verzeichnen Gemeinden mit 400-800 Einwohnern pro  $km^2$  (siehe Anhang, Abbildung 9). In Ballungszentren mit über 3000 Einwohnern pro  $km^2$  hingegen werden nur wenige PV-Anlagen errichtet. Mit steigender Bevölkerungsdichte nimmt die Solaranlagen-Dichte stark ab. Eine hohe Eigentumsquote von über 80% fördert eine hohe Dichte an PV-Anlagen, während mit sinkender Quote auch die Anzahl der installierten PV-Anlagen abnimmt (siehe Anhang, Abbildung 10).

Das dem Entscheidungsbaum und der Feature Analyse zugrunde liegende Modell erreicht mit einem  $R^2$  von 0,03 kein signifikantes Ergebnis, was in Teilen auf die beschriebenen Mängel in der Datenqualität zurückzuführen ist. Darüber hinaus können Multikollinearität, nicht-lineare Beziehungen sowie Modell-inhärente Besonderheiten und Gewichtungen das Bestimmtheitsmaß  $R^2$  negativ beeinflussen.

### 3.2. Zeitreihen-Analysen und SARIMAX-Modellierung

Anhand der Regressions-Analysen konnten die *Einspeisevergütung* und der *Strompreis* als wichtige Einflussfaktoren für den Solarausbau in Deutschland ermittelt werden. In der folgenden *Abbildung 1* ist deren Entwicklung zwischen 2000

und 2023 und demgegenüber die jährlich installierte PV-Bruttoleistung grafisch dargestellt.

### Entwicklung der jährlich installierten PV-Bruttoleistung gegenüber Strompreis und Einspeisevergütung

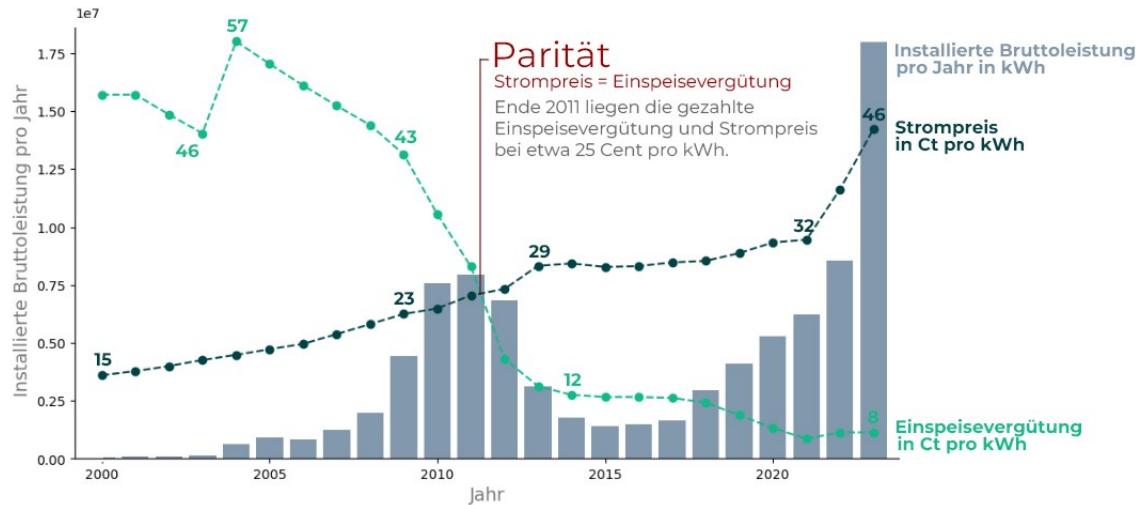


Abbildung 1: Entwicklung der jährlich installierten PV-Bruttoleistung gegenüber Strompreis und Einspeisungsvergütung

Die Einspeisevergütung fiel seit ihrem Höchst-Stand im Jahr 2004 kontinuierlich ab mit den größten Abschlägen zwischen 2009 und 2013. Demgegenüber steigen die Strompreise zwischen 2000 und 2023 um mehr als das 3-fache mit besonders starken Preiserhöhungen zwischen 2021 und 2023. Ende 2011 wird Netzparität erreicht. Das heißt, die Einspeisevergütung pro eingespeister kWh ist gleich hoch wie der Verbraucherpreis für Strom.

## Installierte PV-Anlagen in Deutschland seit 2000

Mit fallenden Einspeisevergütungen und steigenden Strompreisen nimmt die Anzahl der Anlagen mit Volleinspeisung ab 2011 deutlich ab, während mehr Anlagen mit Teileinspeisung errichtet werden.

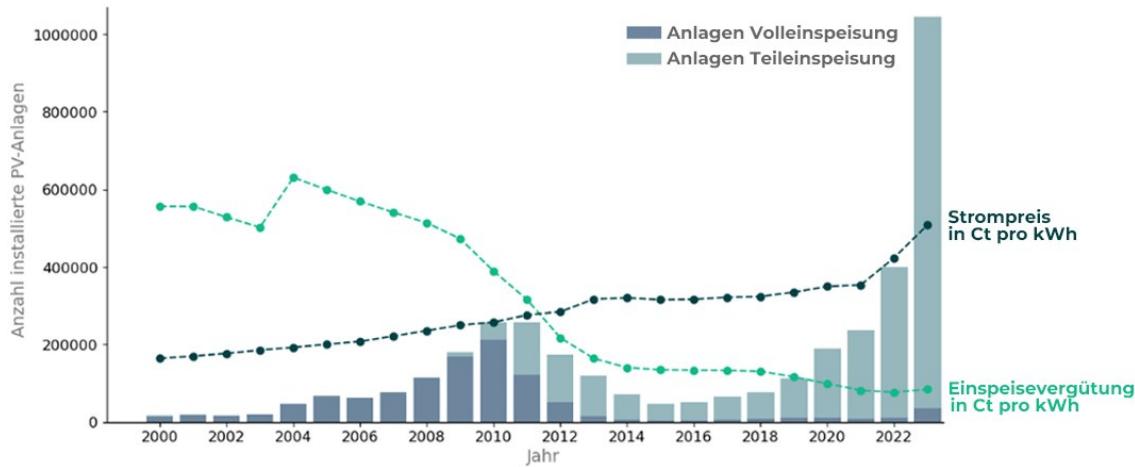


Abbildung 2: Anzahl installierter PV-Anlagen nach Voll- und Teileinspeisung seit 2000

Diese Entwicklungen lösen eine Abkehr von PV-Anlagen mit Volleinspeisung aus, wie in *Abbildung 2* zu sehen ist. Insbesondere die sinkenden Einspeisevergütungen, die für 20 Jahre ab Anmeldung der Anlage pro eingespeister kWh gezahlt werden, machen Volleinspeisung sukzessive unwirtschaftlich. Die steigenden Strompreise hingegen, welche ab 2012 mit wachsendem Abstand über den Einspeisevergütungen liegen, machen Solaranlagen mit Teileinspeisung zur Eigenversorgung rentabel. Daher sehen wir innerhalb weniger Jahre einen Paradigmenwechsel weg von der Volleinspeisung und hin zur Teileinspeisung, der bis heute anhält.

In der folgenden *Abbildung 3* werden historische Ereignisse, politische Entscheidungen und relevante Veränderungen im Zeitverlauf dargestellt und der kumulierten Bruttoleistung sowie der Anzahl installierten PV-Anlagen in Voll- und Teileinspeisung zum jeweiligen Zeitpunkt zugeordnet.

## Solarausbau Deutschland

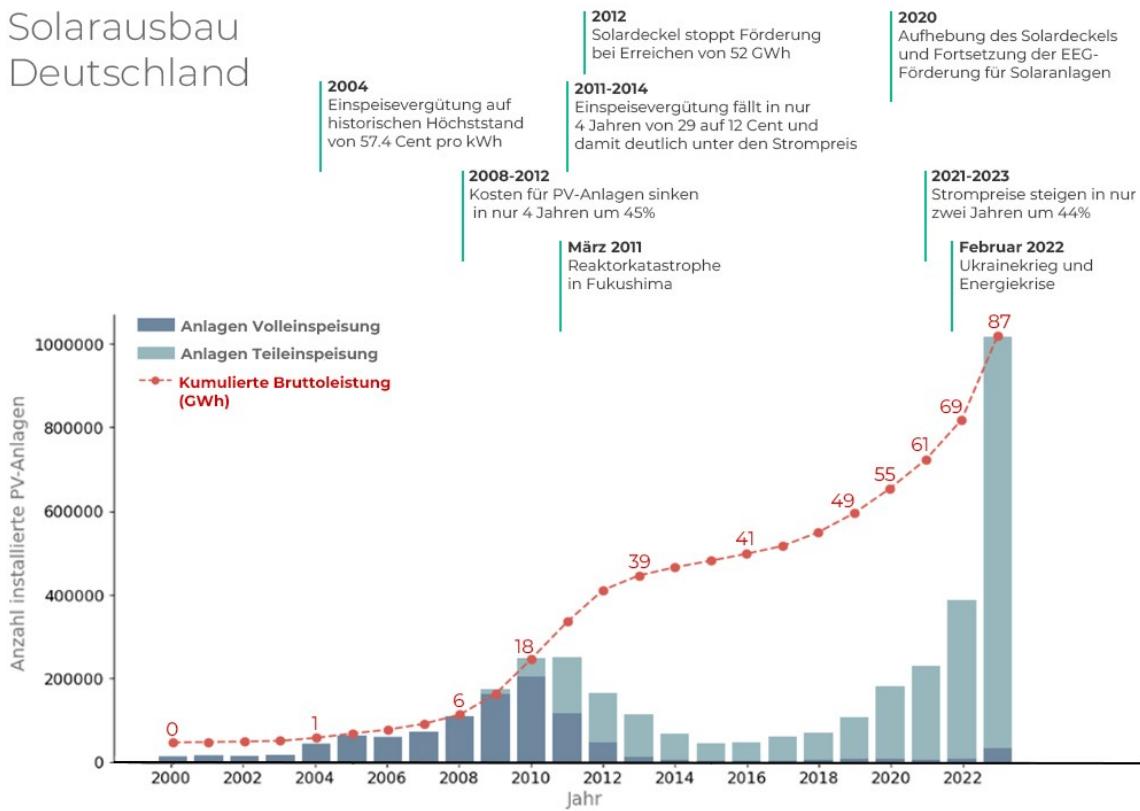


Abbildung 3: Historischer Überblick über den Solarausbau in Deutschland von 2000 - 2023

Ab 2008 sinken die Anschaffungs-Kosten für Solaranlagen deutlich (siehe Anhang, Abbildung 35). Gleichzeitig steigt die Leistung pro Solarmodul (siehe Anhang, Abbildung 16), was die Installation einer Solaranlage auch für kleine Flächen profitabel macht. Ab 2020 kommen zudem steckfertige Erzeugungsanlagen (oder Balkonanlagen) bis 600 Watt auf den Markt, die eine schnelle Inbetriebnahme auf kleinen Flächen, wie Balkonen oder Garagen erlauben und ohne Installationsaufwand direkt für den Eigenverbrauch genutzt werden können. Besonders attraktiv sind diese kleinen Einheiten für Mieter und im städtischen Bereich. In 2023 sind fast 30% der neu installierten Solaranlagen steckfertige Erzeugungsanlagen (siehe Anhang, Abbildung 18).

Die günstigen Solarkosten und sinkenden, aber noch immer über dem Strompreise liegenden Einspeisevergütungen lösen ab 2008 einen Run auf Solaranlagen aus, der bis 2011 anhält. Eine in diesem Zeitraum errichtete Solaranlage erwirtschaftet 30 bis 50% Eigenkapitalrendite und ist damit eine sichere und

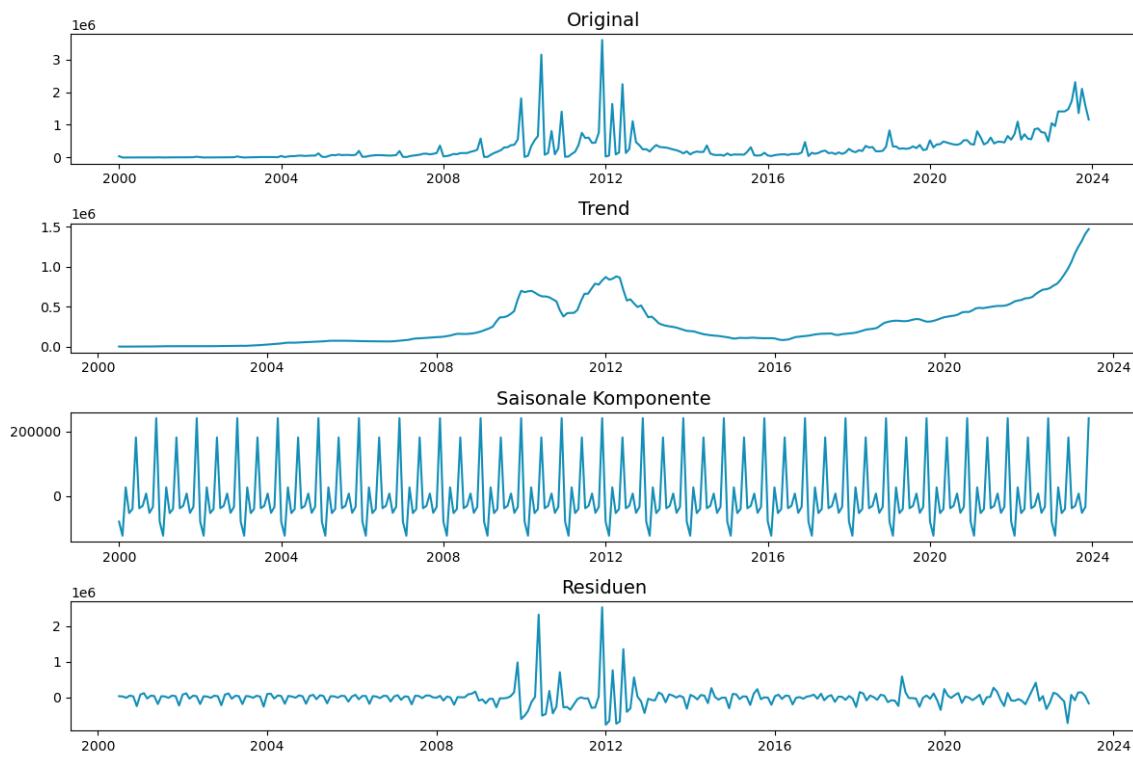
rentable Investition. Finanziert wird die Förderung erneuerbarer Energien aus der EEG-Umlage als Bestandteil des Strompreises, wodurch eine Schieflage in der Lastenverteilung droht, der die Politik begegnen muss (Solarbranche, 2015).

Gleichzeitig löst die Reaktorkatastrophe von Fukushima im März 2011 eine politische Abkehr von Atomenergie und Fokussierung auf den Ausbau erneuerbarer Energien aus. Ausmaß und Folgen der atomaren Verstrahlung für Mensch und Natur bestimmen über Monate die mediale Berichterstattung (Landeszentrale für politische Bildung BW, 2011).

Die angesprochene Überförderung von Solaranlagen wird durch Senkungen der Einspeisevergütungen nicht schnell genug beendet, so dass ab 2012 weitere Maßnahmen in Kraft treten, die den PV-Zubau wirksam begrenzen. In 06/2012 tritt der Solardeckel in Kraft. Er beinhaltete einen Stopp der Solarförderung inklusive Einspeisevergütung bei Erreichen einer installierten PV-Leistung in Höhe von 52 GWh (Leuschner, energie-chronik\_4, 2019). Gleichzeitig wird der jährliche Zubau-Korridor immer weiter verengt und die Degression der Einspeisevergütung monatlich den Zubau-Raten angepasst. Ein Überschreiten der eng gesteckten Ausbauziele führt zu einer stärkeren Absenkung der EEG-Förderung. Diese Einschnitte führen zum Einbruch des Solar-Ausbaus (Leuschner, Energie-chronik\_1, 2018). Zwischen 2013 und 2018 werden lediglich 10 GWh Solarleistung neu installiert.

Ab 2019 steigt die jährlich installierte Bruttolleistung wieder deutlich an und erreicht in 2023 mit fast 18 GWh einen neuen Höchstwert. Der Ausbau wird durch starke Strompreis-Erhöhungen getrieben, die mit Ausbruch des Ukraine-Kriegs im Februar 2022 und der daraus folgenden Energiekrise weiter zunehmen.

Im nächsten Schritt wird im Rahmen einer **Zeitreihen-Zerlegung** betrachtet, wie sich die monatlich installierte Gesamt-Bruttolleistung im Zeitraum von 2000 bis 2023 entwickelt hat und welche Trends und saisonalen Muster der Zeitreihe zu grunde liegen.



**Abbildung 4:** Zeitreihen-Zerlegung der monatlich installierten Gesamt-Bruttolleistung in Trend, Saisonalität und Residuen.

Der erste Plot in *Abbildung 4* zeigt die Entwicklung der Zielvariablen *GesamtBruttolleistung* über die Zeit. Im zweiten Plot ist der Trend dargestellt. Bis 2010 ist ein positiver Trend zu beobachten. Zwischen 2011 und 2013 ist eine starke Entwicklung zu sehen, die ab 2013 an Dynamik verliert und zu sinkenden Werten führt. Erst ab 2016 ist wieder ein positiver Trend erkennbar, der zwischen 2022 und 2023 nochmals deutlich ansteigt.

Auffällig ist der kurze Einbruch des positiven Trends in 2011. In diese Zeit fallen besonders hohe Abschläge bei der Einspeisevergütung, das Erreichen der Netzarbeit und die damit verbundene Umorientierung von Voll- zu Teileinspeisung sowie das Reaktorunglück von Fukushima. Im April 2012 werden weitere Senkungen der Vergütungssätze von rund 41 Prozent in 2012 verabschiedet. Kurzfristig führt das aufgrund der Übergangsfristen zu einem Anstieg der Solar-Installationen. Wer seinen Netzanschluss vor dem 24.02.2012 beantragt und bis zum 30.06.2012 ans Netz geht, profitiert (Leuschner, energie-chronik\_5, 2012).

Das sich jährlich wiederholende Muster im dritten Plot zeigt die saisonale Komponente mit Peaks in der Jahresmitte und gegen Ende des Jahres. Im letzten Diagramm sind die Residuen dargestellt. Sie zeigen deutliche Abweichungen in den Daten, die nicht durch Trend und saisonale Effekte erklärt werden können.

Um den Einfluss der Prädiktoren *Anzahl\_Volleinspeisung*, *Anzahl\_Teileinspeisung*, *Einspeiseverguetung*, *Strompreis*, *PV\_Kosten\_Euro\_pro\_kWp*, *Fukushima*, *Ukrainekrieg* und *Solardeckel* näher zu evaluieren, wird die Performance der **Sarimax-Modelle 1 bis 4** (siehe Tabelle 1) zunächst anhand Plots der Vorhersagen auf die Testdaten, welche die letzten 14 Monate der Zeitreihe umfassen, bewertet.

Vergleich der vorhergesagten mit den tatsächlichen Werten für den Testzeitraum 11/2022 bis 12/2023

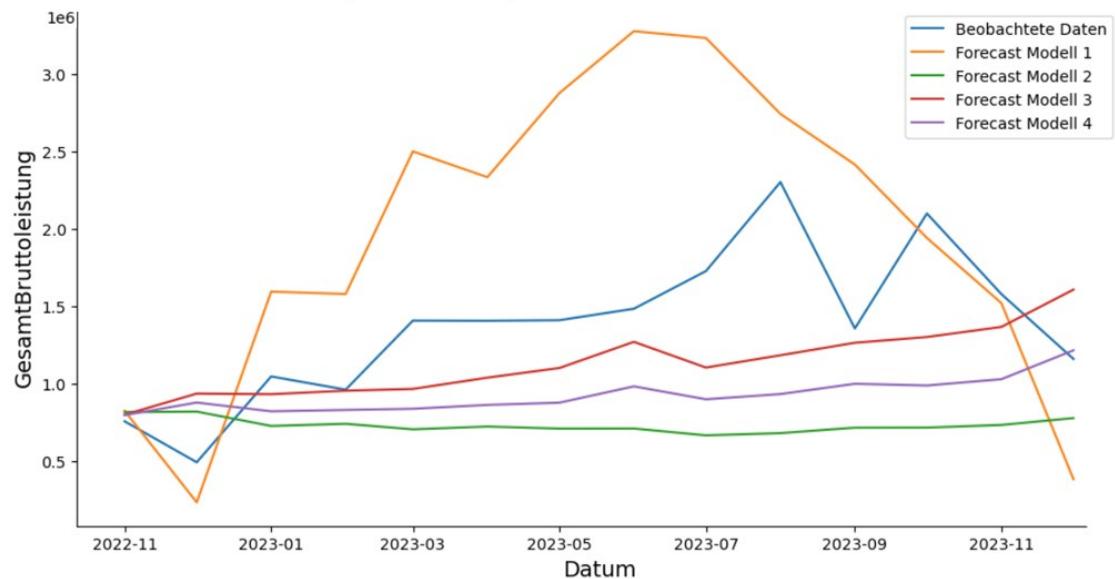


Abbildung 5: Vergleich der Vorhersage-Genauigkeit der vier Sarimax-Modelle

Wie im **Vorhersageplot** in *Abbildung 5* zu sehen, liegt das als rote Linie dargestellte Modell 3, welches die Variablen *Einspeiseverguetung*, *Strompreis*, *Solardeckel* und *PV\_Kosten\_Euro\_pro\_kWp* enthält am nächsten an den tatsächlichen

gemessenen Werten. Das in Gelb visualisierte Modell 1, welches zusätzlich die Anzahl installierter Anlagen in Voll- und Teileinspeisung erfasst, zeigt eine deutlich bessere Anpassung an steigende und fallende Tendenzen in der Originalkurve. Die Vorhersagen weichen jedoch wertmäßig deutlich ab. Das könnte darauf zurückzuführen sein, dass der jüngste Trend hin zu Balkon-Kraftwerken mit geringer Leistungserbringung nicht ausreichend berücksichtigt wird.

Die ermittelten **MAE** und **MAPE** bestätigen die Analyse-Ergebnisse des Prognoseplots (siehe Anhang, Tabelle 15). Mit einem MAPE von 27,3% schneidet Modell 3 deutlich besser ab als Modell 1 mit einem MAPE von 57,6%. Der RMSE liegt bei 632.189 für Modell 3 und 940.217 für Modell 1. Die Modelle 2 und 4 schneiden etwas schlechter ab.

Der **AIC** von Model 1 weist mit einem Wert von 7460 die beste Anpassung des Models an die Daten auf. Zudem liegt der **Ljung-Box**-Wert mit 0,01 unter dem Signifikanzlevel. Alle anderen Modelle zeigen mit Ljung Box-Werten zwischen 28,84 und 36,48 eine signifikante Autokorrelation in den Residuen.

## 4. Diskussion und Handlungsempfehlung

Die Zeitreihen-Analyse hat aufgezeigt, dass der Solarausbau in Deutschland keine kontinuierliche Entwicklung ist und durch zahlreiche, sich im Zeitverlauf ändernde Einflussfaktoren determiniert wird. Auf Basis der vorgelagerten Regressionsanalyse, die den Faktoren **Strompreis** und **Einspeisevergütung** zusammen einen hohen Einfluss auf den Solarausbau zuschreibt, wurden diese Variablen in der folgenden Zeitreihenanalyse detailliert untersucht. Zwischen 2008 und 2011 lösten sinkende Preise für Solaranlagen und hohe Einspeisevergütungen, die weit über dem Strompreis lagen, einen Solarboom aus.

Die ab Inbetriebnahme für über 20 Jahre festgeschriebene Einspeisevergütung bedeutete planbare, hohe Renditen für Solar-Investitionen. Gleichzeitig führte diese Übersubventionierung zu einer gesellschaftlichen Schieflage, da die Kosten über die im Strompreis enthaltene EEG-Umlage sozialisiert wurden.

Mit Erreichen der Netzparität in 2011 und weiteren Abschlägen bei der Einspeisevergütung (siehe Abbildung 1) in den Folgejahren wird das Modell Volleinspeisung wirtschaftlich unattraktiv. Technische Fortschritte bei Solarmodulen (siehe Anhang, Abbildung 16) und Speichern, die weiter sinkenden Preise für Solaranlagen sowie kontinuierlich steigende Strompreise (siehe Anhang, Abbildung 35) machen Solaranlagen in Teileinspeisung, also vorzugsweise zur Eigenversorgung genutzte Anlagen, interessant. Trotz niedriger Einspeisevergütungen erreicht der Solarausbau in 2023 einen neuen Rekord. Das zeigt, dass die Einspeisevergütung an Bedeutung verloren hat, während die steigenden Strompreise zum zentralen Entscheidungs-Kriterium für eine PV-Anlage geworden sind. Als Treiber dieser Entwicklungen können wirtschaftliche Erwägungen in Hinblick auf Inflationsabsicherung, aber auch der Wunsch nach Autarkie und einer nachhaltigen Energieversorgung angesehen werden. Motivation und Voraussetzungen für den Kauf einer Solaranlage haben sich demnach in den letzten Jahren stark verändert, während politisch lediglich nachjustiert wurde.

Mit 87 Prozent wird der Großteil des Solarausbaus in Deutschland von privaten Haushalten realisiert, während Solaranlagen in Industrie, Landwirtschaft, Gewerbe sowie öffentlichen Räumen eine untergeordnete Rolle spielen (siehe Anhang, Abbildung 17). Wie in den Regressionsanalysen ermittelt, beeinflussen **Eigenheimquote** und **Bevölkerungsdichte** die Anzahl an installierten PV-Anlagen maßgeblich. In Deutschland leben jedoch nur 46,7 Prozent der Bürger in der eigenen Immobilie. Im EU-Vergleich nimmt Deutschland damit den letzten Platz ein (Janssen, 2024). Eine Förderung des Erwerbes von Wohneigentum, gekoppelt an nachhaltige Energiekonzepte, könnte starke Anreize setzen. Gleichzeitig spricht die niedrige Solardichte in urbanen Bereichen für fehlende wirksame Konzepte für diese Räume, obwohl sie durch den hohen Anteil an geeigneten Dach- und sonstigen versiegelten Flächen prädestiniert für den Solarausbau wären.

Um die ambitionierten, im Klimaschutzgesetz 2021 festgeschriebenen Klimaziele zu erreichen, sind hohe, jährlich steigende Solarzubau-Raten vorgesehen. Dafür

ist eine zielgerichtete Förderung, die die aktuellen Entwicklungen adaptiert, essenziell.

Es konnte im Rahmen dieser Arbeit gezeigt werden, dass der Solarausbau in Deutschland zahlreichen Einflussfaktoren unterliegt, die sich in der bisherigen Förder-Praxis nicht adäquat widerspiegeln:

- Einspeisevergütungen subventionieren Solaranlagen für 20 Jahre, tragen heute jedoch nicht mehr zu einem schnelleren Solarausbau bei.
- Steigende Strompreise um über 30% in den letzten 3 Jahren (Abbildung 1) fördern den Solarausbau, belasten jedoch Wirtschaft und Privat-Haushalte finanziell, was perspektivisch den Solarausbau behindern kann.
- Die Bevölkerungsdichte und Eigenheimquote beeinflussen den Solarausbau, indem vor allem Gebiete mit hoher Eigentumsquote über 83% (siehe Anhang, Abbildung 10) und einer niedrigen Bevölkerungsdichte bis 800 Einwohner pro Quadratkilometer (siehe Anhang, Abbildung 9) hohe Ausbauquoten aufweisen.
- Sinkende Preise und eine höhere Effizienz (siehe Anhang, Abbildungen 16 und 35) haben PV-Anlagen wirtschaftlich attraktiv gemacht. Die Stromgestehungskosten liegen heute deutlich unter dem Strompreis (Kost, et al., 2021).
- Einschneidende Ereignisse, wie die Reaktor-Katastrophe von Fukushima oder der Ukraine-Krieg können zu starken Verwerfungen bei Energiepreisen, Verfügbarkeit von Rohstoffen und der Wahrnehmung von Klima-Themen in der Bevölkerung führen.
- Es wirken weitere, in den Analysen nicht ermittelte Faktoren auf den Solarausbau ein, die durch Trend oder Saisonalität nicht zu erklären sind und in dieser Arbeit nicht erfasst werden konnten.

Insofern wird eine weiterführende Erforschung potenzieller Einflussfaktoren auf Basis aktueller Daten angeregt. Methodisch können dafür die in der vorliegenden Arbeit vorgestellte Feature Importance Analyse und Decision Tree herangezogen werden.

Ziel der Ermittlung von Einflussfaktoren und deren Wechselwirkungen soll es sein, ein flexibles und effizientes Solar-Fördermodell zu etablieren, das starke Anreize für klimafreundliche Energien setzt, sich kontinuierlich den aktuellen Gegebenheiten anpasst und möglichst viele Interessensgruppen inkludiert.

## 5. Literaturverzeichnis

- Ahmed, R. (22. Mai 2020). *Medium.com*. Abgerufen am 15. Juli 2023 von <https://medium.com/analytics-vidhya/finding-nearest-pair-of-latitude-and-longitude-match-using-python-ce50d62af546>
- Ammon, D., Bruns, T., & Semerow, N. (01. November 2019). *solarwirtschaft.de*. Abgerufen am 16. November 2023 von <https://mailchi.mp/cc4cd867d367/2fkhzq8d2u>
- Ariadne-Konsortium. (11. Oktober 2021). *ariadneprojekt.de*. (K.-P. Ariadne, Hrsg.) Abgerufen am 16. November 2023 von [https://ariadneprojekt.de/media/2022/02/Ariadne\\_Szenarienreport\\_Oktob er2021\\_corr0222.pdf](https://ariadneprojekt.de/media/2022/02/Ariadne_Szenarienreport_Oktob er2021_corr0222.pdf)
- BDEW. (31. Oktober 2023). *Bdew.de*. Abgerufen am 16. November 2023 von <https://www.bdew.de/presse/pressemappen/strompreis/>
- BMUV. (08. September 2023). Abgerufen am 20. November 2023 von <https://www.bmuv.de/themen/nukleare-sicherheit/fukushima-folgemassnahmen/ueberblick>
- BMWK. (13. Januar 2022). *Bmwk.de*. Abgerufen am 16. November 2023 von [https://www.bmwk.de/Redaktion/DE/Binaer/Energiedaten/energiedaten-gesamt-xls-2022.pdf?\\_\\_blob=publicationFile&v=3](https://www.bmwk.de/Redaktion/DE/Binaer/Energiedaten/energiedaten-gesamt-xls-2022.pdf?__blob=publicationFile&v=3)
- Brownlee , J. (17. August 2020). *Machinelearningmastery*. Abgerufen am 26. November 2023 von <https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/>
- Brownlee, J. (28. August 2020). *Machinelearningmastery*. Abgerufen am 26. November 2023 von <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/>
- Bundesamt für Kartographie und Geodäsie. (31. Dezember 2020). *Mis.bkg.bund.de*. Abgerufen am 14. September 2023 von

<https://mis.bkg.bund.de/trefferanzeige?docuuuid=0FF7C595-E012-4ED3-AD7E-F9D9F031E147>

Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit. (07. Juli 2021). *bmuv.de*. Abgerufen am 05. Dezember 2023 von [https://www.bmuv.de/fileadmin/Daten\\_BMU/Download\\_PDF/Klimaschutz/ksg\\_aendg\\_2021\\_3\\_bf.pdf](https://www.bmuv.de/fileadmin/Daten_BMU/Download_PDF/Klimaschutz/ksg_aendg_2021_3_bf.pdf)

Bundesministerium für Wirtschaft und Klimaschutz. (05. Mai 2023). *bmwk.de*. Abgerufen am 10. Dezember 2023 von <https://www.bmwk.de/Redaktion/DE/Publikationen/Energie/photovoltaik-strategie-2023.pdf>

*Bundesnetzagentur*. (02. Januar 2023). Abgerufen am 08. März 2024 von <https://www.smard.de/page/home/topic-article/444/209624>

Bundesnetzagentur. (01. Juli 2023). *Marktstammdatenregister.de*. Abgerufen am 01. Juli 2023 von <https://www.marktstammdatenregister.de/MaStR/Datendownload>

Bundesnetzagentur. (2024). *Bundesnetzagentur.de*. Abgerufen am 09. Januar 2024 von [https://www.bundesnetzagentur.de/DE/Fachthemen/ElektrizitaetundGas/ErneuerbareEnergien/EEG\\_Foerderung/Archiv\\_VergSaetze/start.html](https://www.bundesnetzagentur.de/DE/Fachthemen/ElektrizitaetundGas/ErneuerbareEnergien/EEG_Foerderung/Archiv_VergSaetze/start.html)

Bundesverband Solarwirtschaft e.V. (14. Juni 2023). *solarwirtschaft.de*. Abgerufen am 14. November 2023 von <https://www.solarwirtschaft.de/2023/06/14/solarwirtschaft-rechnet-fuer-2023-mit-starkem-marktwachstum-und-schafft-sprung-in-den-mobilitaets-waermesektor/>

Deutscher Wetterdienst. (16. August 2023). *opendata.dwd.de*. Abgerufen am 29. September 2023 von [https://opendata.dwd.de/climate\\_environment/CDC/observations\\_germany/climate/multi\\_annual/mean\\_91-20/](https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/multi_annual/mean_91-20/)

*Deutschlandatlas*. (13. Juli 2023). Abgerufen am 14. September 2023 von [https://www.deutschlandatlas.bund.de/DE/Service/Downloads/Deutschlandatlas\\_KRS1221.html](https://www.deutschlandatlas.bund.de/DE/Service/Downloads/Deutschlandatlas_KRS1221.html)

*Deutschlandatlas*. (13. Juli 2023). Abgerufen am 14. September 2023 von [https://www.deutschlandatlas.bund.de/DE/Service/Downloads/Deutschlandatlas\\_KRS1220.html](https://www.deutschlandatlas.bund.de/DE/Service/Downloads/Deutschlandatlas_KRS1220.html)

EUPD Research. (06. April 2023). *Eupd-research.com*. (E. Research, Herausgeber) Abgerufen am 09. September 2023 von <https://www.eupd-research.com/fehlerhafte-registrierungen-von-balkon-pv-anlagen-erschweren-detaillierte-analyse-des-photovoltaik-zubaus-in-deutschland/>

Fadheli, A. (Januar 2023). *Thepythoncode*. Abgerufen am 16. Juli 2023 von <https://thepythontocode.com/article/get-geolocation-in-python>

Geeksforgeeks. (20. August 2023). Abgerufen am 08. Oktober 2023 von <https://www.geeksforgeeks.org/decision-tree/>

GovData. (14. Mai 2018). *Govdata.de*. Abgerufen am 02. September 2023 von <https://www.govdata.de/web/guest/suchen/-/details/arbeitslose-arbeitslosenquoten-gemeldete-arbeitsstellen-bundeslander-jahre>

Grimm, V., Groß, C., Marxsen, T., & Schwarz, M. (November 2023). *Wirtschaftsdienst*. Abgerufen am 2. Januar 2024 von <https://www.wirtschaftsdienst.eu/inhalt/jahr/2023/heft/11/beitrag/energiekrise-belastet-haushalte.html>

Grimm, V., Groß, C., Marxsen, T., & Schwarz, M. (November 2023). *Wirtschaftsdienst.eu*. Abgerufen am 28. Januar 2024 von <https://www.wirtschaftsdienst.eu/inhalt/jahr/2023/heft/11/beitrag/energiekrise-belastet-haushalte.html>

Hosp, G., Höltchi, R., Keusch, N., & Schürpf, T. (21. September 2023). *NZZ*. Abgerufen am 05. Januar 2024 von <https://www.nzz.ch/wirtschaft/rohstoffe-als-waffen-die-neusten-entwicklungen-ld.1681268>

- Hyndman, R. (24. Januar 2014). *robjhyndman.com*. Abgerufen am 15. März 2024 von <https://robjhyndman.com/hyndtsight/ljung-box-test/>
- Janssen, F. (27. Februar 2024). *IWD*. Abgerufen am 9. März 2024 von <https://www.iwd.de/artikel/wohneigentumsquoten-in-europa-612730/>
- Karvouniaris, C. (15. Januar 2024). *ITnext*. Abgerufen am 19. Februar 2024 von <https://itnext.io/python-pickle-from-whys-and-hows-to-dos-and-donts-70e08c3a2c5f>
- Kost, C., Shammugam, S., Fluri, V., Peper, D., Davoodi Memar, A., & Schlegl, T. (Juni 2021). *Frauenhoffer ISE*. Abgerufen am 12. November 2023 von <https://www.ise.fraunhofer.de/de/veroeffentlichungen/studien/studie-stromgestehungskosten-erneuerbare-energien.html>
- Landeszentrale für politische Bildung BW. (2011). *lpb-bw*. Abgerufen am 04. Januar 2024 von <https://www.lpb-bw.de/fukushima>
- Leuschner, U. (März 2012). *energie-chronik\_5*. Abgerufen am 8. Januar 2024 von <https://www.energie-chronik.de/120301.htm>
- Leuschner, U. (November 2018). *Energie-chronik\_1*. Abgerufen am 26. Januar 2024 von <https://www.energie-chronik.de/181102.htm#solardeckel>
- Leuschner, U. (September 2019). *energie-chronik\_4*. Abgerufen am 08. Januar 2024 von <https://www.energie-chronik.de/190903.htm>
- Leuschner, U. (Juni 2020). *energie-chronik\_3*. Abgerufen am 10. Januar 2024 von <https://www.energie-chronik.de/200607.htm>
- LpB BW. (März 2023). *Lpb-bw.de*. Abgerufen am 28. Januar 2024 von <https://www.lpb-bw.de/fukushima>
- Mehandzhiyski, V. (21. April 2023). *365datascience*. Abgerufen am 14. März 2024 von <https://365datascience.com/tutorials/time-series-analysis-tutorials/arma-model/>
- Our World in Data. (12. Dezember 2023). *Ourworldindata.org*. Abgerufen am 01. Februar 2024 von <https://ourworldindata.org/grapher/solar-pv-prices>

- Peixeiro, M. (04. Oktober 2022). *Time Series Forecasting in Python*. New York: Manning Publications Co. LLC.
- Python*. (6. Juni 2023). Abgerufen am 12. August 2023 von <https://www.python.org/downloads/release/python-3114/>
- Quaschning, P., Orth, N., Weniger, D., Bergner, J., Siegel, B., & Zoll, M. (24. November 2021). *solar.htw-berlin.de*. Abgerufen am 29. Dezember 2023 von <https://solar.htw-berlin.de/wp-content/uploads/HTW-Studie-Solarstromausbau-fuer-den-Klimaschutz.pdf>
- REWAG. (2022). *Rewag.de*. Abgerufen am 01. Februar 2024 von <https://www.rewag.de/aktuelle-marktlage/eeg-umlage#group3578>
- Sackmann, C. (06. August 2022). *Focus*. Abgerufen am 05. Januar 2024 von [https://www.focus.de/finanzen/news/frankreich-muss-reaktoren-drosseln-atomkraft-in-der-krise-wie-die-hitzewelle-europas-strompreise-weiter-antreibt\\_id\\_128688904.html](https://www.focus.de/finanzen/news/frankreich-muss-reaktoren-drosseln-atomkraft-in-der-krise-wie-die-hitzewelle-europas-strompreise-weiter-antreibt_id_128688904.html)
- Schwochow, M. (15. Juli 2023). *suche-postleitzahl.org*. Abgerufen am 16. Juli 2023 von <https://www.suche-postleitzahl.org/plz-karte-erstellen>
- Solarbranche*. (15. Oktober 2015). Abgerufen am 16. Februar 2024 von <https://www.solarbranche.de/news/nachrichten/artikel-29903-photovoltaik-anlagen-2009-bis-2011-waren-die-fetten-jahre>
- Solarenergie Förderverein. (10. Januar 2017). *Sfv.de/lokal-mails/sj/verguetu*. Abgerufen am 09. Januar 2024 von <https://www.sfv.de/lokal-mails/sj/verguetu>
- SQLite. (10. Oktober 2023). *SQLite.org*. Abgerufen am 22. Januar 2024 von <https://www.sqlite.org/about.html>
- Statisticshowto\_MAE*. (2024). Abgerufen am 15. März 2024 von <https://www.statisticshowto.com/absolute-error/>
- Statisticshowto\_MAPE*. (2024). Abgerufen am 15. März 2024 von <https://www.statisticshowto.com/mean-absolute-percentage-error-mape/>

*Statisticshowto\_RMSE.* (2024). Abgerufen am 15. März 2024 von <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/>

Statistische Ämter des Bundes und der Länder. (01. November 2022). *Statistikportal.de.* Abgerufen am 14. September 2023 von <https://www.statistikportal.de/de/veroeffentlichungen/einkommen-der-privaten-haushalte>

Statistische Ämter des Bundes und der Länder. (31. Dezember 2022). *Statistikportal.de/de.* Abgerufen am 02. September 2023 von <https://www.statistikportal.de/de/sbe/ergebnisse/einkommen-armutsgefaehrdung-und-soziale-lebensbedingungen/armutsgefaehrdung-und-3>

Statistisches Bundesamt. (25. August 2020). *DeStatis.* Abgerufen am 02. September 2023 von [https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/beschreibung-gebietseinheiten.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/beschreibung-gebietseinheiten.pdf?__blob=publicationFile)

*Statistisches Bundesamt.* (26. September 2021). Abgerufen am 04. September 2023 von <https://www.bundeswahlleiter.de/bundestagswahlen/2021/ergebnisse/weitere-ergebnisse.html>

Statistisches Bundesamt (Hrsg.). (31. Dezember 2021). *Destatis.de.* Abgerufen am 07. September 2023 von <https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/04-kreise>

*Statsmodels.* (28. Februar 2024). Abgerufen am 14. März 2024 von [https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal\\_decompose.html](https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal_decompose.html)

*Statsmodels.* (28. Februar 2024). Abgerufen am 14. März 2024 von <https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html#statsmodels.tsa.statespace.sarimax.SARIMAX>

- Vöpel, A. (02. 01 2024). *Gruenes.haus*. Abgerufen am 01. Februar 2024 von <https://gruenes.haus/photovoltaik-preisentwicklung/>
- Webster, C., Hogenson, G., Harvey, B., Patel, M., Saisang, C., Martens, J., . . . Partlow, J. (01. März 2024). *Learn.Microsoft*. Abgerufen am 12. März 2024 von <https://learn.microsoft.com/de-de/visualstudio/python/managing-required-packages-with-requirements-txt?view=vs-2022>
- Weitekamp, L. (01. Juli 2022). *Enercity.de*. Abgerufen am 01. Februar 2024 von <https://www.enercity.de/magazin/unsere-welt/was-ist-die-eeg-umlage>
- Wirth, D. (16. Januar 2024). Abgerufen am 20. Januar 2024 von [ise.fraunhofer.de:](https://www.ise.fraunhofer.de/de/veroeffentlichungen/studien/aktuelle-fakten-zur-photovoltaik-in-deutschland.html)  
<https://www.ise.fraunhofer.de/de/veroeffentlichungen/studien/aktuelle-fakten-zur-photovoltaik-in-deutschland.html>
- Zensus Datenbank*. (09. Mai 2011). Abgerufen am 22. September 2023 von <https://ergebnisse2011.zensus2022.de/datenbank/online/statistic/1000A/table/1000A-1005>
- Zensus Datenbank*. (09. Mai 2011). Abgerufen am 08. September 2023 von <https://ergebnisse2011.zensus2022.de/datenbank/online/statistic/4000W/table/4000W-1005>
- Zensus Datenbank*. (09. Mai 2011). Abgerufen am 22. September 2023 von <https://ergebnisse2011.zensus2022.de/datenbank/online/statistic/2000S/table/2000S-1010>
- Zensus Datenbank*. (09. Mai 2011). Abgerufen am 14. September 2023 von <https://ergebnisse2011.zensus2022.de/datenbank/online/statistic/1000A/table/1000A-3002>

## 6. Anhang

### 6.1. Weitere Auswertungen und Analyse-Ergebnisse

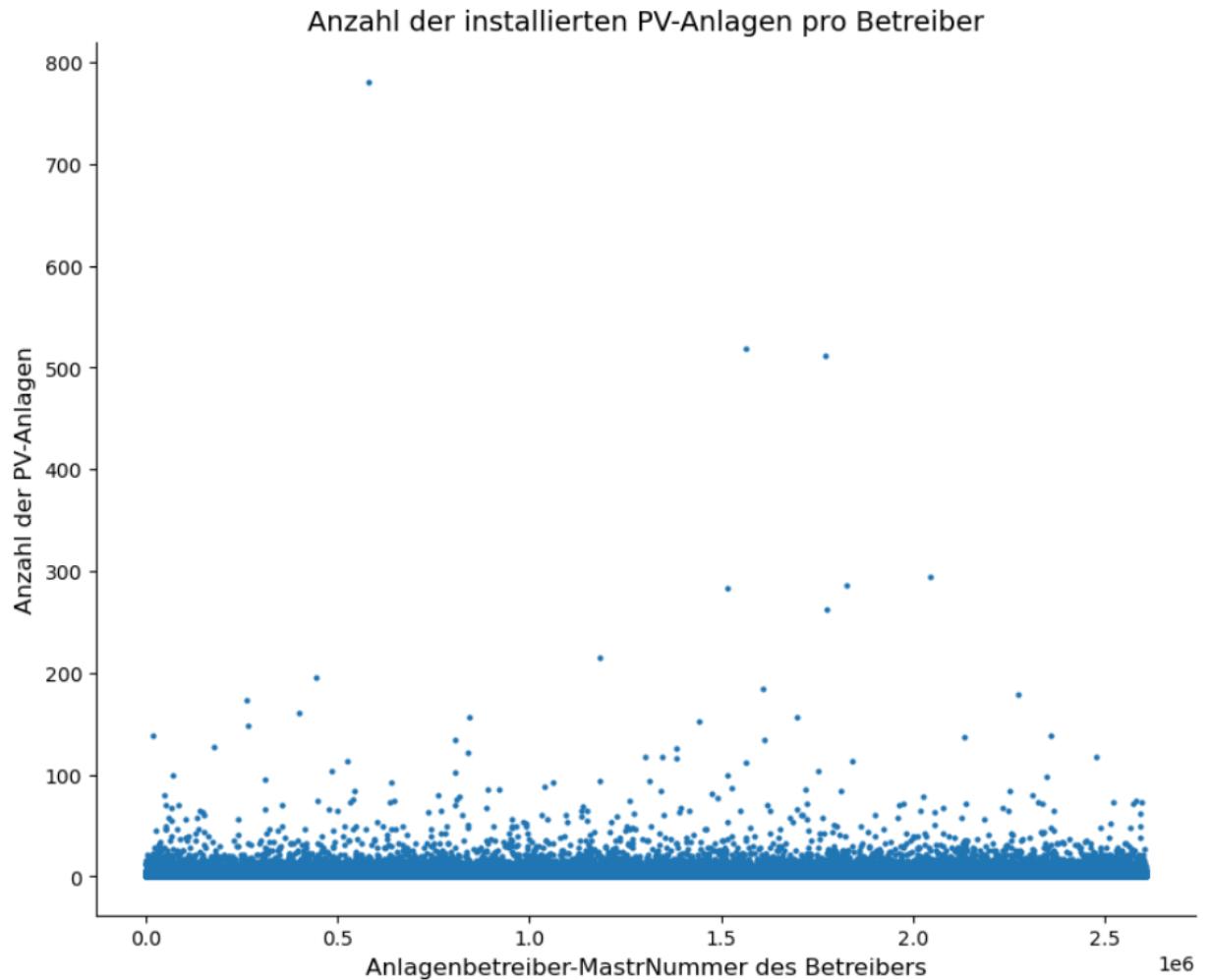


Abbildung 6: Anzahl der installierten PV-Anlagen pro Betreiber.

Im Plot sind folgende Ausreißer zu erkennen:

Betreiber-Nummer ABR922330502361 mit **781** PV-Anlagen (Brandenburg)

Betreiber-Nummer ABR960080516405 mit **519** PV-Anlagen (Sachsen)

Betreiber-Nummer ABR967988325672 mit **512** PV-Anlagen (Hessen)

Alle anderen Betreiber liegen unter 300 PV-Anlagen.

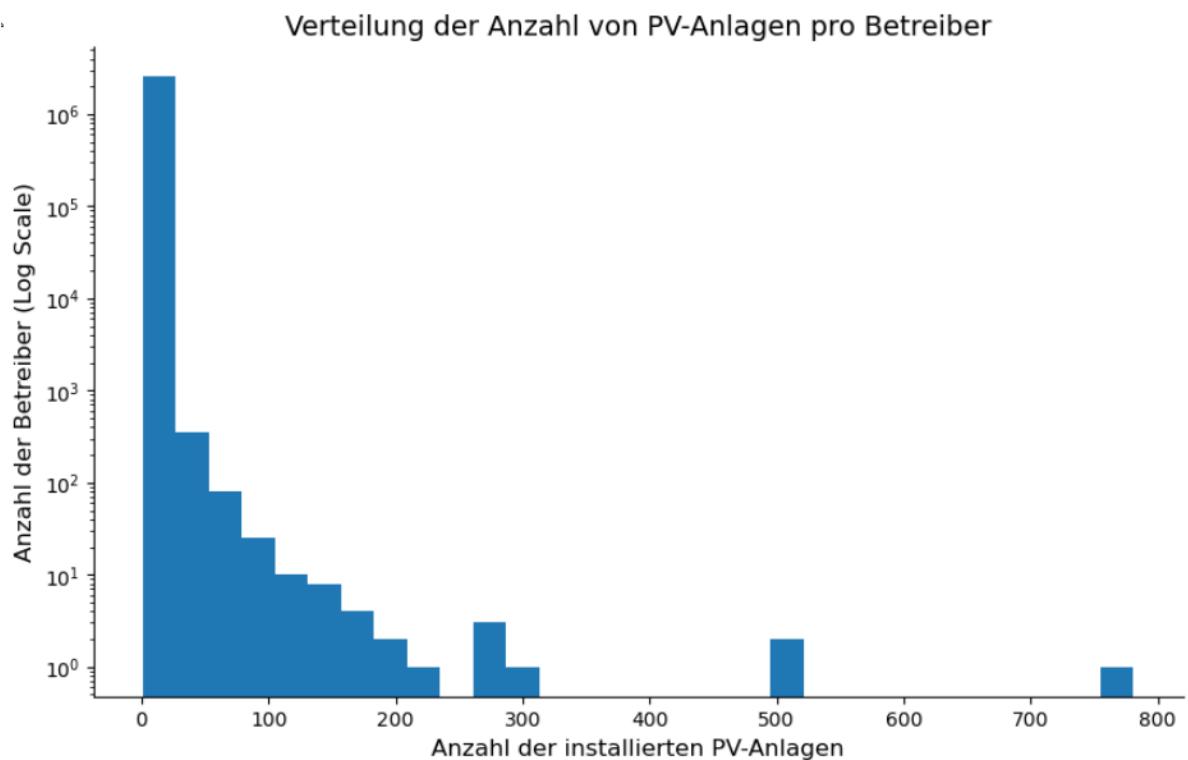
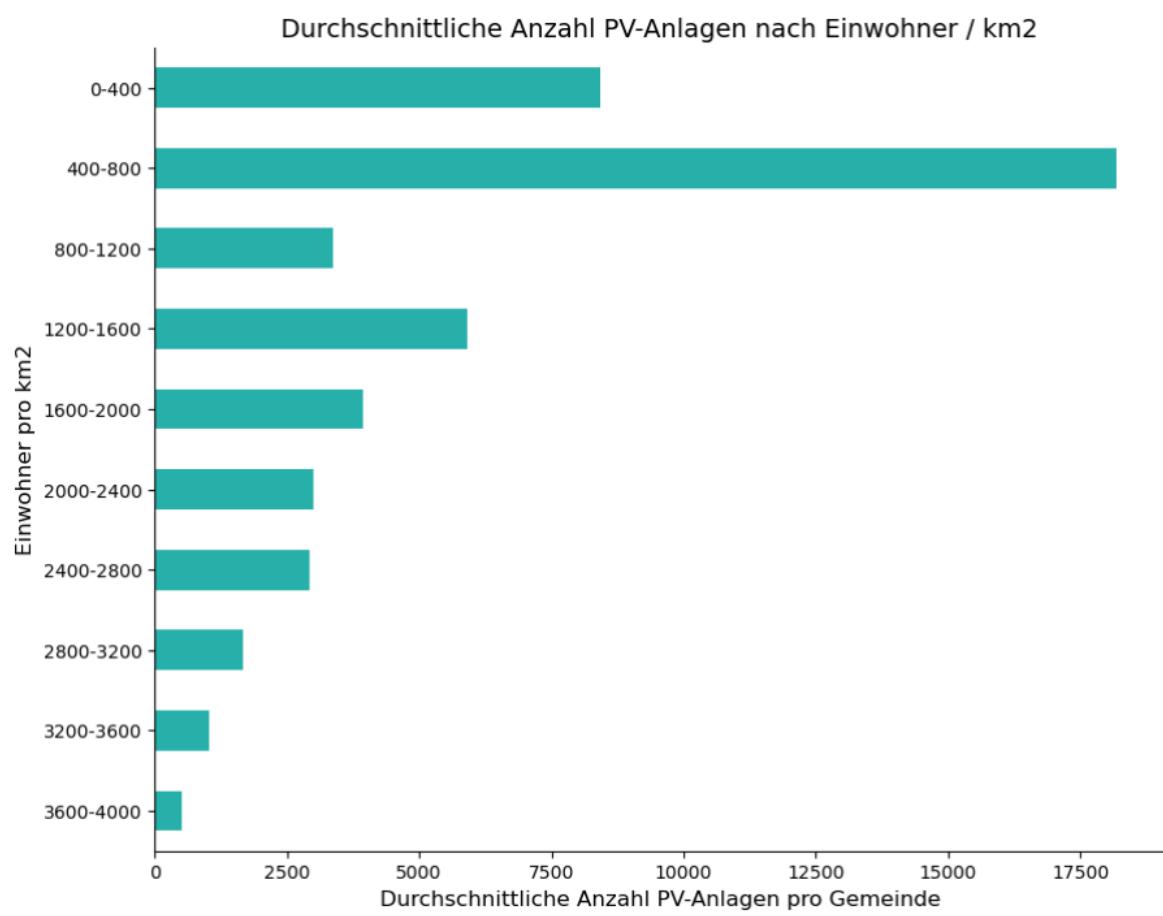


Abbildung 7: Histogramm zur Verteilung der Anzahl PV-Anlagen pro Betreiber.



**Abbildung 8:** Anzahl der installierten PV-Anlagen nach Bevölkerungsdichte.

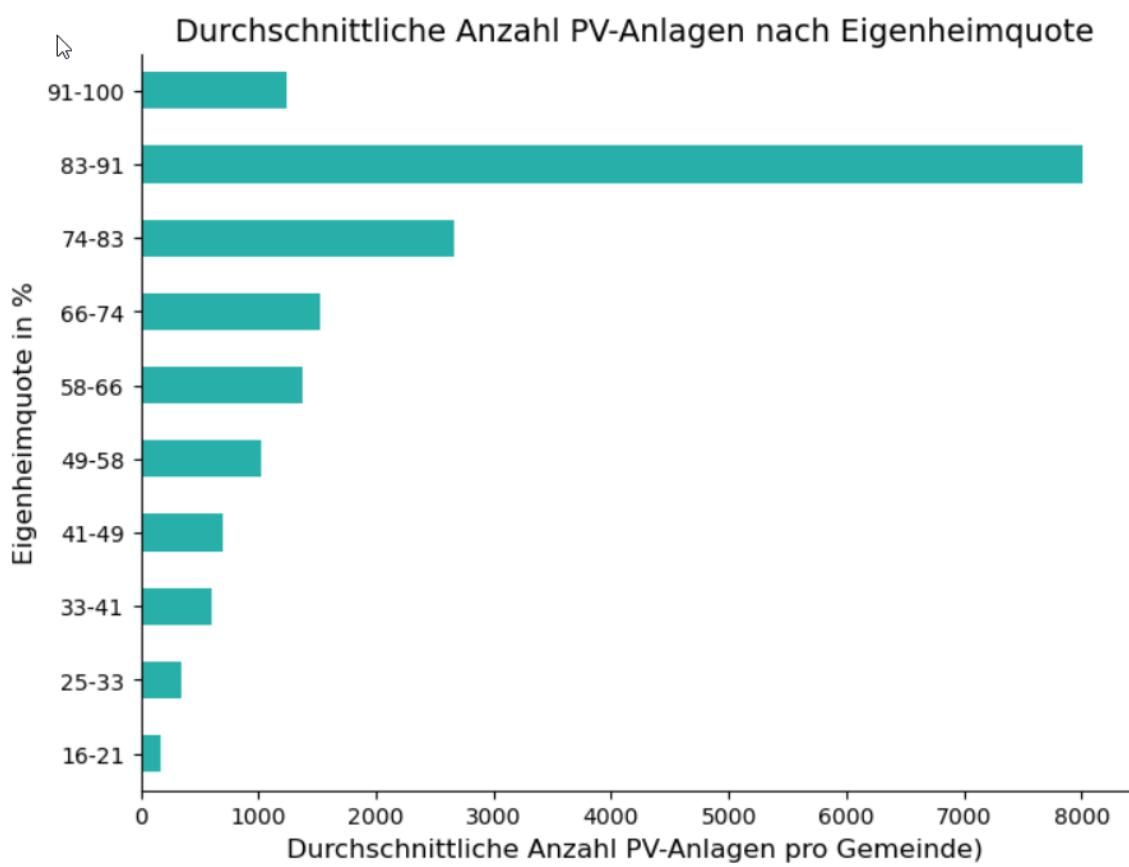


Abbildung 9: Anzahl der installierten PV-Anlagen nach Eigenheimquote

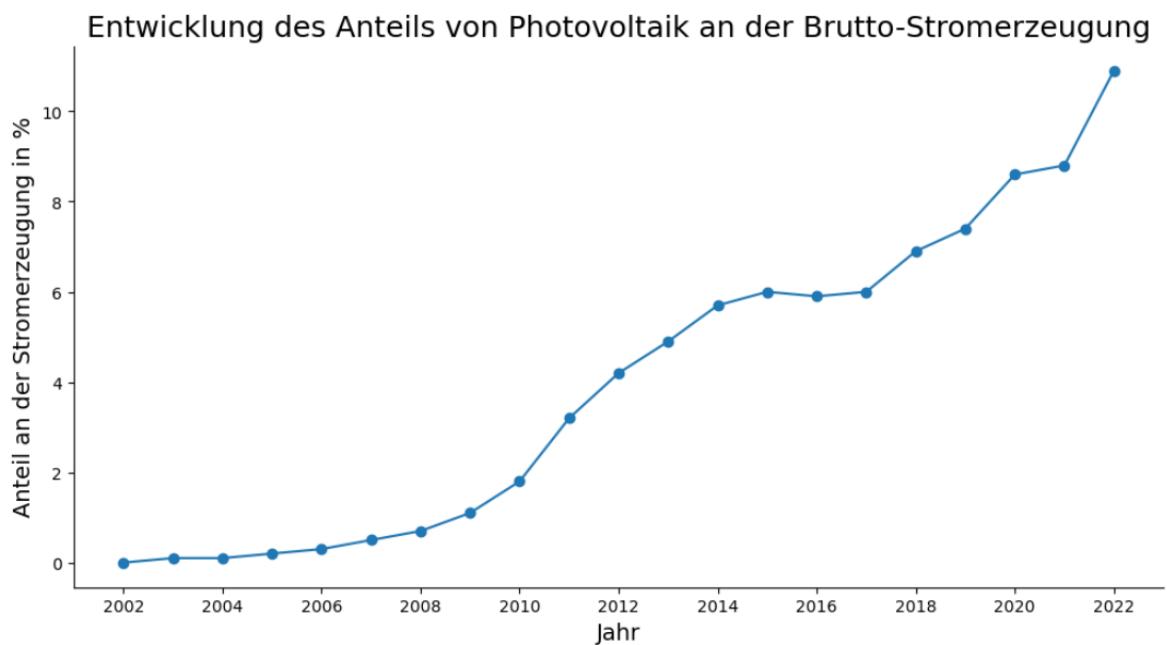


Abbildung 10: Anteil von Photovoltaik an der Brutto-Stromerzeugung in Deutschland.

- starker Anstieg des Anteils an PV an der Stromerzeugung zwischen 2009 und 2014
- stagnierender Ausbau zwischen 2014 und 2017
- seit 2018 erneut stetiger Anstieg bis auf 10,9% in 2022

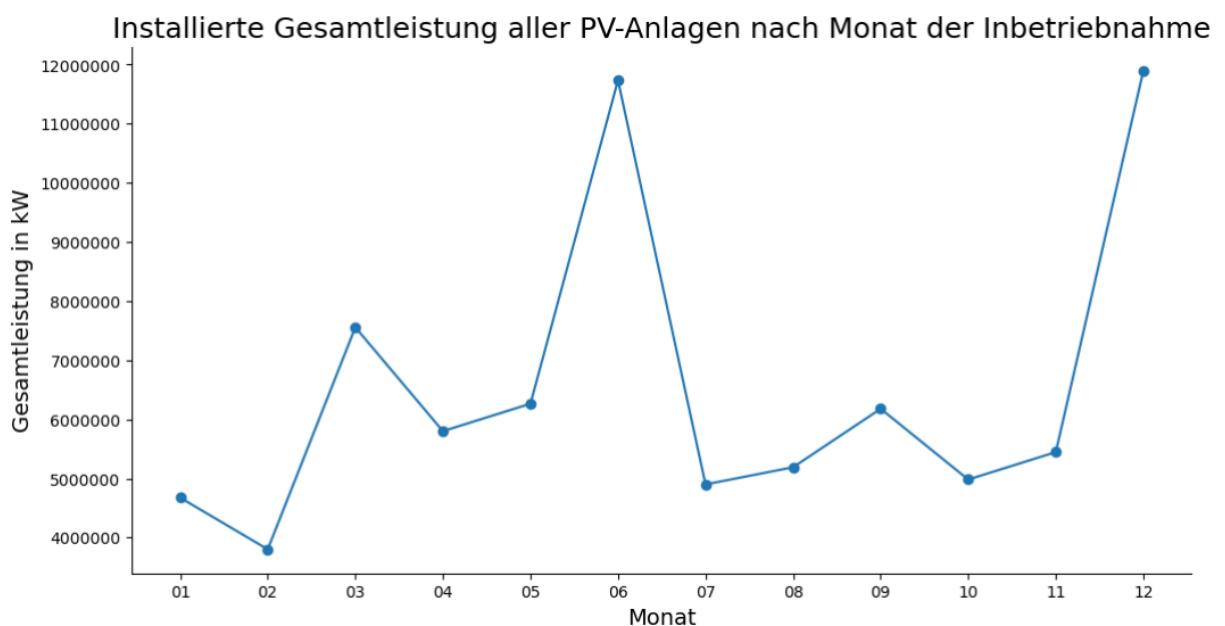


Abbildung 11: Installierte Gesamtleistung aller PV-Anlagen nach Monat der Inbetriebnahme.

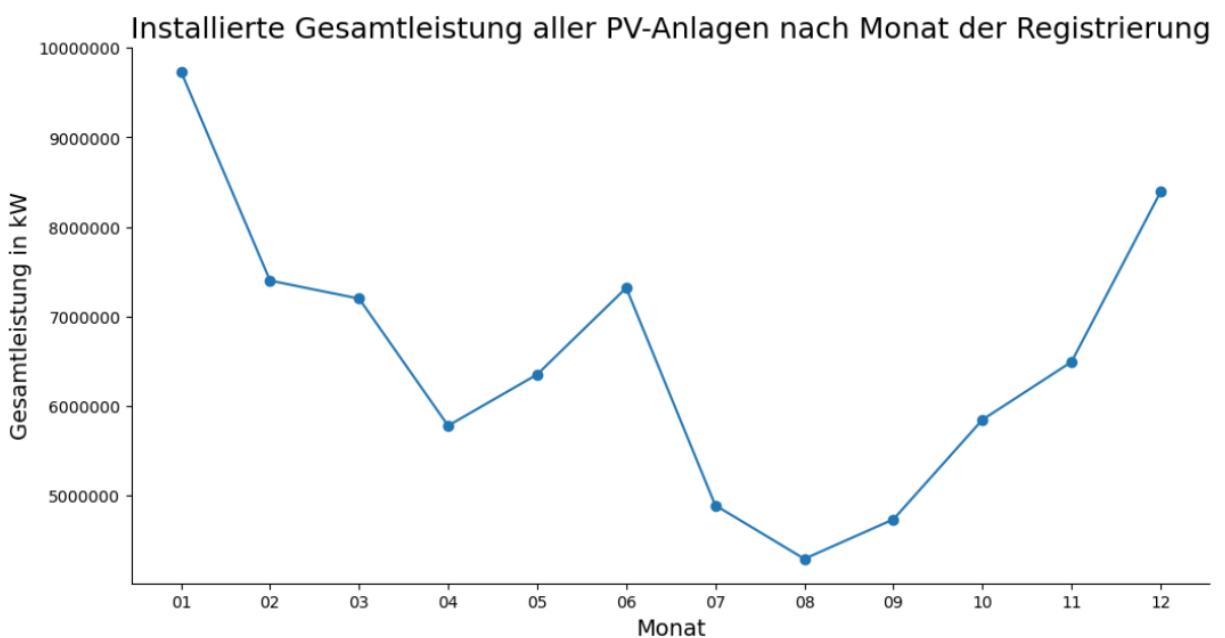


Abbildung 12: Installierte Gesamtleistung aller PV-Anlagen nach Monat der Registrierung.

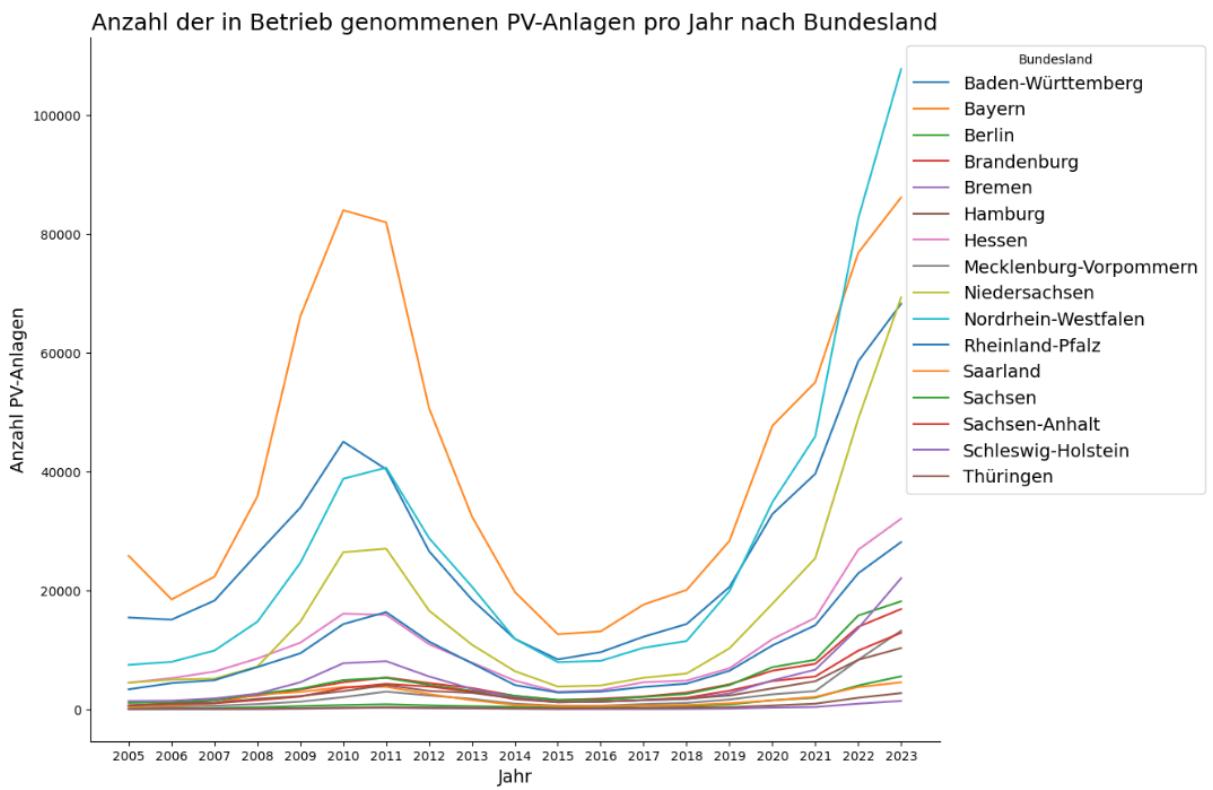


Abbildung 13: Anzahl der in Betrieb genommenen PV-Anlagen pro Jahr nach Bundesland.

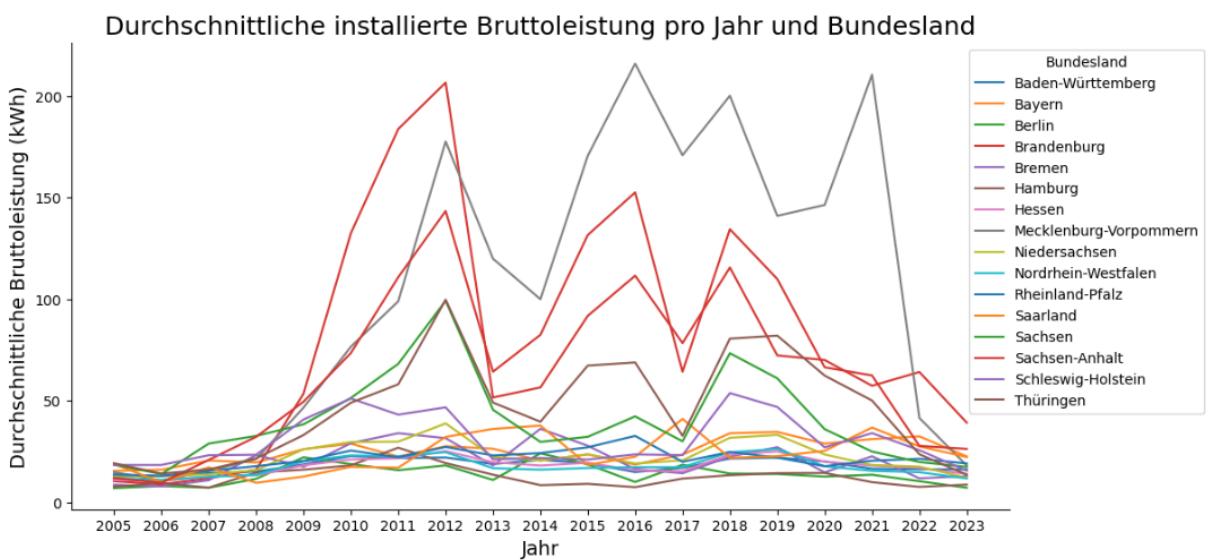


Abbildung 14: Durchschnittliche installierte Bruttoleistung pro Jahr und Bundesland.

Bundesland	Durchschnitt_Bruttoleistung	Rang
Mecklenburg-Vorpommern	105	1
Brandenburg	75	2
Sachsen-Anhalt	72	3
Thüringen	46	4
Sachsen	40	5
Schleswig-Holstein	30	6
Bayern	25	7
Saarland	23	8
Niedersachsen	22	9
Rheinland-Pfalz	21	10
Bremen	20	11
Baden-Württemberg	18	12
Hessen	18	13
Nordrhein-Westfalen	18	14
Berlin	14	15
Hamburg	12	16

**Tabelle 2:** Durchschnittliche installierte Brutto-Leistung nach Bundesland in absteigender Rangfolge.

Die weitaus größten PV-Anlagen werden in Mecklenburg-Vorpommern installiert.

Aber auch in Brandenburg und Sachsen-Anhalt ist die durchschnittliche Brutto-Leistung hoch.

Die im Durchschnitt kleinsten Anlagen befinden sich im städtischen Bereich.

<b>Postleitzahl</b>	<b>Ort</b>	<b>Bundesland</b>	<b>Prozentualer_Zuwachs</b>
12687	Berlin	Berlin	1250.000000
50677	Köln	Nordrhein-Westfalen	1100.000000
69117	Heidelberg	Baden-Württemberg	900.000000
10825	Berlin	Berlin	900.000000
12059	Berlin	Berlin	900.000000
20255	Hamburg	Hamburg	800.000000
10585	Berlin	Berlin	800.000000
13409	Berlin	Berlin	800.000000
10559	Berlin	Berlin	766.666667
12159	Berlin	Berlin	750.000000
13407	Berlin	Berlin	733.333333
13351	Berlin	Berlin	700.000000
12163	Berlin	Berlin	700.000000
99331	Geratal	Thüringen	655.555556
50674	Köln	Nordrhein-Westfalen	650.000000

**Tabelle 3:** TOP 15 Postleitzahlengebiete mit dem höchsten prozentualer Zuwachs an PV-Anlagen zwischen 2018 und 2023.

	Ort	Lage	Anzahl_Vor_2018	Anzahl_Ab_2018	Prozentualer_Zuwachs
0	Berlin	Bauliche Anlagen (Hausdach, Gebäude und Fassade)	6475	10642	164.355212
1	Berlin	Bauliche Anlagen (Sonstige)	106	185	174.528302
2	Berlin	Freifläche	8	29	362.500000
3	Berlin	Steckerfertige Erzeugungsanlage (sog. Plug-In-...)	11	3599	32718.181818

**Tabelle 4:** Prozentualer Zuwachs an PV-Anlagen in Berlin zwischen 2018 und 2023, sortiert nach Lage der installierten Anlagen.

In städtischen Bereichen, hier beispielhaft Berlin, machen steckfertige PV-Anlagen oder sogenannte Balkon-Anlagen den weitaus höchsten Zuwachs-Anteil an den Neuinstallationen aus.

<b>Postleitzahl</b>	<b>Ort</b>	<b>Bundesland</b>	<b>Prozentualer_Zuwachs</b>
42103	Wuppertal	Nordrhein-Westfalen	11.111111
82447	Spatzenhausen	Bayern	6.060606
40213	Düsseldorf	Nordrhein-Westfalen	0.000000
33333	Gütersloh	Nordrhein-Westfalen	0.000000
40212	Düsseldorf	Nordrhein-Westfalen	0.000000
82475	Schneefernerhaus	Bayern	0.000000
83256	Frauenchiemsee	Bayern	0.000000
10779	Berlin	Berlin	0.000000
25867	Oland	Schleswig-Holstein	0.000000
81248	München	Bayern	0.000000
86941	St. Ottilien	Bayern	0.000000
20249	Hamburg	Hamburg	0.000000
20354	Hamburg	Hamburg	0.000000
78266	Büsingen	Baden-Württemberg	0.000000
22301	Hamburg	Hamburg	0.000000

**Tabelle 5:** BOTTOM 15 Postleitzahlengebiete mit dem niedrigsten prozentualen Zuwachs an PV-Anlagen zwischen 2018 und 2023.

<b>Ort</b>	<b>Bundesland</b>	<b>AGS-5</b>	<b>PV_Anlagen_pro_10k_Einwohner</b>
Arnstorf	Bayern	09277	1501
Moosthenning	Bayern	09279	1443
Mallersdorf-Pfaffenberg	Bayern	09278	1407
Fürstenzell	Bayern	09275	1229
Geisenhausen	Bayern	09274	1193
Kaisheim	Bayern	09779	1184
Deggendorf	Bayern	09271	1163
Kronburg	Bayern	09778	1148
Röhrnbach	Bayern	09272	1144
Rötz	Bayern	09372	1140
Feuchtwangen	Bayern	09571	1118
Mainburg	Bayern	09273	1102
Perach	Bayern	09171	1091
Roßhaupten	Bayern	09777	1075
Glött	Bayern	09773	1058
Bad Windsheim	Bayern	09575	1032
Frauenau	Bayern	09276	1025
Pappenheim	Bayern	09577	1007
Auerbach	Bayern	09371	989
Neustadt	Bayern	09374	985

**Tabelle 6:** Top-20 Gemeinden mit der höchsten Anzahl PV-Anlagen pro 10.000 Einwohner.

Alle Top-20 Gemeinden mit den höchsten Dichten an PV-Anlagen bezogen auf die Einwohnerzahl liegen in Bayern.

<b>Ort</b>	<b>Bundesland</b>	<b>AGS-5</b>	<b>PV_Anlagen_pro_10k_Einwohner</b>
Rostock	Mecklenburg-Vorpommern	13003	97
Wiesbaden	Hessen	06414	93
Essen	Nordrhein-Westfalen	05113	93
Leipzig	Sachsen	14713	93
Stuttgart	Baden-Württemberg	08111	93
Bremerhaven	Bremen	04012	92
Duisburg	Nordrhein-Westfalen	05112	91
Potsdam	Brandenburg	12054	88
Dresden	Sachsen	14612	88
Herne	Nordrhein-Westfalen	05916	86
Gelsenkirchen	Nordrhein-Westfalen	05513	82
Kiel	Schleswig-Holstein	01002	77
Köln	Nordrhein-Westfalen	05315	77
Bremen	Bremen	04011	70
München	Bayern	09162	67
Düsseldorf	Nordrhein-Westfalen	05111	62
Berlin	Berlin	11000	55
Offenbach	Hessen	06413	53
Hamburg	Hamburg	02000	51
Frankfurt	Hessen	06412	51

**Tabelle 7:** BOTTOM-20 Gemeinden mit der niedrigsten Anzahl PV-Anlagen pro 10.000 Einwohner.

Die Bottom-20 Gemeinden sind auf verschiedene Bundesländer verteilt. Es fällt jedoch auf, dass es sich ausnahmslos um größere Städte oder Großstädte handelt.

Ort	Bundesland	AGS-5	Bruttoleistung_pro_10k_Einwohner
Pritzwalk	Brandenburg	12070	63863
Luckaitztal	Brandenburg	12066	63269
Fredersdorf-Vogelsdorf	Brandenburg	12064	57208
Mallersdorf-Pfaffenbergs	Bayern	09278	50884
Holsthum	Rheinland-Pfalz	07232	49046
Herzberg	Brandenburg	12062	46643
Moosthenning	Bayern	09279	46266
Maroldsweisach	Bayern	09674	45492
Malliß	Mecklenburg-Vorpommern	13076	42998
Spremberg	Brandenburg	12071	42744
Mainburg	Bayern	09273	40683
Arnstorf	Bayern	09277	39922
Kuhfelde	Sachsen-Anhalt	15081	39849
Pappenheim	Bayern	09577	39625
Gachenbach	Bayern	09185	39048
Bad Windsheim	Bayern	09575	38276
Feuchtwangen	Bayern	09571	38235
Allstedt	Sachsen-Anhalt	15087	38106
Prenzlau	Brandenburg	12073	37648
Deggendorf	Bayern	09271	36755

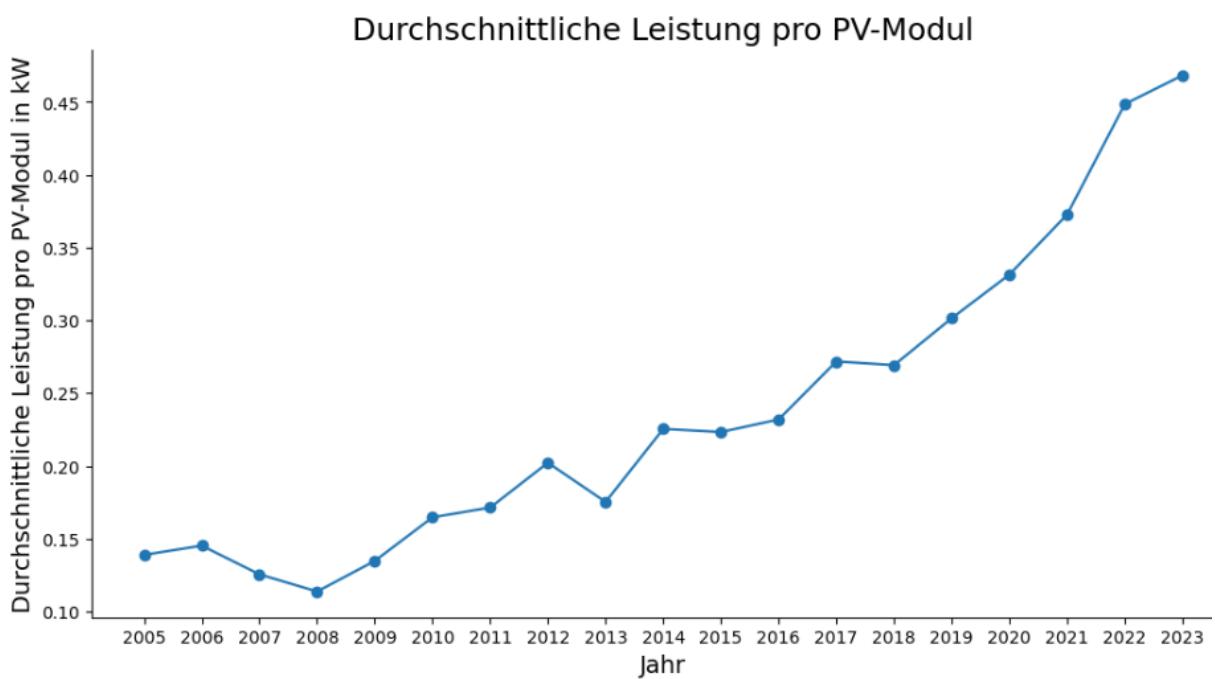
Tabelle 8: Top-20 Gemeinden nach installierter Bruttoleistung (kWh) pro 10.000 Einwohner.

Im Gegensatz zur Anzahl an installierten Anlagen befinden sich im Ranking um die installierte Leistung nicht nur Gemeinden aus Bayern. Das Feld führen drei Gemeinden aus Brandenburg an.

<b>Ort</b>	<b>Bundesland</b>	<b>AGS-5</b>	<b>Bruttoleistung_pro_10k_Einwohner</b>
Jena	Thüringen	16053	1477
Gelsenkirchen	Nordrhein-Westfalen	05513	1450
Bremerhaven	Bremen	04012	1387
Bonn	Nordrhein-Westfalen	05314	1369
Dresden	Sachsen	14612	1357
Wuppertal	Nordrhein-Westfalen	05124	1348
Bremen	Bremen	04011	1313
Wiesbaden	Hessen	06414	1286
Duisburg	Nordrhein-Westfalen	05112	1225
Stuttgart	Baden-Württemberg	08111	1183
Köln	Nordrhein-Westfalen	05315	1181
Offenbach	Hessen	06413	1109
Kiel	Schleswig-Holstein	01002	1075
Essen	Nordrhein-Westfalen	05113	1074
Herne	Nordrhein-Westfalen	05916	1030
Düsseldorf	Nordrhein-Westfalen	05111	886
München	Bayern	09162	822
Frankfurt	Hessen	06412	794
Berlin	Berlin	11000	655
Hamburg	Hamburg	02000	558

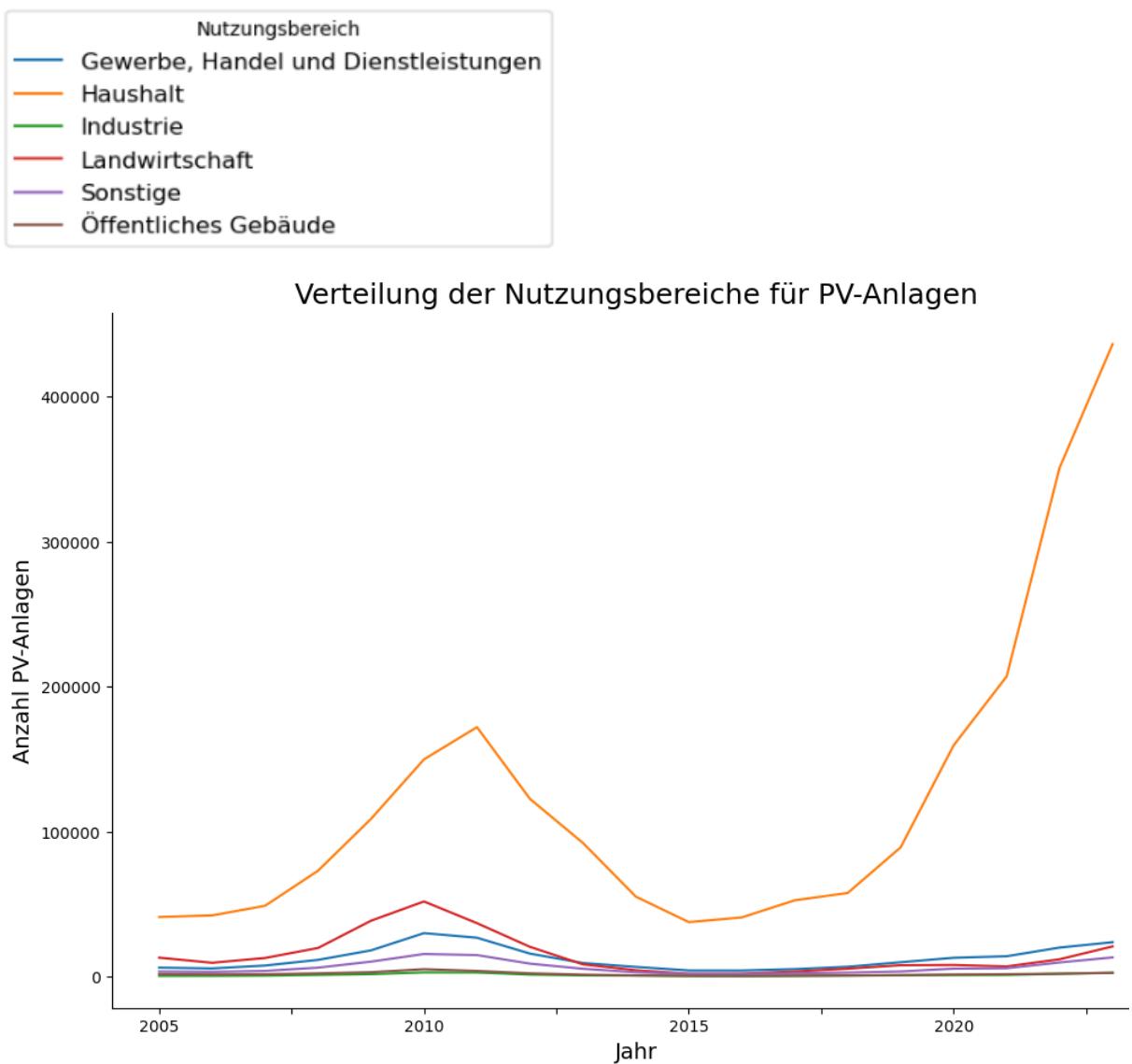
Tabelle 9: Bottom-20 Gemeinden nach installierter Bruttoleistung (kWh) pro 10.000 Einwohner.

Sowohl die Anzahl an installierten Anlagen als auch die hier dargestellte installierte Bruttoleistung ist in größeren Städten / Großstädten am geringsten. Die Schlusslichter bilden Hamburg und Berlin.



**Abbildung 15:** Entwicklung der durchschnittlichen Leistung pro installiertem PV-Modul von 2005 bis 2023.

Zwischen 2005 und 2023 hat sich die durchschnittliche Leistung mehr als verdreifacht und stieg von 0,14 auf 0,47 kWh pro Modul. Insbesondere seit 2021 gab es einen starken Anstieg der Leistung pro Modul. Durch diese Effizienzsteigerung können selbst kleine Anlagen einen hohen Ertrag erwirtschaften.



**Abbildung 16:** Verteilung der Nutzungsbereiche aller installierten PV-Anlagen von 2005 bis 2023.

Maßgeblicher Treiber beim PV-Ausbau sind private Haushalte, die den weit größten Anteil aller installierten PV-Anlagen ausmachen.

Bis 2011 haben PV-Anlagen in der Landwirtschaft noch einen wesentlichen Anteil am gesamten PV-Ausbau ausgemacht.

Nach dem Einbruch des Ausbaus zwischen 2012 und 2018 wurde dieses Niveau nicht wieder erreicht.

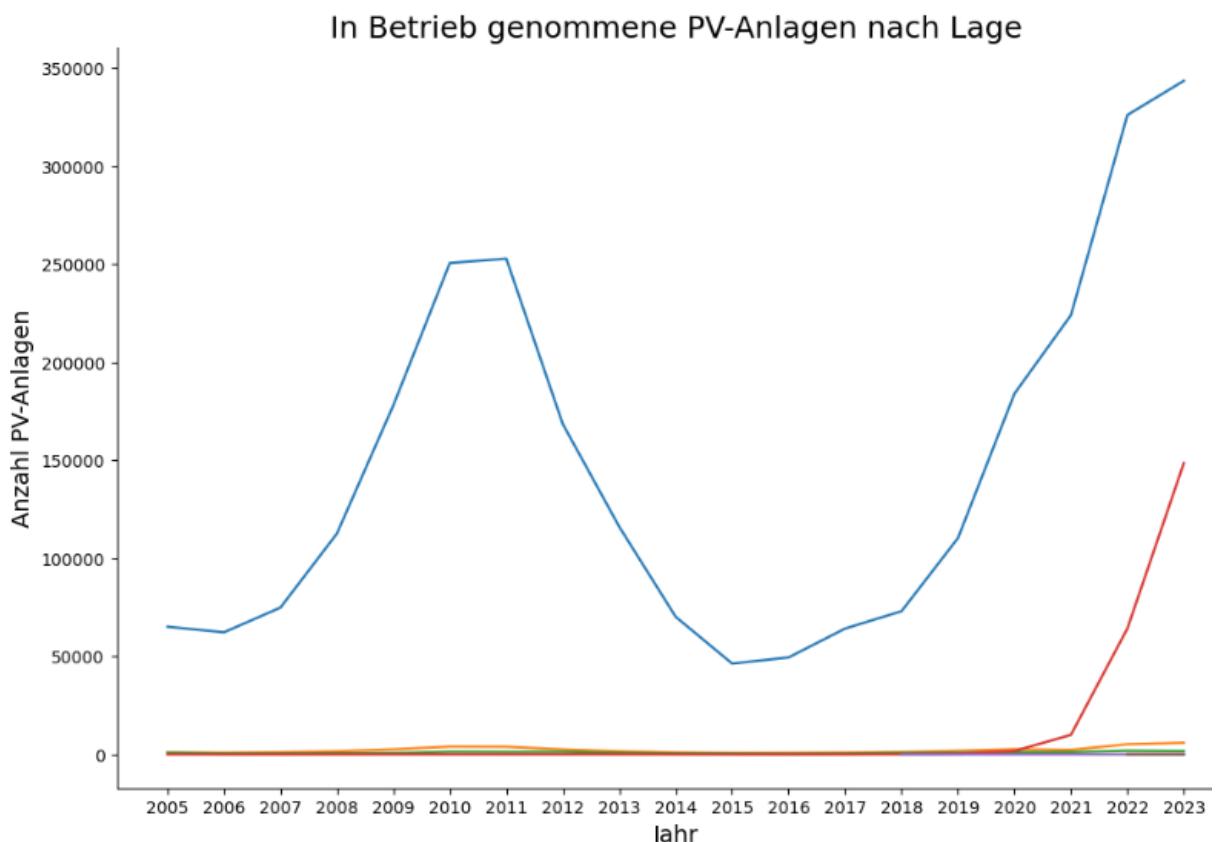
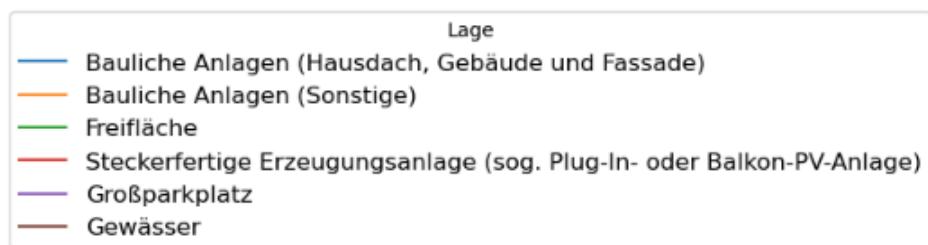
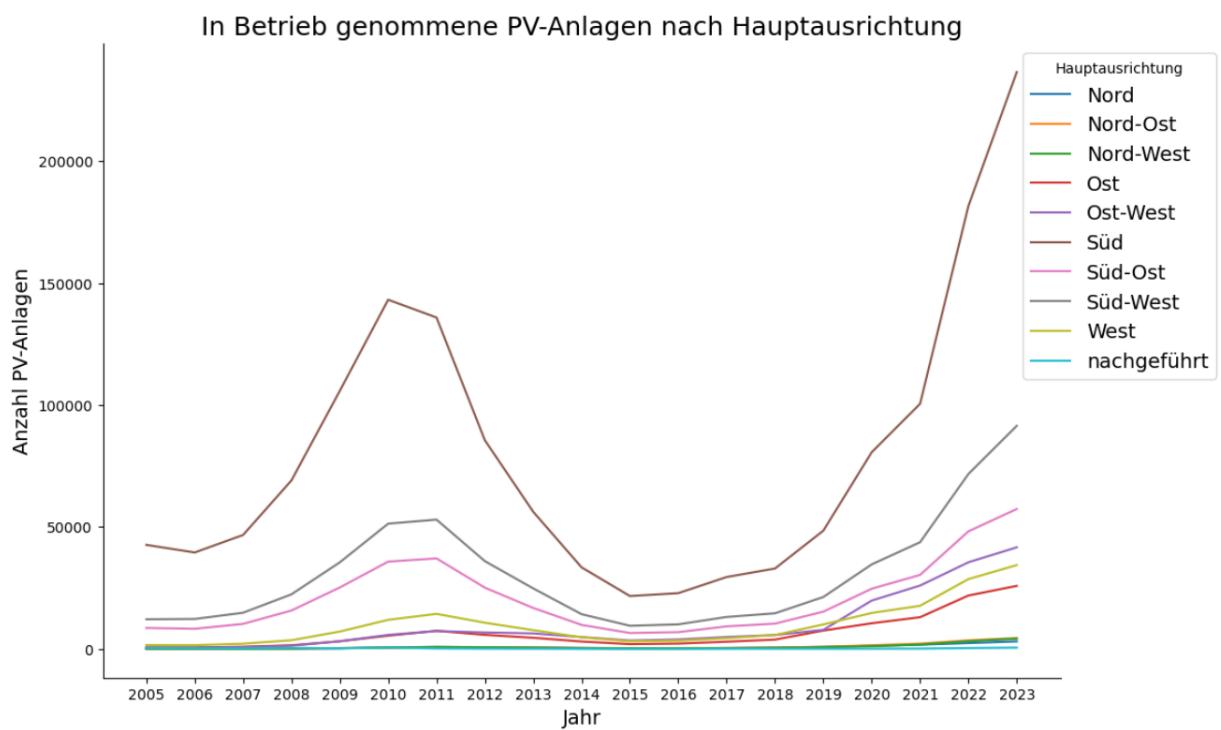


Abbildung 17: Verteilung der Lage aller installierten PV-Anlagen von 2005 bis 2023.

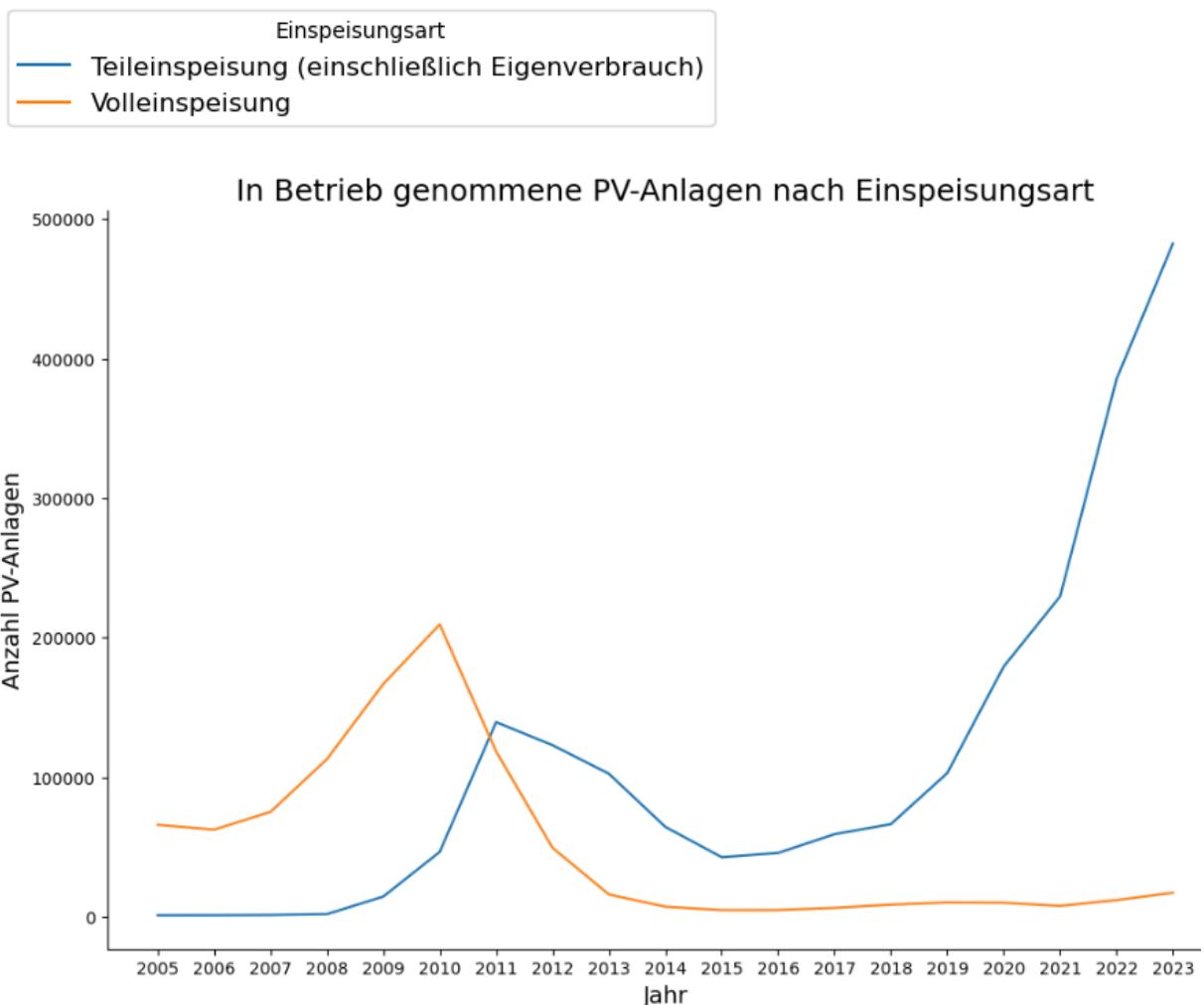
Der weit größte Anteil von PV-Anlagen wird auf baulichen Anlagen, wie Hausdächern, Gebäuden oder Fassaden installiert.

Seit 2021 gibt es einen sprunghaften Anstieg sogenannter Plug-In oder Balkon-Anlagen.



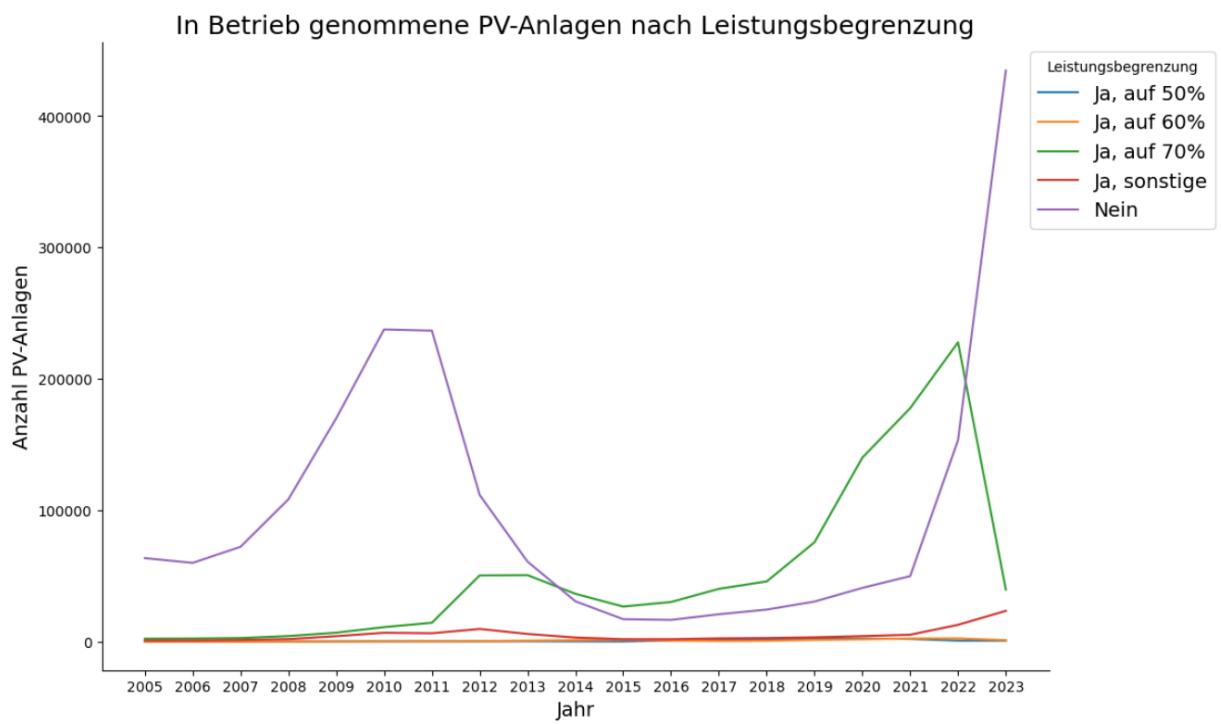
**Abbildung 18:** Entwicklung der Hauptausrichtung der installierten PV-Anlagen von 2005 bis 2023.

Es dominiert die wirtschaftlich günstigste Südausrichtung, gefolgt von Süd-West und Süd-Ost.



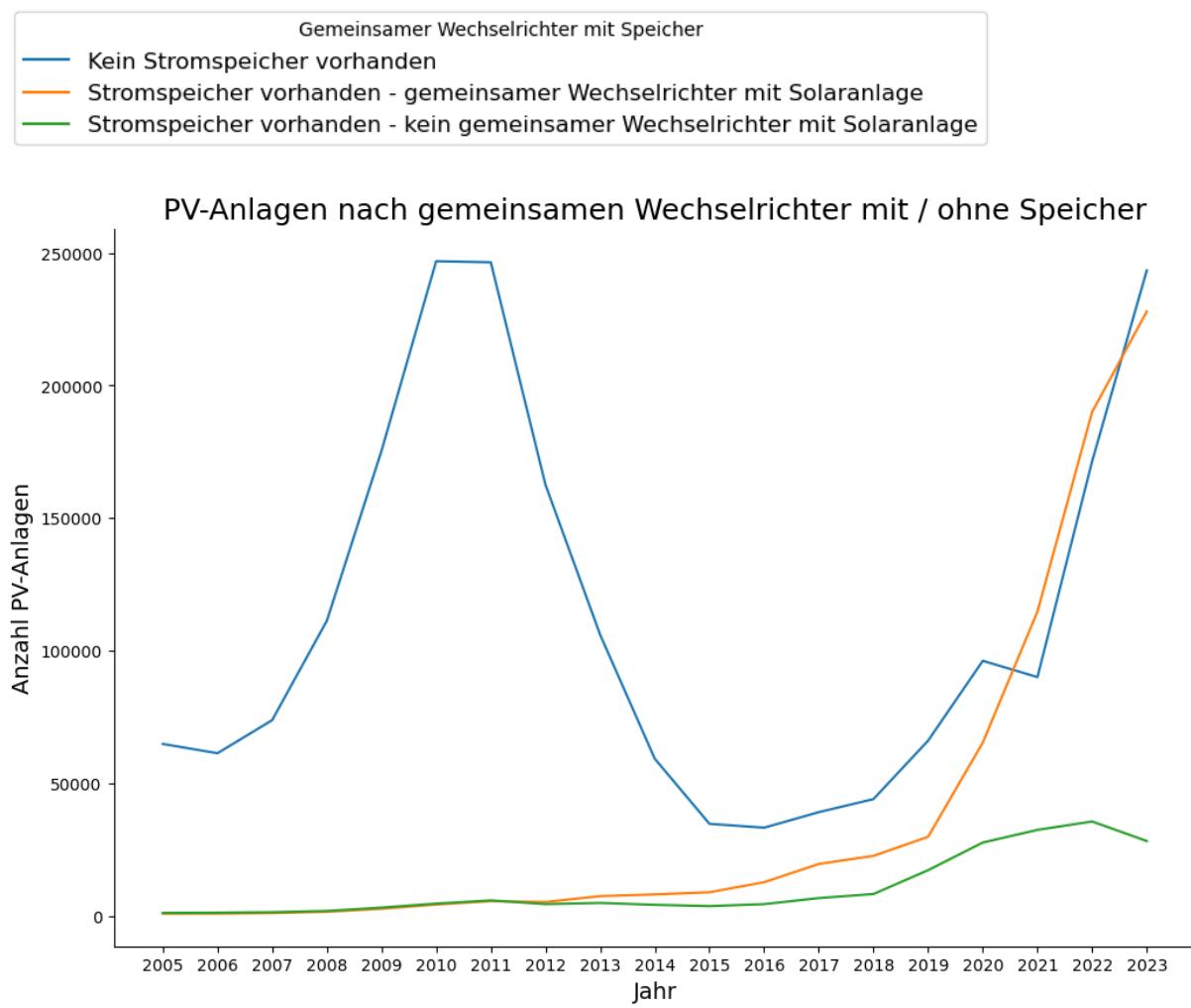
**Abbildung 19:** Entwicklung der Einspeisungsarten Voll- und Teileinspeisung für zwischen 2005 und 2023 installierte PV-Anlagen.

Zwischen 2010 und 2011 gibt es einen grundlegenden Wechsel weg von der Volleinspeisung und hin zur Teileinspeisung. Der erzeugte Strom wird also nicht mehr verkauft, sondern in erster Linie selbst verbraucht.



**Abbildung 20:** Entwicklung der Leistungsbegrenzung der installierten PV-Anlagen von 2005 bis 2023.

Während bis 2021 zunehmend Anlagen mit einer Leistungsbegrenzung von 70% installiert wurden, werden seit 2021 fast ausschließlich Anlagen ohne Leistungsbegrenzung eingebaut.



**Abbildung 21:** Entwicklung nach Installation von PV-Anlagen mit Stromspeichern und mit oder ohne gemeinsamen Wechselrichter von 2005 bis 2023.

Installierte Bruttoleistung in Deutschland 2005-2008

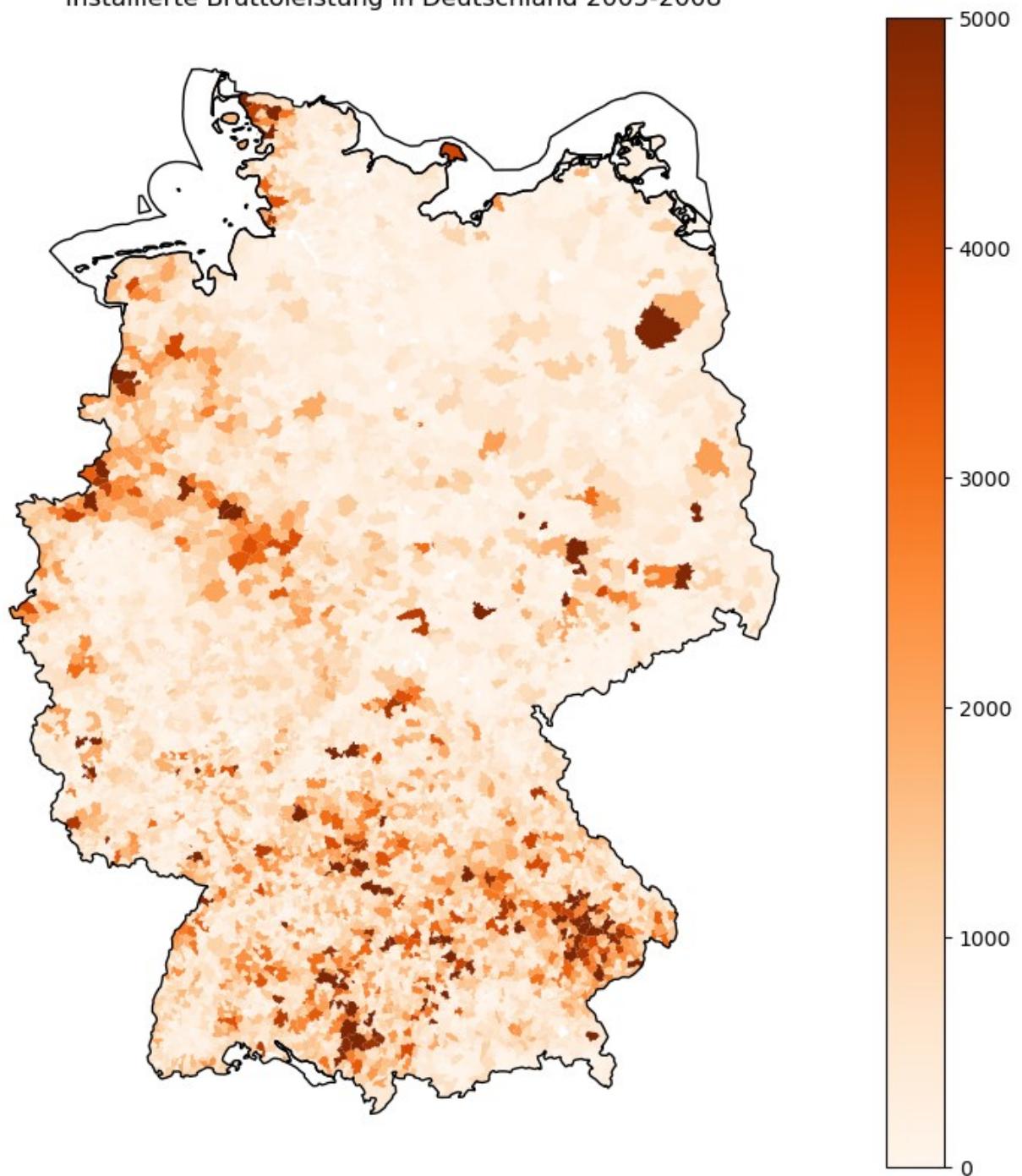


Abbildung 22: Installierte Bruttoleistung in kWh nach Postleitzahlengebiet im Zeitraum 2005 bis 2008.

Installierte Bruttolleistung in Deutschland 2009-2012

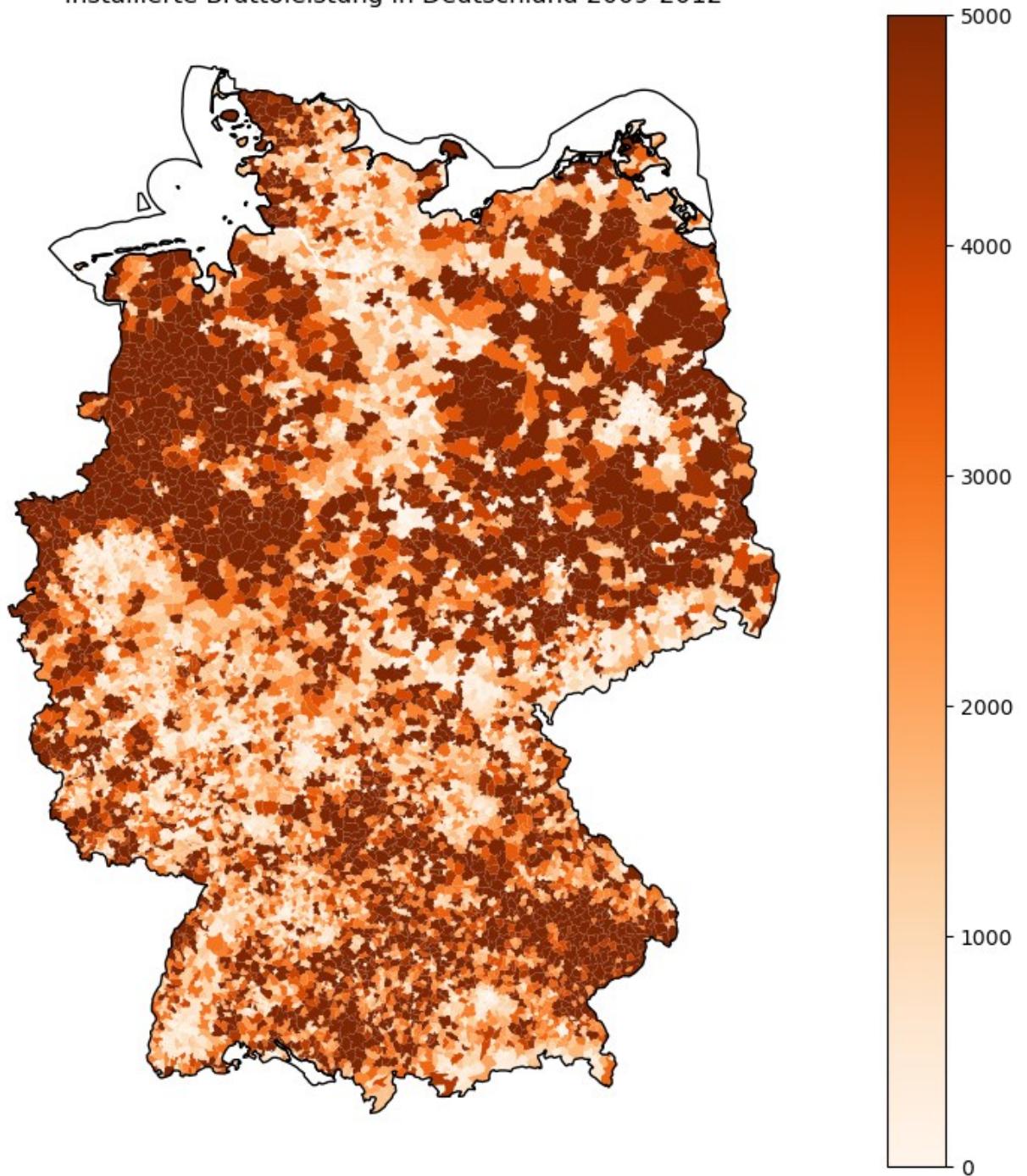
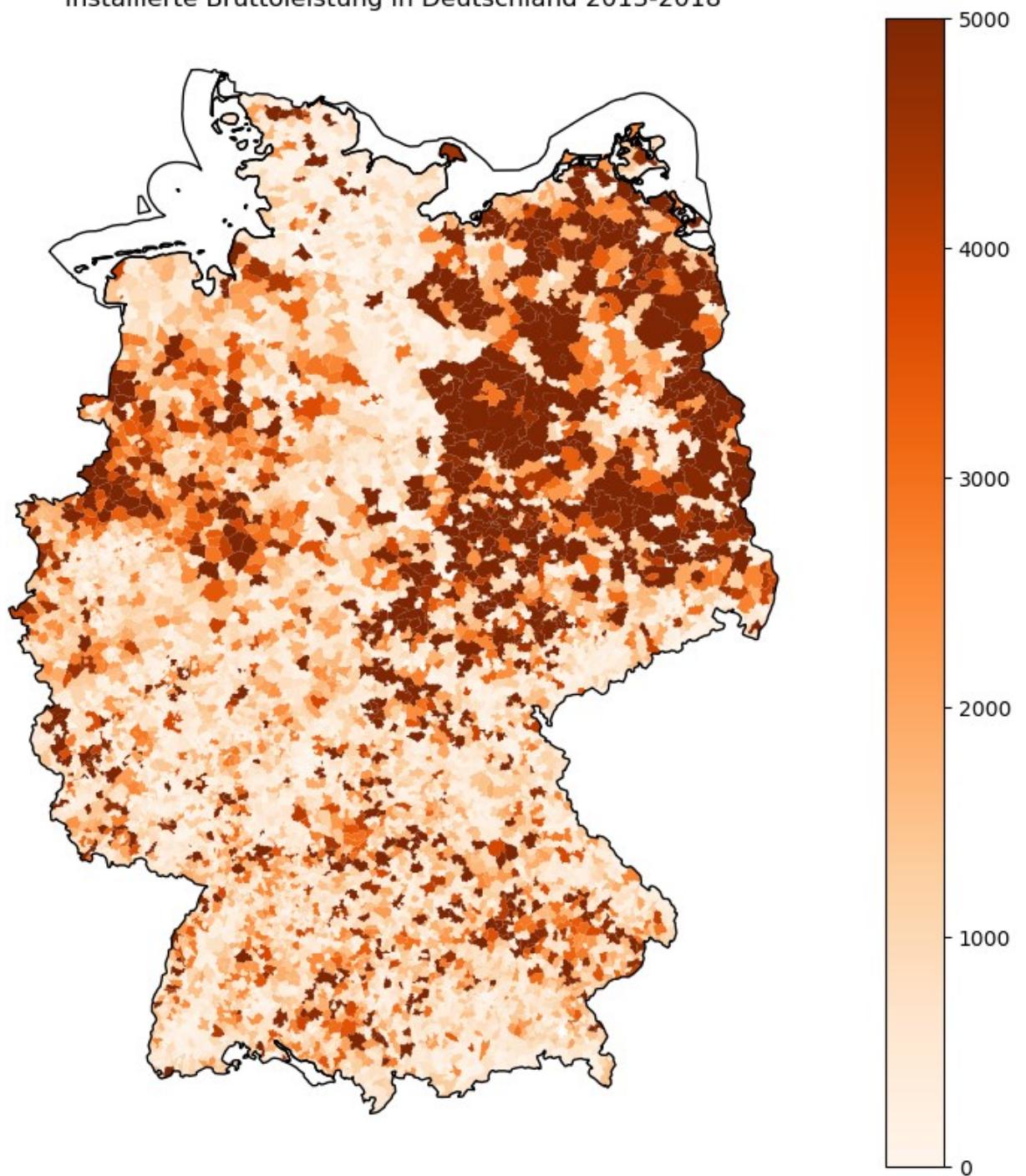


Abbildung 23: Installierte Bruttolleistung in kWh nach Postleitzahlengebiet im Zeitraum 2009 bis 2012

Installierte Bruttolleistung in Deutschland 2013-2018



**Abbildung 24:** Installierte Bruttolleistung in kWh nach Postleitzahlengebiet im Zeitraum 2013 bis 2018.

Installierte Bruttoleistung in Deutschland 2019-2023

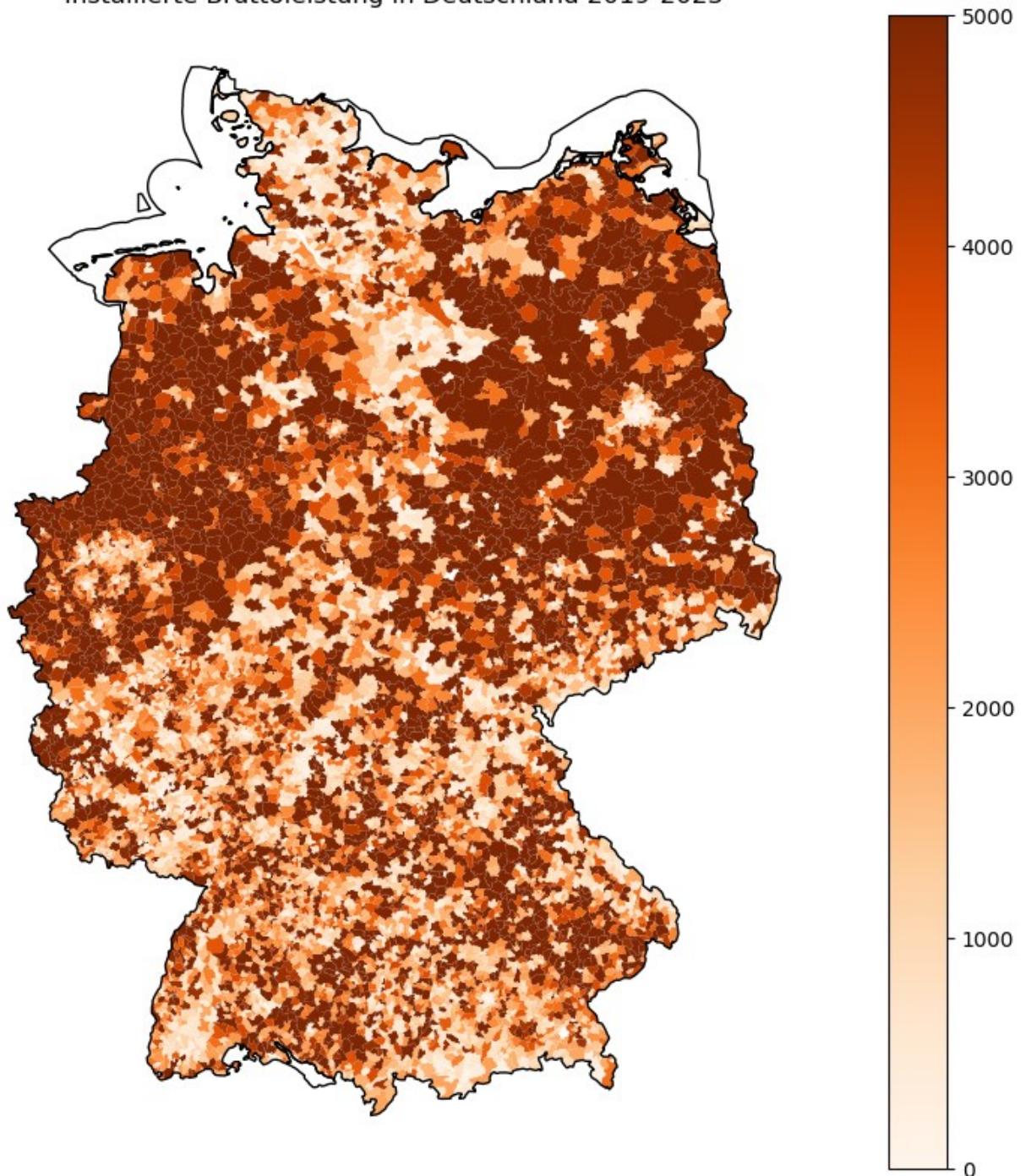
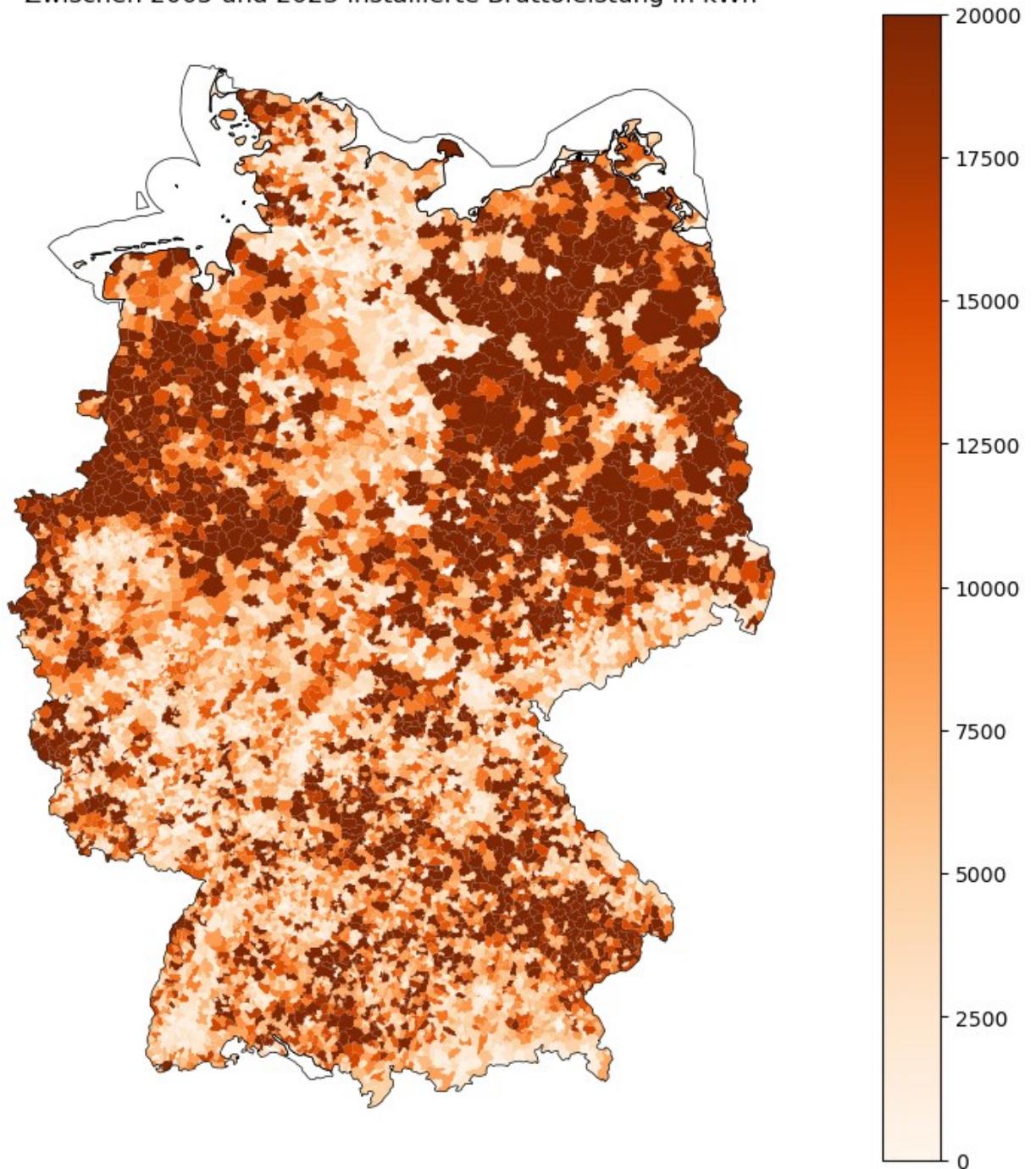
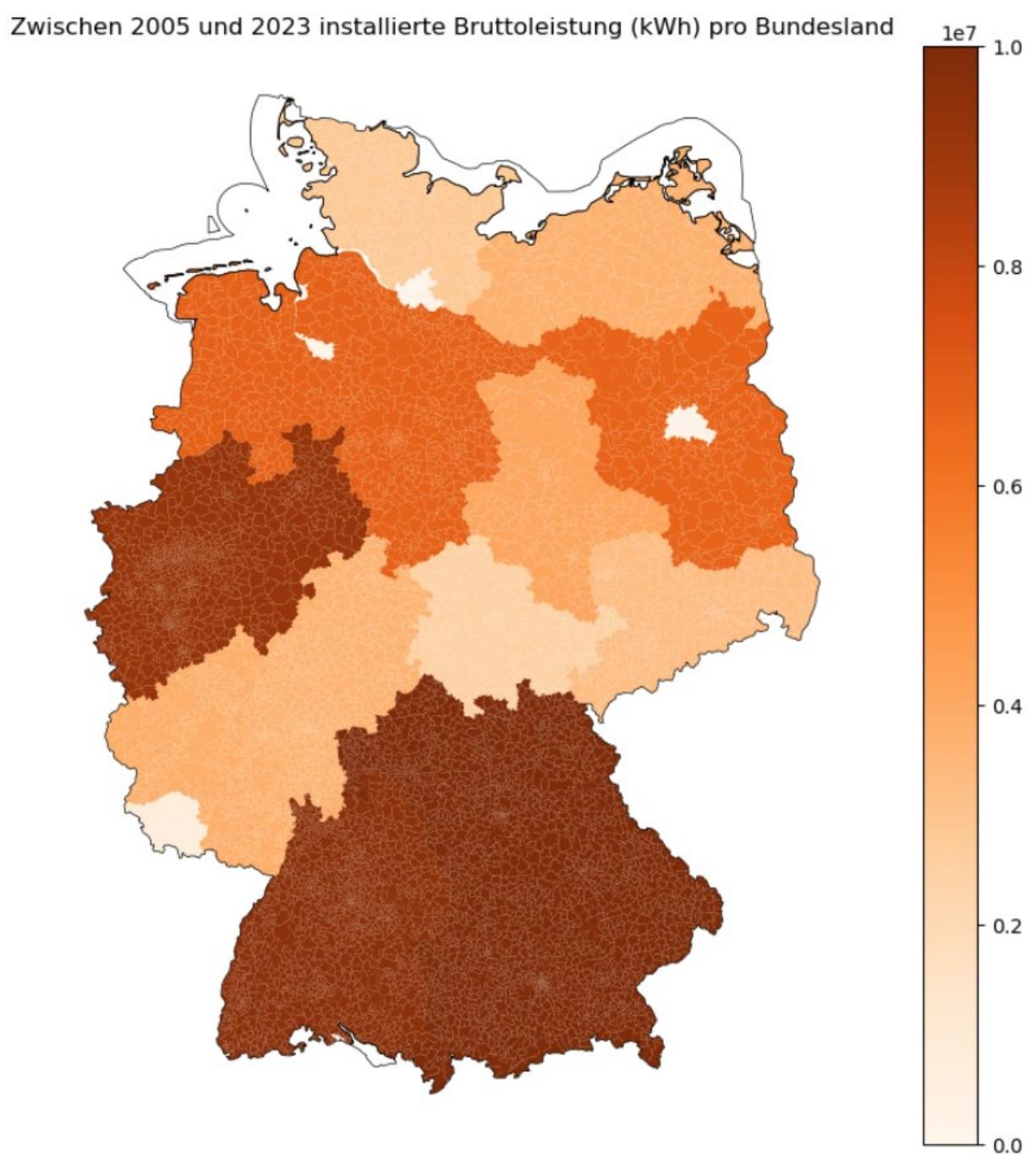


Abbildung 25: Installierte Bruttoleistung in kWh nach Postleitzahlengebiet im Zeitraum 2019 bis 2023.

Zwischen 2005 und 2023 installierte Bruttoleistung in kWh



**Abbildung 26:** Installierte Bruttoleistung in kWh nach Postleitzahlengebiet im Gesamt-Zeitraum 2005 bis 2023.

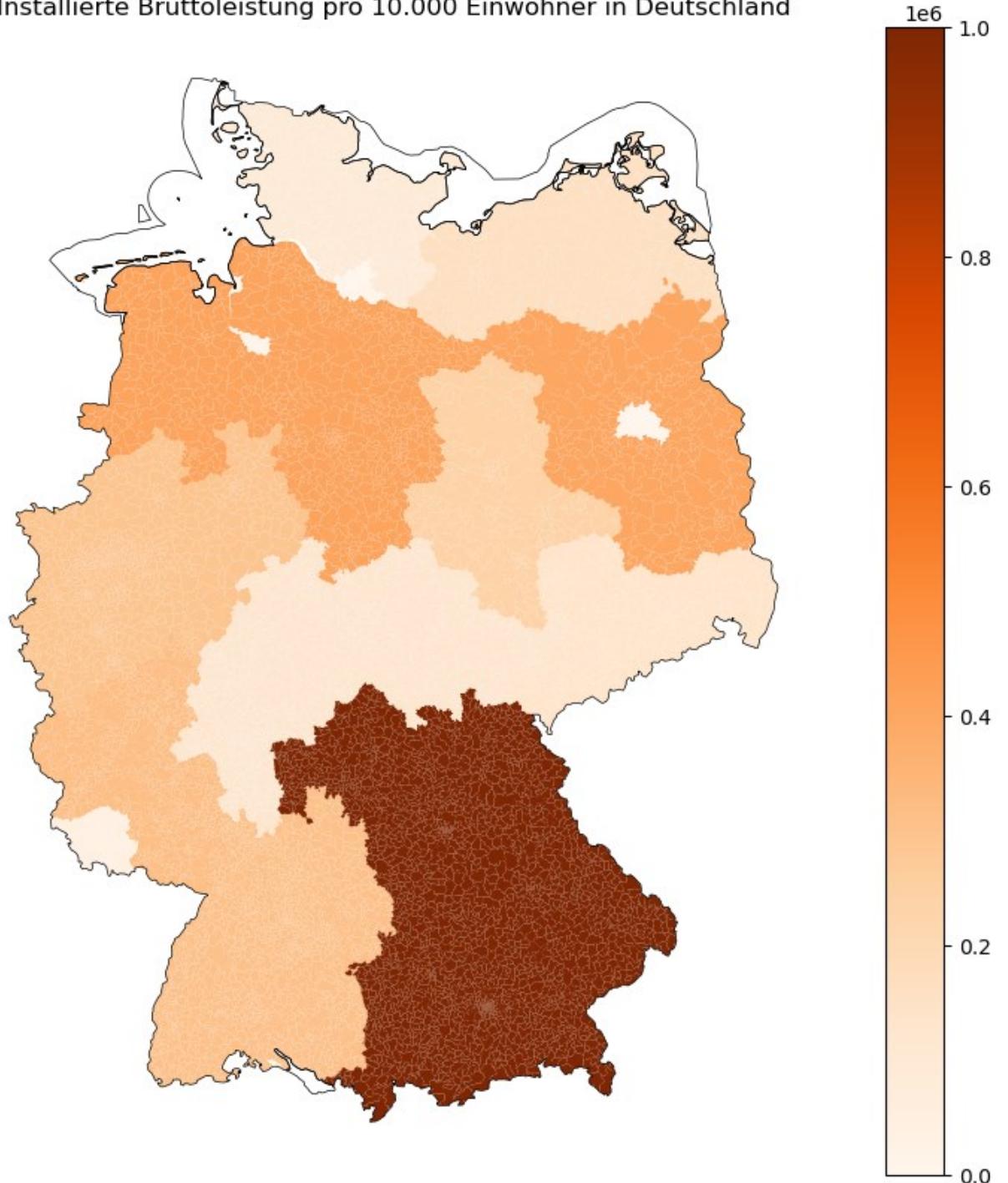


**Abbildung 27:** Installierte Bruttoleistung in kWh nach Bundesländern im Gesamt-Zeitraum 2005 bis 2023.

Bundesland	Bruttoleistung	Ranking
Bayern	2.108105e+07	1
Baden-Württemberg	9.587048e+06	2
Nordrhein-Westfalen	9.210430e+06	3
Niedersachsen	6.721980e+06	4
Brandenburg	6.664551e+06	5
Sachsen-Anhalt	4.174769e+06	6
Rheinland-Pfalz	3.843363e+06	7
Mecklenburg-Vorpommern	3.738120e+06	8
Hessen	3.619059e+06	9
Sachsen	3.262856e+06	10
Schleswig-Holstein	2.843289e+06	11
Thüringen	2.423419e+06	12
Saarland	8.114290e+05	13
Berlin	2.448674e+05	14
Hamburg	1.057363e+05	15
Bremen	9.055026e+04	16

**Tabelle 10:** Ranking installierte Bruttoleistung in kWh nach Bundesländern im Zeitraum von 2005 bis 2023.

Installierte Bruttolleistung pro 10.000 Einwohner in Deutschland

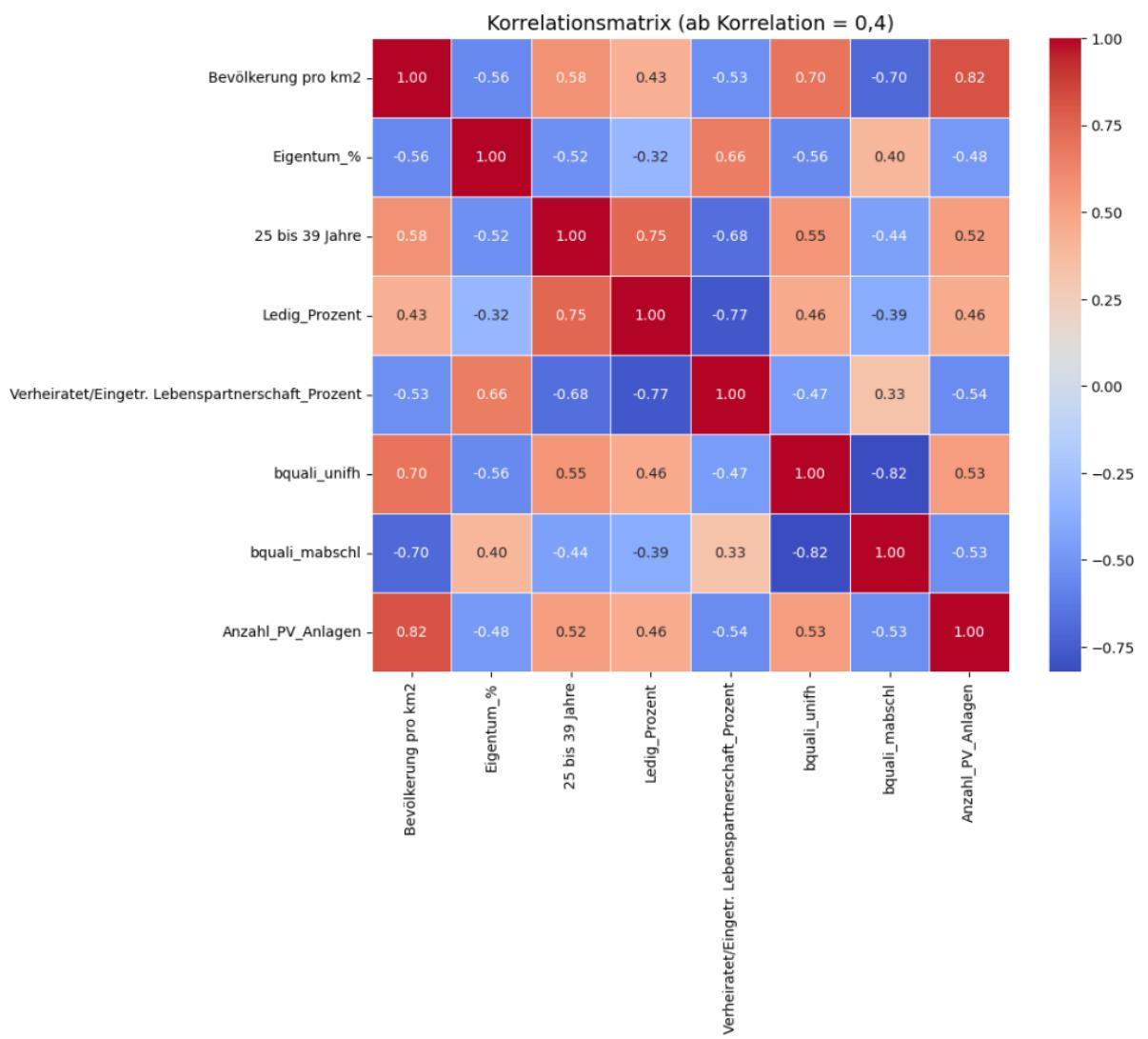


**Abbildung 28:** Installierte Bruttolleistung pro 10.000 Einwohner in kWh nach Bundesländern im Gesamt-Zeitraum 2005 bis 2023.

Bundesland	Bruttoleistung_pro_10k_Einwohner	Ranking
Bayern	1.079834e+06	1
Niedersachsen	4.073088e+05	2
Brandenburg	4.017065e+05	3
Rheinland-Pfalz	3.131712e+05	4
Baden-Württemberg	3.043420e+05	5
Nordrhein-Westfalen	2.898978e+05	6
Sachsen-Anhalt	2.367548e+05	7
Mecklenburg-Vorpommern	1.587643e+05	8
Sachsen	1.342159e+05	9
Hessen	1.161217e+05	10
Thüringen	1.136544e+05	11
Schleswig-Holstein	8.932003e+04	12
Saarland	4.189945e+04	13
Bremen	8.001048e+03	14
Berlin	6.658581e+02	15
Hamburg	5.703344e+02	16

25

**Tabelle 11:** Ranking installierte Bruttoleistung in kWh pro 10.000 Einwohner nach Bundesländern im Zeitraum von 2005 bis 2023.



**Abbildung 29:** Korrelationsmatrix aller numerischen Variablen ab einer Korrelation von 0,4.

```

OLS Regression Results
=====
Dep. Variable: Anzahl_PV_Anlagen R-squared:      0.391
Model:           OLS   Adj. R-squared:      0.391
Method:          Least Squares F-statistic:    3571.
Date:           Sat, 18 Nov 2023 Prob (F-statistic): 0.00
Time:            14:17:28 Log-Likelihood: -82531.
No. Observations: 11107 AIC:             1.651e+05
Df Residuals:    11104 BIC:             1.651e+05
Df Model:        2
Covariance Type: nonrobust
=====
              coef    std err      t    P>|t|    [0.025    0.975]
-----
Intercept     418.9541   22.182   18.887   0.000    375.474   462.434
Eigenumsquote -5.0825    0.261   -19.453   0.000    -5.595    -4.570
Bevoelkerungsdichte 1.5029   0.020    75.424   0.000    1.464    1.542
=====
Omnibus:       14438.289 Durbin-Watson:      1.185
Prob(Omnibus): 0.000   Jarque-Bera (JB): 13537592.683
Skew:          6.631   Prob(JB):        0.00
Kurtosis:      173.517 Cond. No.       1.57e+03
=====

```

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 1.57e+03. This might indicate that there are strong multicollinearity or other numerical problems.

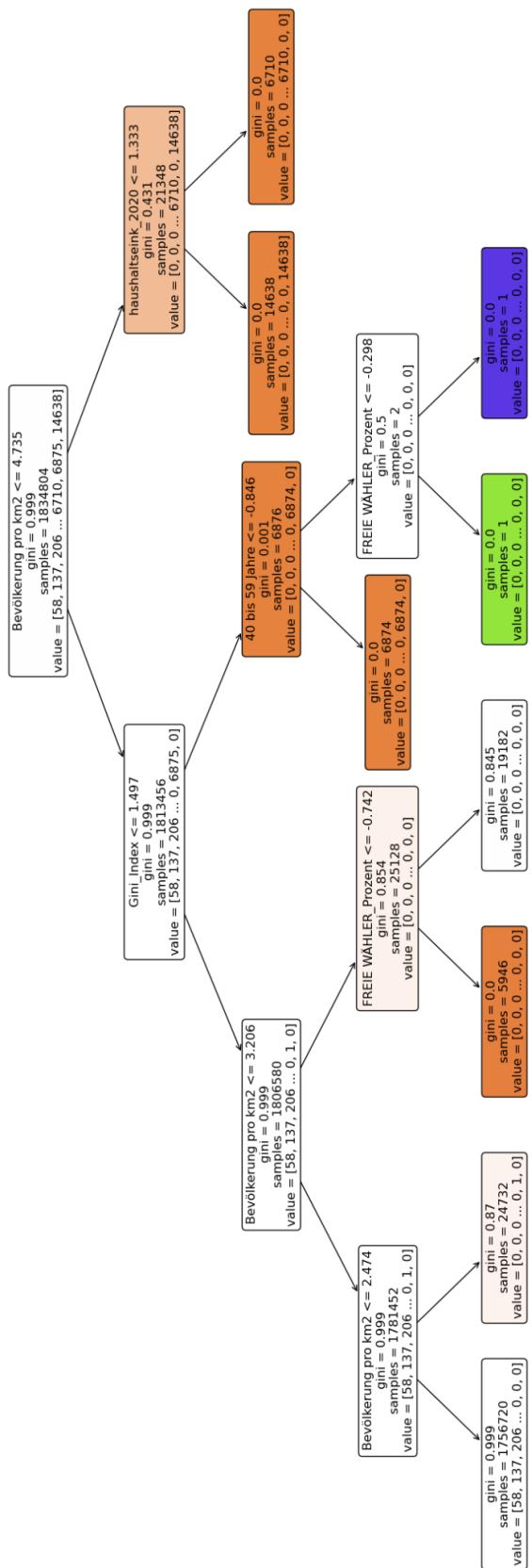
**Tabelle 12:** Ergebnisse der OLS-Regression (Anzahl\_PV\_Anlagen ~ Eigenumsquote + Bevoelkerungsdichte)

```

OLS Regression Results
=====
Dep. Variable: Anzahl_PV_Anlagen R-squared:      0.337
Model:           OLS   Adj. R-squared:      0.336
Method:          Least Squares F-statistic:    2159.
Date:        Tue, 09 Apr 2024 Prob (F-statistic): 0.00
Time:         12:49:09 Log-Likelihood:     -65759.
No. Observations: 8514   AIC:             1.315e+05
Df Residuals:    8511   BIC:             1.315e+05
Df Model:        2
Covariance Type: nonrobust
=====
                                         coef      std err      t      P>|t|      [0.025]      [0.975]
-----
Intercept                      -3508.4136    70.839   -49.527      0.000    -3647.275    -3369.552
Strom_Verbraucherpreis          118.2114     1.995    59.267      0.000     114.302     122.121
Einspeiseverguetung_kleine_PV_Anlagen 31.2173     0.751    41.589      0.000      29.746     32.689
-----
Omnibus:                  12461.055 Durbin-Watson:       0.773
Prob(Omnibus):            0.000 Jarque-Bera (JB): 12037065.613
Skew:                     8.455 Prob(JB):        0.00
Kurtosis:                 186.426 Cond. No.       481.
-----
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

**Tabelle 13:** Ergebnisse der OLS-Regression ( $\text{Anzahl\_PV\_Anlagen} \sim \text{Strom\_Verbraucherpreis} + \text{Einspeiseverguetung\_kleine\_PV\_Anlagen}$ )



**Abbildung 30:** Entscheidungsbaum zum Einfluss sozioökonomische Einflussfaktoren auf die Anzahl installierter Solaranlagen in einer Region.

```

Feature: Bevölkerung pro km2, Importance: 0.5136237620359968
Feature: Gini_Index, Importance: 0.3140037907619937
Feature: 60 bis 66 Jahre, Importance: 0.10254088146124327
Feature: eAuto, Importance: 0.06975335172880169
Feature: bquali_unifh, Importance: 5.8656242444302775e-05
Feature: Eigentum_%, Importance: 1.9557769520272196e-05
Feature: Bruttoleistung, Importance: 0.0
Feature: AnzahlModule, Importance: 0.0
Feature: haushaltseink_2011, Importance: 0.0
Feature: haushaltseink_2020, Importance: 0.0
Feature: Unter 3 Jahre, Importance: 0.0
Feature: 3 bis 5 Jahre, Importance: 0.0
Feature: 6 bis 9 Jahre, Importance: 0.0
Feature: 10 bis 15 Jahre, Importance: 0.0
Feature: 16 bis 18 Jahre, Importance: 0.0
Feature: 19 bis 24 Jahre, Importance: 0.0
Feature: 25 bis 39 Jahre, Importance: 0.0
Feature: 40 bis 59 Jahre, Importance: 0.0
Feature: 75 Jahre und älter, Importance: 0.0
Feature: Ledig_Prozent, Importance: 0.0
Feature: Verheiratet/Eingetr. Lebenspartnerschaft_Prozent, Importance: 0.0
Feature: bquali_mabschl, Importance: 0.0
Feature: CDU_Prozent, Importance: 0.0
Feature: AfD_Prozent, Importance: 0.0
Feature: FDP_Prozent, Importance: 0.0
Feature: GRÜNE_Prozent, Importance: 0.0
Feature: FREIE WÄHLER_Prozent, Importance: 0.0
Feature: Sonnenstunden, Importance: 0.0
Feature: eLade, Importance: 0.0
Feature: Wohnen_EZFH, Importance: 0.0
Feature: Wohnen_MFH, Importance: 0.0
Feature: Heizung_Erneuerbare_%, Importance: 0.0
Feature: Einspeisungsart_Teileinspeisung (einschließlich Eigenverbrauch), Importance: 0.0
Feature: Einspeisungsart_Volleinspeisung, Importance: 0.0
Feature: Hauptausrichtung_Nord, Importance: 0.0
Feature: Hauptausrichtung_Nord-Ost, Importance: 0.0
Feature: Hauptausrichtung_Nord-West, Importance: 0.0
Feature: Hauptausrichtung_Ost, Importance: 0.0
Feature: Hauptausrichtung_Ost-West, Importance: 0.0
Feature: Hauptausrichtung_Süd, Importance: 0.0
Feature: Hauptausrichtung_Süd-Ost, Importance: 0.0
Feature: Hauptausrichtung_Süd-West, Importance: 0.0
Feature: Hauptausrichtung_West, Importance: 0.0
Feature: Hauptausrichtung_nachgeführt, Importance: 0.0
Feature: Nutzungsbereich_Gewerbe, Handel und Dienstleistungen, Importance: 0.0
Feature: Nutzungsbereich_Haushalt, Importance: 0.0
Feature: Nutzungsbereich_Industrie, Importance: 0.0
Feature: Nutzungsbereich_Landwirtschaft, Importance: 0.0
Feature: Nutzungsbereich_Sonstige, Importance: 0.0
Feature: Nutzungsbereich_Öffentliches Gebäude, Importance: 0.0

```

**Tabelle 14:** Feature Importance der sozioökonomischen Einflussfaktoren in Bezug auf die Anzahl installierter PV-Anlagen.

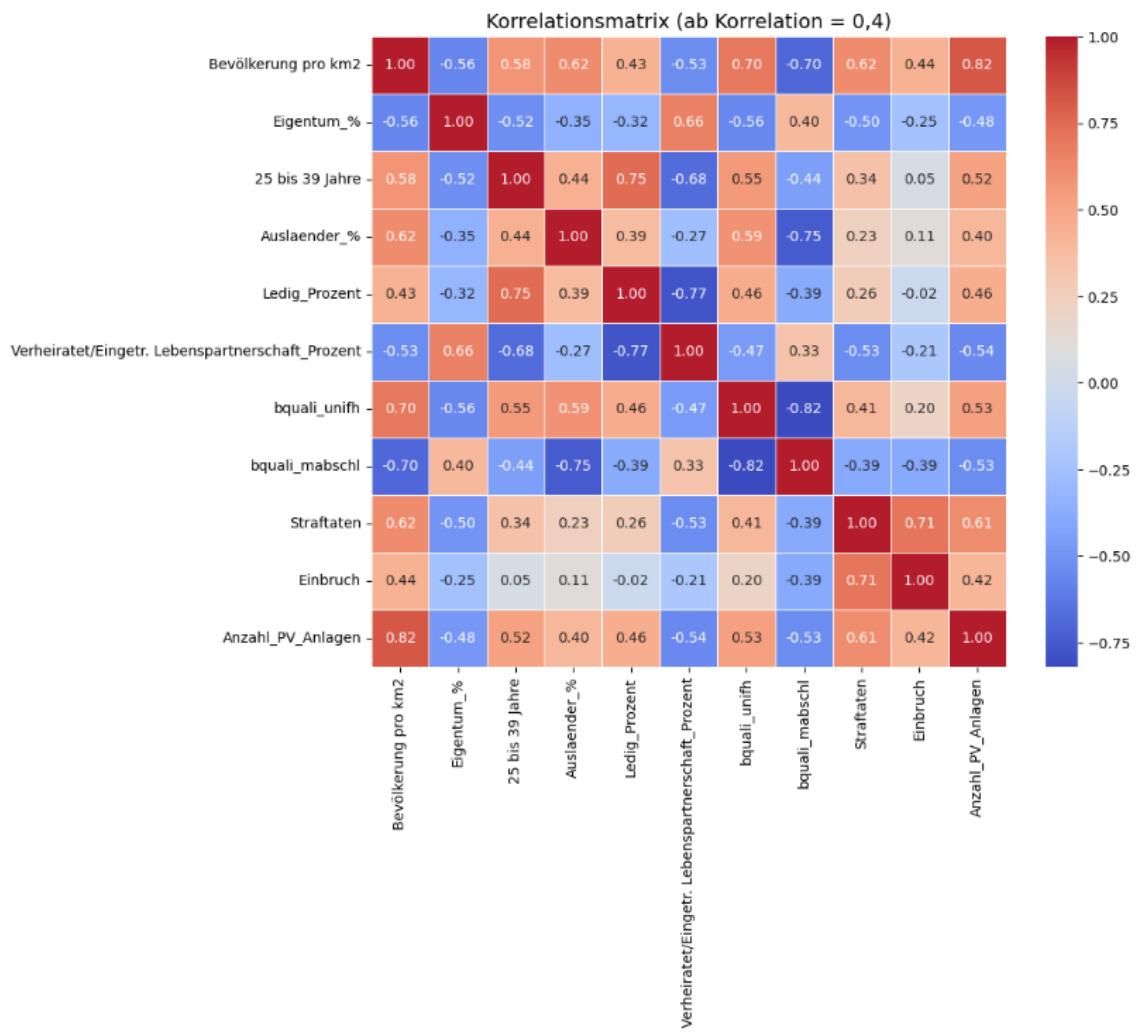


Abbildung 31: Korrelationsmatrix sozioökonomische Einflussfaktoren.

## Erkenntnisse:

Die Ergebnisse der Korrelationsanalyse und die Feature Importance der Decision Tree Analyse messen teilweise unterschiedlichen Features eine hohe Bedeutung bezüglich Entscheidung für eine PV-Anlage zu.

Beide Analysen sehen die **Bevölkerungsdichte** als wichtigsten Einflussfaktor.

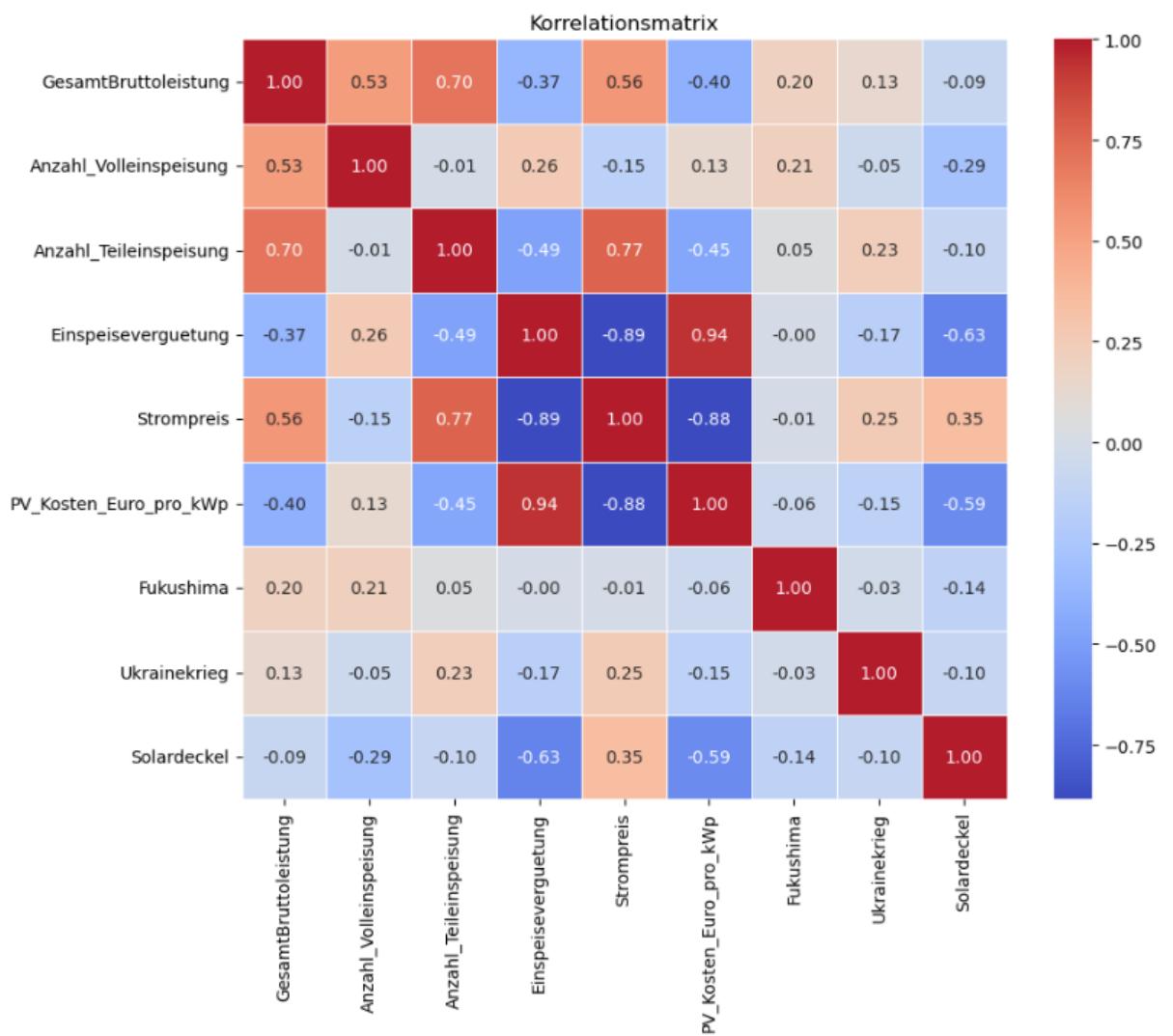
Während **Bildungsniveau, Familienstand und Eigenheimquote** zentrale Features in der Korrelationsmatrix darstellen, kommen beim Decision Tree ausser dem Familienstand auch der **Gini-Index, Alterstruktur, Wahlergebnisse (CDU und FDP) und der Anteil an erneuerbaren Heizmethoden in Neubauten** als mögliche Einflussfaktoren zum Tragen.

Mögliche Gründe für diese Abweichungen:

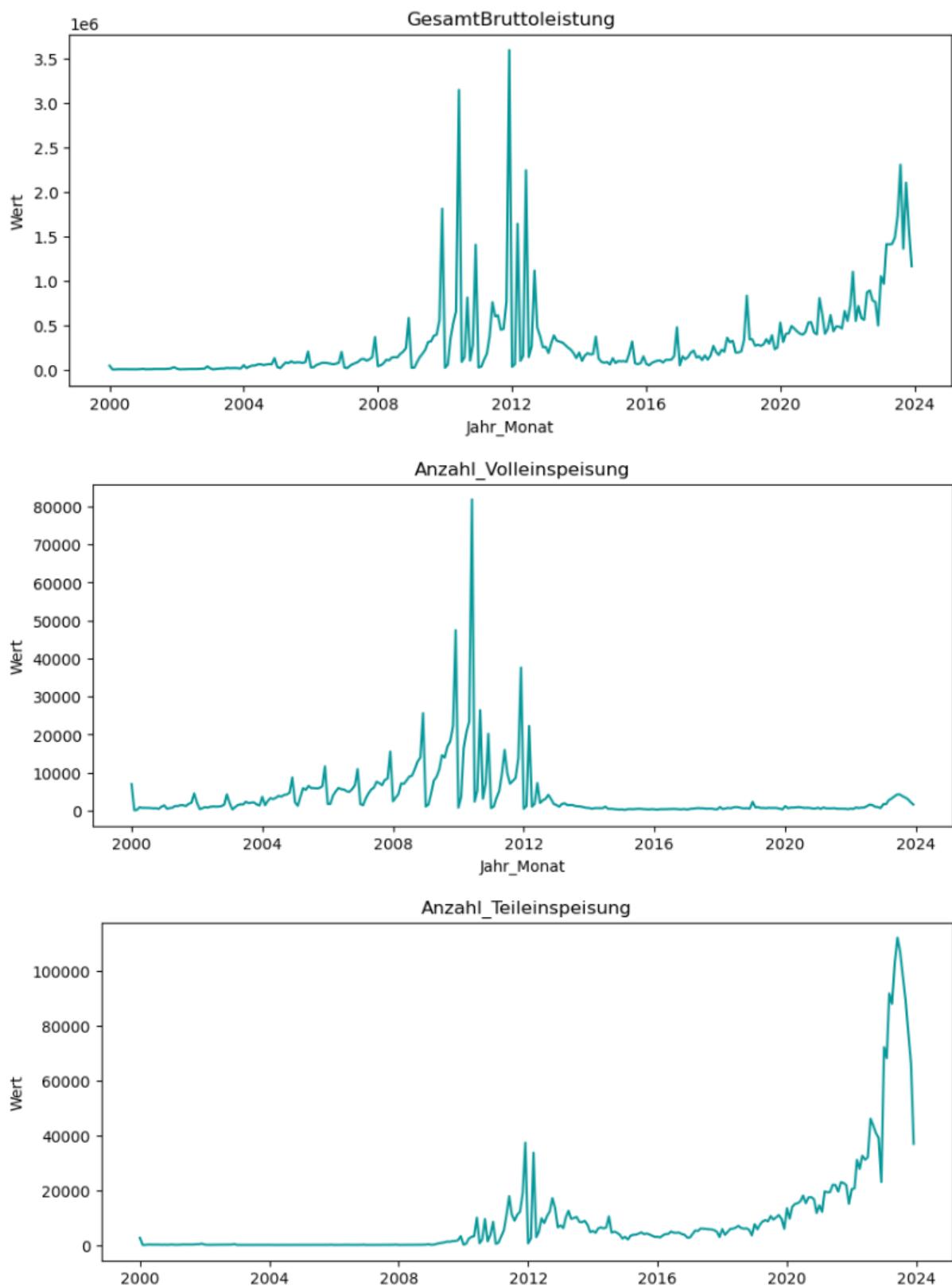
- Nicht-lineare Beziehungen (Korrelationsanalysen basieren ausschließlich auf linearen Beziehungen zwischen den Variablen.)
- Multikollinearität (Korrelationen zwischen den Features selbst.)
- Modellabhängigkeit (Unterschiedliche Modelle beinhalten unterschiedliche Berechnungsgrundlagen und Gewichtungen.)

Bewertung des Modells:

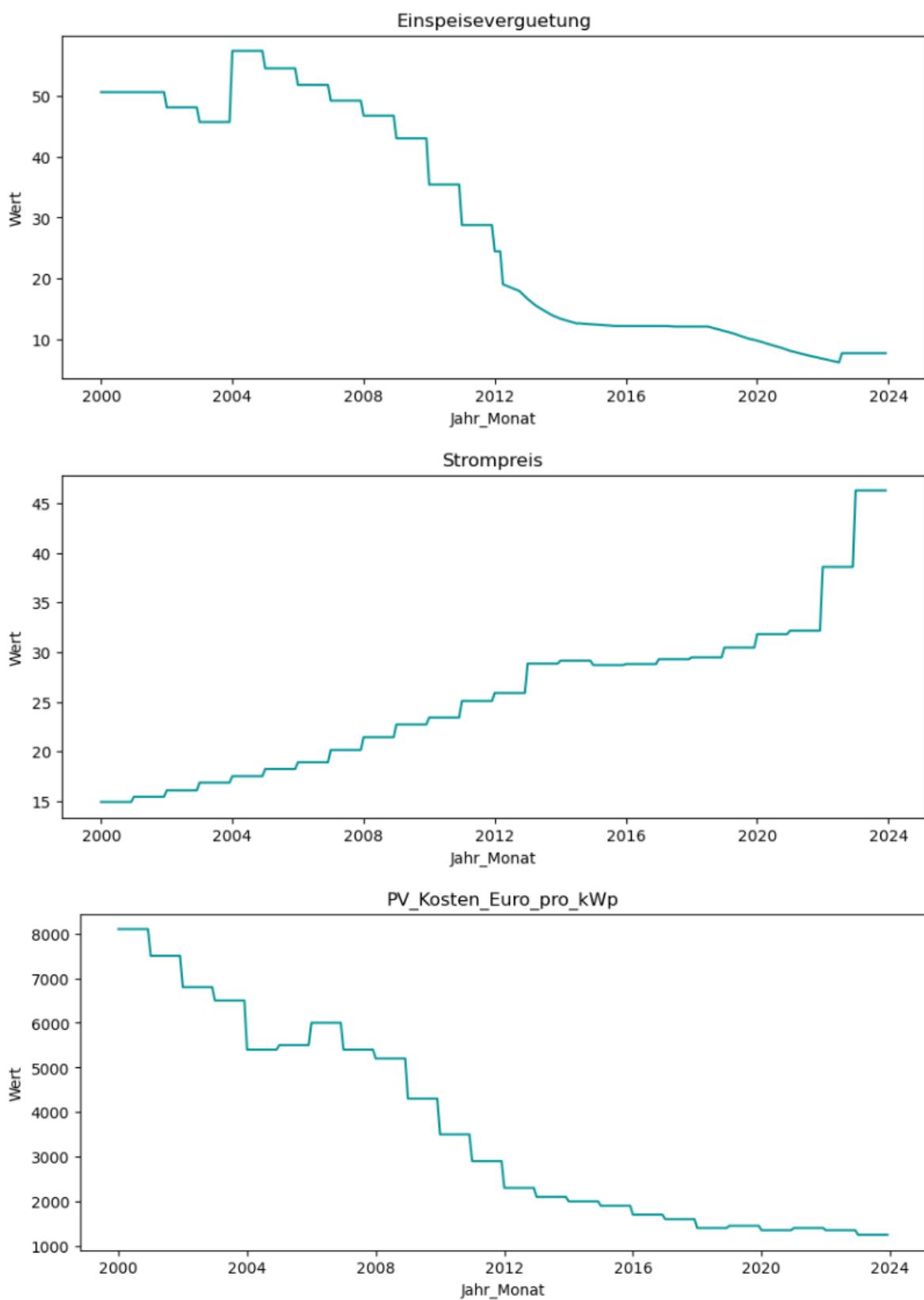
Der R<sup>2</sup>-Score des Entscheidungsbaums beträgt für die Trainingsdaten 0.03202 und für die Testdaten 0.03208. Die Werte liegen sehr nah beieinander, was für eine gute Anpassung des Modells spricht. Allerdings erklärt das Modell nur 3 Prozent der tatsächlich relevanten Einflussfaktoren und liefert damit **kein signifikantes Ergebnis.**



**Abbildung 32:** Korrelationsmatrix Kosten, Vergütungen und Ereignisse.



**Abbildung 33:** Visualisierung der Zeitreihen GesamtBruttoleistung, Anzahl\_Volleinspeisung und Anzahl\_Teileinspeisung.



**Abbildung 34:** Visualisierung der Zeitreihen Einspeisevergütung, Strompreis und PV\_Kosten\_Euro\_pro\_kWp.

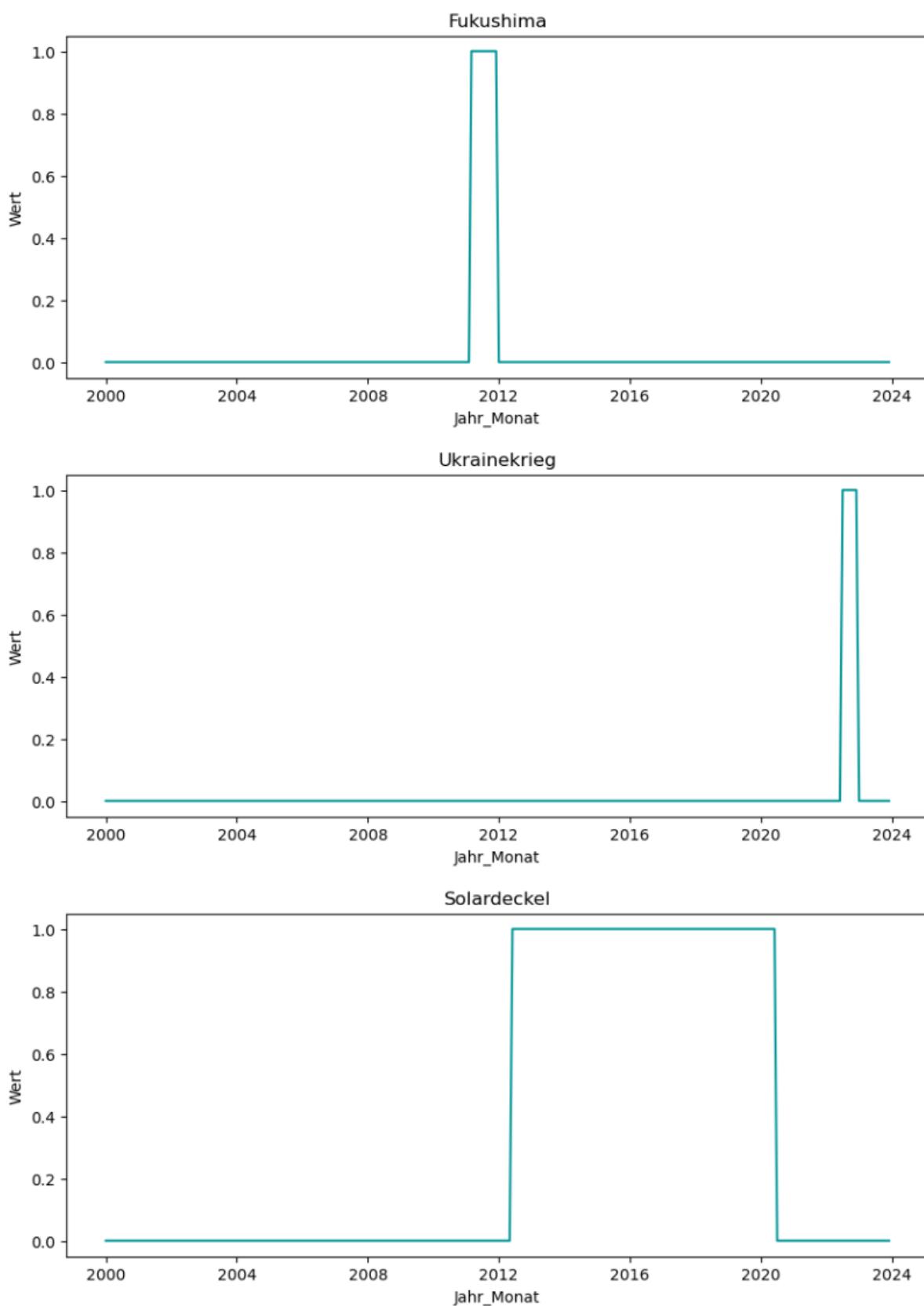


Abbildung 35: Visualisierung der Zeitreihen Fukushima, Ukrainekrieg und Solardeckel.

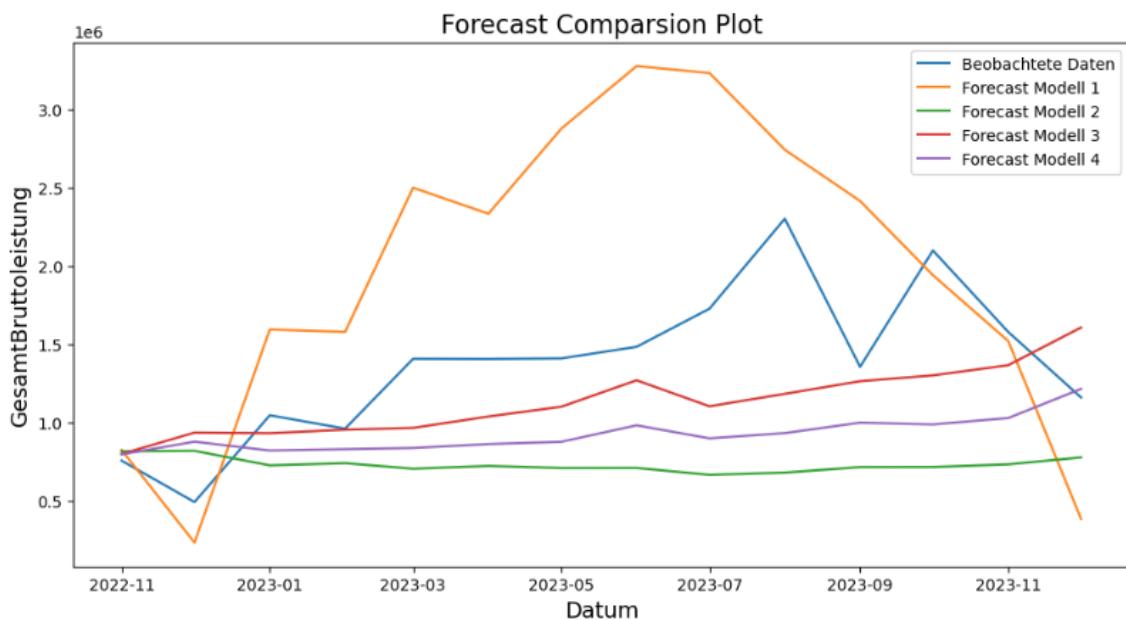


Abbildung 36: Vergleich der Sarimax-Modelle anhand Vorhersagen auf die Testdaten.

	Modell	RMSE	MAPE	MAE
0	Modell 1	940217.655049	0.576464	7.689138e+07
1	Modell 2	812418.184153	0.469980	6.939853e+07
2	Modell 3	478788.765972	0.272519	3.734282e+07
3	Modell 4	632189.569124	0.353165	5.140275e+07

Tabelle 15: Vergleich der Sarimax-Modelle anhand RMSE, MAPE und MAE.

```

SARIMAX Results
=====
Dep. Variable: GesamtBruttoleistung No. Observations: 274
Model: SARIMAX(9, 2, 5)x(2, 0, [1, 2, 3, 4, 5, 6, 7], 12) Log Likelihood -3698.050
Date: Sat, 09 Mar 2024 AIC 7460.101
Time: 10:28:15 BIC 7575.486
Sample: 01-01-2000 HQIC 7506.424
- 10-01-2022
Covariance Type: opg
=====
              coef    std err      z   P>|z|   [0.025   0.975]
-----
Anzahl_Volleinspeisung  2.925e+06  4.56e-07  6.41e+12  0.000  2.92e+06  2.92e+06
Anzahl_Teileinspeisung  4.458e+06  4.47e-07  9.97e+12  0.000  4.46e+06  4.46e+06
Einspeiseverguetung   -5.565e+05  2.32e-07 -2.4e+12  0.000 -5.57e+05 -5.57e+05
Strompreis            -2.505e+06  1.82e-07 -1.38e+13  0.000 -2.5e+06 -2.5e+06
PV_Kosten_Euro_pro_kWP 2.708e+05  2.8e-07  9.67e+11  0.000 2.71e+05  2.71e+05
Fukushima             6.449e+05  1.2e-06  5.37e+11  0.000 6.45e+05  6.45e+05
Ukrainekrieg          9.289e+04  8.31e-07  1.12e+11  0.000 9.29e+04  9.29e+04
Solardeckel            1.383e+06  4.57e-06  3.02e+11  0.000 1.38e+06  1.38e+06
ar.L1                 -1.9267   0.342   -5.632  0.000 -2.597  -1.256
ar.L2                 -0.9003   0.532   -1.693  0.090 -1.942  0.142
ar.L3                 0.5805   0.503   1.154  0.248 -0.405  1.566
ar.L4                 0.6342   0.412   1.538  0.124 -0.174  1.442
ar.L5                 0.2850   0.254   1.122  0.262 -0.213  0.783
ar.L6                 0.1382   0.261   0.530  0.596 -0.373  0.649
ar.L7                 -0.3116   0.196   -1.589  0.112 -0.696  0.073
ar.L8                 -0.5469   0.167   -3.267  0.001 -0.875  -0.219
ar.L9                 -0.2600   0.077   -3.369  0.001 -0.411  -0.109
ma.L1                 0.7157   0.321   2.233  0.026  0.088  1.344
ma.L2                 -1.3541   0.264   -5.131  0.000 -1.871  -0.837
ma.L3                 -1.4514   0.411   -3.530  0.000 -2.257  -0.645
ma.L4                 0.4196   0.259   1.617  0.106 -0.089  0.928
ma.L5                 0.6709   0.340   1.975  0.048  0.005  1.337
ar.S.L12               -0.4833   0.721   -0.670  0.503 -1.897  0.931
ar.S.L24               -0.5043   0.677   -0.745  0.456 -1.832  0.823
ma.S.L12               0.7299   0.722   1.011  0.312 -0.685  2.145
ma.S.L24               0.6598   0.730   0.903  0.366 -0.772  2.091
ma.S.L36               0.2066   0.239   0.864  0.388 -0.262  0.675
ma.S.L48               0.1455   0.320   0.454  0.650 -0.482  0.773
ma.S.L60               0.1972   0.292   0.675  0.500 -0.375  0.770
ma.S.L72               0.2245   0.220   1.019  0.308 -0.207  0.656
ma.S.L84               0.2333   0.094   2.478  0.013  0.049  0.418
sigma2                4.594e+10  2.08e-11  2.2e+21  0.000  4.59e+10  4.59e+10
=====
Ljung-Box (L1) (Q):      0.01  Jarque-Bera (JB):        4755.73
Prob(Q):                  0.90  Prob(JB):           0.00
Heteroskedasticity (H):  14.86  Skew:              -1.17
Prob(H) (two-sided):     0.00  Kurtosis:           23.35
=====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 8.7e+36. Standard errors may be unstable.

```

Tabelle 16: Summary Sarimax Modell 1.

Variablen:

```
target_train = train_data['GesamtBruttoleistung']
```

```
exog_train = train_data[['Anzahl_Volleinspeisung', 'Anzahl_Teileinspeisung',
'Einspeiseverguetung', 'Strompreis', 'PV_Kosten_Euro_pro_kWp', 'Fukushima',
'Ukrainekrieg', 'Solardeckel']]
```

```
SARIMAX Results
=====
Dep. Variable: GesamtBruttoleistung No. Observations: 274
Model: SARIMAX(7, 2, 3)x(3, 0, [1, 2, 3, 4, 5], 12) Log Likelihood: -3985.602
Date: Sat, 09 Mar 2024 AIC: 8021.205
Time: 10:28:33 BIC: 8111.350
Sample: 01-01-2000 HQIC: 8057.395
- 10-01-2022
Covariance Type: opg
=====
            coef    std err      z   P>|z|      [0.025]     [0.975]
-----
Einspeiseverguetung  8.827e+06  3.35e-08  2.64e+14  0.000   8.83e+06  8.83e+06
Strompreis           5.663e+05  1.82e-09  3.12e+14  0.000   5.66e+05  5.66e+05
PV_Kosten_Euro_pro_kWp 6.39e+06  1.56e-08  4.1e+14  0.000   6.39e+06  6.39e+06
Fukushima            1.871e+06  7.52e-08  2.49e+13  0.000   1.87e+06  1.87e+06
Ukrainekrieg         2.945e+04  4.03e-09  7.3e+12  0.000   2.94e+04  2.94e+04
Solardeckel           1.558e+06  1.67e-07  9.34e+12  0.000   1.56e+06  1.56e+06
ar.L1                -1.0195   0.088   -11.633  0.000   -1.191   -0.848
ar.L2                -0.2055   0.074   -2.771  0.006   -0.351   -0.060
ar.L3                0.3192   0.053   6.071  0.000   0.216   0.422
ar.L4                -0.1103   0.050   -2.222  0.026   -0.208   -0.013
ar.L5                -0.1124   0.061   -1.848  0.065   -0.232   0.007
ar.L6                -0.0608   0.050   -1.223  0.221   -0.158   0.037
ar.L7                -0.0260   0.020   -1.299  0.194   -0.065   0.013
ma.L1                0.4959   0.090   5.536  0.000   0.320   0.671
ma.L2                -0.6393   0.065   -9.833  0.000   -0.767   -0.512
ma.L3                -0.8535   0.071   -11.973 0.000   -0.993   -0.714
ar.S.L12              -0.0210   0.200   -0.105  0.916   -0.412   0.370
ar.S.L24              -0.0473   0.186   -0.255  0.799   -0.411   0.316
ar.S.L36              -0.8806   0.166   -5.301  0.000   -1.206   -0.555
ma.S.L12              0.0547   0.205   0.267  0.789   -0.346   0.456
ma.S.L24              0.0424   0.191   0.222  0.824   -0.332   0.416
ma.S.L36              0.8372   0.198   4.237  0.000   0.450   1.224
ma.S.L48              0.0553   0.036   1.559  0.119   -0.014   0.125
ma.S.L60              0.0069   0.032   0.218  0.828   -0.055   0.069
sigma2               1.789e+11  1.36e-12  1.31e+23 0.000   1.79e+11  1.79e+11
=====
Ljung-Box (L1) (Q): 28.84 Jarque-Bera (JB): 5485.86
Prob(Q): 0.00 Prob(JB): 0.00
Heteroskedasticity (H): 0.47 Skew: -1.47
Prob(H) (two-sided): 0.00 Kurtosis: 24.80
=====
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 6.45e+37. Standard errors may be unstable.
```

Tabelle 17: Summary Sarimax Modell 2.

Variablen:

```
target_train_2 = train_data['GesamtBruttoleistung']

exog_train_2      =      train_data[['Einspeiseverguetung',      'Strompreis',
'PV_Kosten_Euro_pro_kWp', 'Fukushima', 'Ukrainekrieg', 'Solardeckel']]
```

```

SARIMAX Results
=====
Dep. Variable: GesamtBruttoleistung No. Observations: 274
Model: SARIMAX(7, 2, 3)x(3, 0, [1, 2, 3, 4, 5, 6], 12) Log Likelihood -4006.363
Date: Sat, 09 Mar 2024 AIC 8060.726
Time: 10:28:53 BIC 8147.265
Sample: 01-01-2000 HQIC 8095.468
- 10-01-2022
Covariance Type: opg
=====
            coef    std err      z   P>|z|   [0.025   0.975]
-----
Einspeiseverguetung  9.877e+06  5.22e-08  1.89e+14  0.000  9.88e+06  9.88e+06
Strompreis           7.85e+05  9.46e-09  8.3e+13  0.000  7.85e+05  7.85e+05
PV_Kosten_Euro_pro_kWp 7.78e+06  3.43e-08  2.27e+14  0.000  7.78e+06  7.78e+06
Solardeckel          1.559e+06  1.8e-07  8.68e+12  0.000  1.56e+06  1.56e+06
ar.L1                -0.3829  0.237   -1.619   0.106  -0.847   0.081
ar.L2                -0.3124  0.102   -3.068   0.002  -0.512   -0.113
ar.L3                0.3266  0.072   4.561   0.000  0.186   0.467
ar.L4                -0.0687  0.037   -1.864   0.062  -0.141   0.004
ar.L5                -0.0398  0.037   -1.079   0.281  -0.112   0.033
ar.L6                0.0044  0.035   0.126   0.900  -0.063   0.072
ar.L7                0.0009  0.025   0.037   0.971  -0.048   0.050
ma.L1                -0.1106  0.235   -0.472   0.637  -0.571   0.349
ma.L2                -0.1635  0.154   -1.060   0.289  -0.466   0.139
ma.L3                -0.7014  0.147   -4.765   0.000  -0.990  -0.413
ar.S.L12              0.5214  0.136   3.842   0.000  0.255   0.787
ar.S.L24              0.5600  0.172   3.249   0.001  0.222   0.898
ar.S.L36              -0.8398  0.086   -9.758   0.000  -1.008  -0.671
ma.S.L12              -0.5811  0.146   -3.974   0.000  -0.868  -0.295
ma.S.L24              -0.5644  0.183   -3.092   0.002  -0.922  -0.207
ma.S.L36              0.8601  0.103   8.350   0.000  0.658   1.062
ma.S.L48              0.0745  0.042   1.794   0.073  -0.007   0.156
ma.S.L60              -0.0406  0.045   -0.893   0.372  -0.130   0.048
ma.S.L72              -0.0764  0.057   -1.350   0.177  -0.187   0.034
sigma2               1.792e+11  6.98e-12  2.57e+22  0.000  1.79e+11  1.79e+11
=====
Ljung-Box (L1) (Q): 33.64 Jarque-Bera (JB): 5325.19
Prob(Q): 0.00 Prob(JB): 0.00
Heteroskedasticity (H): 0.54 Skew: -1.70
Prob(H) (two-sided): 0.00 Kurtosis: 24.41
=====
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 9.12e+37. Standard errors may be unstable.

```

**Tabelle 18:** Summary Sarimax Modell 3.

Variablen:

```

target_train_3 = train_data['GesamtBruttoleistung']

exog_train_3      =      train_data[['Einspeiseverguetung',      'Strompreis',
'PV_Kosten_Euro_pro_kWp', 'Solardeckel']]

```

```

SARIMAX Results
=====
Dep. Variable: GesamtBruttoleistung No. Observations: 274
Model: SARIMAX(7, 2, 3)x(3, 0, [1, 2, 3, 4, 5, 6], 12) Log Likelihood -4027.142
Date: Sat, 09 Mar 2024 AIC 8100.285
Time: 10:29:14 BIC 8183.218
Sample: 01-01-2000 HQIC 8133.580
- 10-01-2022
Covariance Type: opg
=====
              coef    std err      z   P>|z|   [0.025   0.975]
-----
Einspeiseverguetung  9.875e+06  3.22e-08  3.07e+14  0.000  9.88e+06  9.88e+06
Strompreis            7.847e+05  7.39e-09  1.06e+14  0.000  7.85e+05  7.85e+05
PV_Kosten_Euro_pro_kWp 7.78e+06  3.34e-08  2.33e+14  0.000  7.78e+06  7.78e+06
ar.L1                -1.1210   0.070   -16.034  0.000  -1.258   -0.984
ar.L2                -0.1852   0.055   -3.374  0.001  -0.293   -0.078
ar.L3                0.3726   0.042    8.959  0.000  0.291    0.454
ar.L4                -0.0838   0.041   -2.024  0.043  -0.165   -0.003
ar.L5                -0.1119   0.059   -1.901  0.057  -0.227   0.003
ar.L6                -0.0620   0.048   -1.304  0.192  -0.155   0.031
ar.L7                0.0002   0.017    0.010  0.992  -0.034   0.034
ma.L1                0.6053   0.077    7.827  0.000  0.454    0.757
ma.L2                -0.6954   0.063   -10.983 0.000  -0.819   -0.571
ma.L3                -0.9090   0.056   -16.196 0.000  -1.019   -0.799
ar.S.L12              0.5810   0.179    3.248  0.001  0.230    0.932
ar.S.L24              0.5738   0.207    2.776  0.006  0.169    0.979
ar.S.L36              -0.8752   0.115   -7.585  0.000  -1.101   -0.649
ma.S.L12              -0.5705   0.192   -2.974  0.003  -0.947   -0.195
ma.S.L24              -0.5893   0.211   -2.797  0.005  -1.002   -0.176
ma.S.L36              0.8421   0.132    6.378  0.000  0.583    1.101
ma.S.L48              0.1048   0.043    2.421  0.015  0.020    0.190
ma.S.L60              -0.0218   0.054   -0.405  0.686  -0.127   0.084
ma.S.L72              -0.0655   0.060   -1.099  0.272  -0.182   0.051
sigma2               1.69e+11  8.35e-13  2.02e+23  0.000  1.69e+11  1.69e+11
=====
Ljung-Box (L1) (Q): 36.48  Jarque-Bera (JB): 4883.80
Prob(Q):           0.00  Prob(JB):          0.00
Heteroskedasticity (H): 0.30  Skew:           -1.70
Prob(H) (two-sided): 0.00  Kurtosis:         23.48
=====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 9.87e+38. Standard errors may be unstable.

```

**Tabelle 19:** Summary Sarimax Modell 4.

Variablen:

```

target_train_4 = train_data['GesamtBruttoleistung']

exog_train_4 = train_data[['Einspeiseverguetung', 'Strompreis', 'PV_Kosten_Euro_pro_kWp']]

```

```

OLS Regression Results
=====
Dep. Variable: GesamtBruttoleistung R-squared:      0.437
Model:           OLS   Adj. R-squared:      0.429
Method:          Least Squares F-statistic:       55.01
Date: Fri, 12 Apr 2024 Prob (F-statistic): 2.82e-34
Time: 19:38:58 Log-Likelihood:     -4084.4
No. Observations: 288 AIC:             8179.
Df Residuals:    283 BIC:            8197.
Df Model:        4
Covariance Type: nonrobust
=====
              coef    std err      t      P>|t|      [0.025      0.975]
-----
Intercept      -1.308e+06  3.03e+05   -4.317      0.000    -1.9e+06   -7.12e+05
Strompreis      5.573e+04  7304.990    7.629      0.000     4.13e+04   7.01e+04
Einspeiseverguetung 8107.7092  3604.514    2.249      0.025    1012.649   1.52e+04
Fukushima       4.658e+05  1.17e+05    3.993      0.000     2.36e+05   6.95e+05
Solardeckel     -1.724e+05  7.14e+04   -2.416      0.016    -3.13e+05  -3.19e+04
=====
Omnibus:          337.050 Durbin-Watson:      1.832
Prob(Omnibus):    0.000 Jarque-Bera (JB): 15931.643
Skew:             5.199 Prob(JB):         0.00
Kurtosis:         37.922 Cond. No.        597.
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

**Tabelle 20:** Ergebnisse der OLS-Regression (GesamtBruttoleistung ~ Strompreis + Einspeiseverguetung + Fukushima + Solardeckel)

## 6.2. ChatGPT-Protokoll

**Spalten des df pv\_raw\_m:** EinheitMastrNummer DatumLetzteAktualisierung LokationMaStRNummer NetzbetreiberpruefungStatus AnlagenbetreiberMastrNummer Land Bundesland Landkreis Gemeinde Gemeindeschluessel ... InAnspruchGenommeneAckerflaeche GeplantesInbetriebnahmedatum GenMastrNummer Einsatzverantwortlicher WeicDisplayName DatumBeginnVoruebergehendeStilllegung DatumWiederaufnahmeBetrieb Buergerenergie Kraftwerksnummer Weic Ich möchte fehlende Werte in der Spalte AnzahlModule ersetzen mit der durchschnittlichen Anzahl Module für andere Anlagen mit der gleichen Bruttoleistung

```
import pandas as pd
```

```
# Annahme: Sie haben einen DataFrame pv_raw_m
```

```
# Beispiel DataFrame erstellen (ersetzen Sie dies durch Ihren tatsächlichen DataFrame) pv_raw_m = pd.DataFrame({  
    'EinheitMastrNummer': [1, 2, 3, 4, 5],  
    'Bruttoleistung': [100, 200, 100, 200, 300],  
    'AnzahlModule': [50, 100, None, 150, None]  
})
```

```
# Fehlende Werte in "AnzahlModule" durch den Durchschnitt der Anzahl Module für jede Bruttoleistung ersetzen pv_raw_m['AnzahlModule']
```

```
=      pv_raw_m.groupby('Bruttoleistung')['AnzahlModule'].trans-
form(lambda x: x.fillna(x.mean()))
```

```
# Ergebnisse anzeigen
```

```
print(pv_raw_m)
```

**Ersetzte nicht mit dem Durchschnitt, sondern berechne einen möglichst genauen Wert, indem du: zunächst anhand Inbetriebnahmedatum das Jahr der der Inbetriebnahme berechnest, denn die durchschnittliche Bruttoleistung pro Modul in diesem Jahr berechnest und zuletzt aus der gegebenen Bruttoleistung die Fehlende AnzahlModule berechnest (gerundet auf ganze Zahl).**

```
import pandas as pd
```

```
# Beispiel DataFrame erstellen (ersetzen Sie dies durch Ihren tatsächlichen DataFrame) pv_raw_m = pd.DataFrame({
    'EinheitMastrNummer': [1, 2, 3, 4, 5],
    'Bruttoleistung': [100, 200, 100, 200, 300],
    'Inbetriebnahmedatum': ['2021-01-01', '2021-01-01', '2022-01-01',
                           '2022-01-01', '2023-01-01'],
    'AnzahlModule': [50, 100, None, 150, None]
})
```

```
# Konvertieren Sie das Inbetriebnahmedatum in ein DateTime-Objekt  
pv_raw_m['Inbetriebnahmedatum'] = pd.to_datetime(pv_raw_m['Inbetriebnahmedatum'])
```

```
# Extrahieren Sie das Jahr der Inbetriebnahme  
pv_raw_m['Inbetriebsjahr'] = pv_raw_m['Inbetriebnahmedatum'].dt.year
```

```
# Berechnen Sie die durchschnittliche Bruttolleistung pro Modul für jedes  
Inbetriebsjahr average_power_per_module =  
pv_raw_m.groupby('Inbetriebsjahr')['Bruttolleistung'].mean() /  
pv_raw_m.groupby('Inbetriebsjahr')['AnzahlModule'].mean()
```

```
# Extrapolieren Sie die fehlenden Werte für AnzahlModule basierend auf  
der gegebenen Bruttolleistung
```

```
pv_raw_m['AnzahlModule'] = pv_raw_m.apply(lambda row:  
round(row['Bruttolleistung'] / average_power_per_module[row['Inbetriebsjahr']]), axis=1)
```

```
# Ergebnisse anzeigen
```

```
print(pv_raw_m)
```

**Beispieltabelle: Gebietskennziffer Ort BIP 0 1001000 Flensburg, Stadt 63.55 1  
1002000 Kiel, Stadt 68.26 füge bei der Gebietskennziffer eine führende 0 an,  
wenn die Länge 7 stellig ist. Dann schreibe die ersten 5 ziffern in eine neue  
Spalte 'AGS-5'**

```
import pandas as pd
```

```
# Beispiel-Daten-
satz data = {
    'Gebietskennziffer': [1001000, 1002000],
    'Ort': ['Flensburg, Stadt', 'Kiel, Stadt'],
    'BIP': [63.55, 68.26]
}
```

```
df = pd.DataFrame(data)
```

```
# Überprüfen und Hinzufügen der führenden Null
df['Gebietskennziffer'] = df['Gebietskennziffer'].astype(str).apply(lambda x: '0' + x if
len(x) == 7 else x)
```

```
# Erstellen der neuen Spalte 'AGS-5'
df['AGS-5'] = df['Gebietskennziffer'].str[:5]
```

```
# Anzeigen des aktualisierten DataFrames
print(df) ein
```

## Wie kann ich fehlende Werte mit möglichst ähnlichen Zeilen auffüllen?

Das Auffüllen von fehlenden Werten in einem DataFrame mit ähnlichen Zeilen kann eine sinnvolle Strategie sein, um die Integrität der Daten zu bewahren. Hier sind einige Schritte, wie du dies in Python mit Pandas tun kannst:

Angenommen, du hast ein DataFrame namens df, und du möchtest die fehlenden Werte in diesem DataFrame basierend auf ähnlichen Zeilen auffüllen:

Verwende die Methode `fillna()` von Pandas, um die fehlenden Werte basierend auf ähnlichen Zeilen aufzufüllen. Du kannst die Methode `fillna()` zusammen mit der Methode `fillna(method='ffill')` oder `fillna(method='bfill')` verwenden, um vorwärts (forward-fill) oder rückwärts (backward-fill) aufzufüllen. Welche Methode besser ist, hängt von deinen Daten und Anforderungen ab.

Vorwärts-Auffüllen: Fehlende Werte werden mit dem nächsten verfügbaren Wert in derselben Spalte aufgefüllt.

```
py-
thon
Copy
code
df_filled = df.fillna(method='ffill')
```

Rückwärts-Auffüllen: Fehlende Werte werden mit dem vorherigen verfügbaren Wert in derselben Spalte aufgefüllt.

```
py-
thon
```

```
Cop  
y  
code  
df_filled = df.fillna(method='bfill')
```

Du kannst auch eine Kombination aus Vorwärts- und Rückwärts-Auffüllen verwenden, um eine bessere Interpolation zu erzielen:

python

Copy code

```
df_filled = df.fillna(method='ffill').fillna(method='bfill')
```

Durch diese Schritte werden die fehlenden Werte in deinem DataFrame basierend auf den Werten in ähnlichen Zeilen aufgefüllt. Beachte jedoch, dass die Definition von "ähnlichen Zeilen" von deinen Daten und deinem Anwendungsfall abhängt. In einigen Fällen kann es sinnvoll sein, benutzerdefinierte Logik oder Machine-Learning-basierte Modelle zu verwenden, um fehlende Werte aufzufüllen, insbesondere wenn die Daten komplexer sind.

**Beispieltabelle migration: Status AGS-8 Deutsche mit beidseitigem Migrationshintergrund Deutsche mit einseitigem Migrationshintergrund Ausländer/-innen mit eigener Migrationserfahrung Ausländer/-innen ohne eigene Migrationserfahrung Deutsche mit eigener Migrationserfahrung Deutsche ohne eigene Migrationserfahrung Ausländer/-innen Deutsche mit Migrationshintergrund Insgesamt Personen mit Migrationshintergrund Personen ohne Migrationshintergrund 0 01001000 1730.0 1860.0 4290.0 820.0 4340.0 3600.0 5110.0 7940.0 81780.0 13050.0 68730.0 1 01002000 7230.0 5960.0 13100.0 2850.0 15200.0 13190.0 15950.0 28380.0 234550.0 44330.0**

**190220.0 2 01003000 4300.0 5140.0 10110.0 2630.0 12780.0 9450.0  
12740.0 22230.0 208000.0 34970.0 173030.0 User Berechne die pro-  
zentuale Verteilung auf Basis der Spalte 'Insgesamt' (außer AGS-8)  
und füge die Ergebnisse als neue Spalten mit Anhang\_Prozent in die  
Tabelle migration ein.**

```
import pandas as pd

# Annahme: Sie haben bereits ein DataFrame namens 'migration' mit den
entsprechenden Spalten

# Hier wird ein Beispiel-DataFrame erstellt: data = {
    'AGS-8': ['01001000', '01002000', '01003000'],
    'Insgesamt': [81780.0, 234550.0, 208000.0],
    'Deutsche mit beidseitigem Migrationshintergrund': [1730.0, 7230.0,
4300.0],
    'Deutsche mit einseitigem Migrationshintergrund': [1860.0, 5960.0,
5140.0],
    # Weitere Spalten hier einfügen...
}

migration = pd.DataFrame(data)

# Berechne die prozentuale Verteilung für jede Zeile
ags_8_stellig = migration['AGS-8']
migration_prozent = migration.drop(columns=['AGS-8'])
migration_prozent = migration_prozent.div(migration_prozent['Insgesamt'], axis=0) *
100
migration_prozent.columns = [col + "_Prozent" for col in migration_prozent.columns]
migration_prozent['AGS-8'] = ags_8_stellig
```

```
# Anzeige des aktualisierten DataFrames print(migration_prozent)

sample_df.groupby("Bevölkerung pro km2")["Anzahl_PV_Anlagen"].mean() ich möchte die ausgabe plotten
```

Um die Ausgabe der Gruppierung und Berechnung des Durchschnitts in einem Plot darzustellen, kannst du die matplotlib-Bibliothek in Kombination mit pandas verwenden.

Hier ist ein Beispiel, wie du das tun kannst:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Beispiel DataFrame (ersetze dies durch dein eigenes DataFrame) data = {
    'Bevölkerung pro km2': [100, 150, 200, 100, 150],
    'Anzahl_PV_Anlagen': [10, 12, 8, 15, 10]
}

sample_df = pd.DataFrame(data)

# Gruppiere nach "Bevölkerung pro km2" und berechne den Durchschnitt für "Anzahl_PV_Anlagen" grouped_data = sample_df.groupby("Bevölkerung pro km2")["Anzahl_PV_Anlagen"].mean()

# Plot plt.figure(figsize=(8, 6)) grouped_data.plot(kind='bar', color='skyblue') plt.xlabel('Bevölkerung pro km2') plt.ylabel('Durchschnittliche Anzahl PV-Anlagen')
```

```
plt.title('Durchschnittliche Anzahl PV-Anlagen pro Bevölkerung pro km2') plt.show()
```

Die Werte für Bevölkerung pro km2 liegen zwischen 35 und 4788. Schreibe den beigefügten Code so um, dass 12 Bins entstehen in jeweils 400er Schritten, also 0-400, 401-800, 801-1200 usw. Sortiere aufsteigend. bins = pd.cut(sample\_df['Bevölkerung pro km2'], bins=15, labels=False) # Definiere die Bin-Grenzen und die Reihenfolge für die Bins bin\_labels = [f'{int(sample\_df["Bevölkerung pro km2"].groupby(bins).min())[i]}-{int(sample\_df["Bevölkerung pro km2"].groupby(bins).max())[i]}' for i in range(15)] bin\_labels = bin\_labels[::-1] # Invertiere die Reihenfolge der Beschriftungen # Weise die Bin-Grenzen und die Reihenfolge zu bins = pd.cut(sample\_df['Bevölkerung pro km2'], bins=15, labels=False, right=False) sample\_df['Bevölkerung pro km2 Bins'] = bins bev\_dichte = sample\_df.groupby('Bevölkerung pro km2 Bins')['Anzahl\_PV\_Anlagen'].mean() bev\_dichte = bev\_dichte.sort\_index(ascending=False) plt.figure(figsize=(8, 6)) bev\_dichte.plot(kind='barh', color='lightseagreen') plt.xlabel('Durchschnittliche Anzahl PV-Anlagen pro Gemeinde', fontsize=12) plt.ylabel('Bevölkerung pro km2', fontsize=12) plt.title('Durchschnittliche Anzahl PV-Anlagen nach Einwohner / km2', fontsize=14) # Verwende die definierten Bin-Grenzen und Reihenfolge für die Beschriftungen der Y-Achse plt.yticks(range(len(bev\_dichte)), bin\_labels) plt.show()

Sicher, hier ist der umgeschriebene Code, um 12 Bins in 400er Schritten von 0 bis

4788 zu erstellen und die Bins aufsteigend zu sortieren:

```

import numpy as np

# Annahme: Sie haben ein DataFrame namens 'sample_df'

# Verwenden Sie den tatsächlichen Namen Ihres DataFrames, falls dieser
anders ist.

# Definiere die Bin-Grenzen bin_edges = np.arange(0, 4800, 400)
#   Bins   in 400er Schritten von 0 bis 4788 bins =
pd.cut(sample_df['Bevölkerung pro km2'], bins=bin_edges, la-
bels=False)

# Definiere die Bin-Grenzen und die Reihenfolge für die Bins
bin_labels = [f'{i}-{i+400} Einwohner pro km2' for i in range(0,
4800, 400)] bin_labels = bin_labels[::-1] # Invertiere die Reihen-
folge der Beschriftungen

# Weise die Bin-Grenzen und die Reihenfolge zu
sample_df['Bevölkerung pro km2 Bins'] = bins

bev_dichte = sample_df.groupby('Bevölkerung pro km2 Bins')['An-
zahl_PV_Anlagen'].mean()

bev_dichte = bev_dichte.sort_index(ascending=True) # Sortiere aufstei-
gend

plt.figure(figsize=(8, 6)) bev_dichte.plot(kind='barh', color='lightseag-
reen') plt.xlabel('Durchschnittliche Anzahl PV-Anlagen pro Gemeinde',
fontsize=12) plt.ylabel('Bevölkerung pro km2', fontsize=12)

```

```
plt.title('Durchschnittliche Anzahl PV-Anlagen nach Einwohner / km2',  
         fontsize=14)
```

```
# Verwende die definierten Bin-Grenzen und Reihenfolge für die Beschrif-  
tungen der Y-Achse plt.yticks(range(len(bev_dichte)), bin_labels)
```

```
plt.show()
```

**es werden nur 9 bins bis 3600 dargestellt.**

Es tut mir leid für das Missverständnis. Möglicherweise werden nicht alle Bins im Diagramm angezeigt, weil es nicht genügend Daten in diesen Bins gibt, um Balken zu erzeugen. Eine mögliche Lösung könnte darin bestehen, die Bins manuell anzugeben, um sicherzustellen, dass sie alle im Diagramm dargestellt werden: `bin_edges = [0, 400, 800, 1200, 1600, 2000, 2400, 2800, 3200, 3600, 4000, 4400, 4800]` `bins = pd.cut(sample_df['Bevölkerung pro km2'], bins=bin_edges, labels=False)`

```
# Rest deines Codes...
```

Durch explizites Festlegen der Bin-Grenzen kannst du sicherstellen, dass jeder Bin im Diagramm vertreten ist. Bitte passe die Grenzen entsprechend deiner Datenverteilung an, um die bestmögliche Darstellung zu erhalten.

```
bins = pd.cut(sample_df['Bevölkerung pro km2'], bins=15, la-  
bels=False) sample_df['Bevölkerung pro km2 Bins'] = bins  
bev_dichte = sample_df.groupby('Bevölkerung pro km2 Bins')['An-  
zahl_PV_Anlagen'].mean() bev_dichte = bev_dichte.sort_in-  
dex(ascending=False) plt.figure(figsize=(8, 6))
```

```

bev_dichte.plot(kind='barh', color='lightseagreen') plt.xlabel('Durchschnittliche Anzahl PV-Anlagen pro Gemeinde', fontsize=12) plt.ylabel('Bevölkerung pro km2', fontsize=12) plt.title('Durchschnittliche Anzahl PV-Anlagen nach Einwohner / km2', fontsize=14) yticks = [f'{int(sample_df["Bevölkerung pro km2"].groupby(bins).min()[i])}-{int(sample_df["Bevölkerung pro km2"].groupby(bins).max()[i])} Einwohner pro km2' for i in bev_dichte.index] yticks = yticks[::-1] # Invertiere die Reihenfolge der Beschriftungen plt.yticks(range(len(bev_dichte)), yticks) plt.show() die bins werden im Diagramm falsch dargestellt. warum?

```

Es sieht so aus, als ob das Problem mit der Reihenfolge der Bins im Diagramm zusammenhängt. Sie haben die Beschriftungen der Y-Achse umgekehrt (`yticks[::-1]`), aber die Bins selbst könnten möglicherweise noch in aufsteigender Reihenfolge sortiert sein.

Sie können sicherstellen, dass die Bins in der gewünschten Reihenfolge sind, indem Sie die `order`-Parameter in der `cut`-Funktion verwenden. Hier ist ein Beispiel:

```

# Annahme: Sie haben ein DataFrame namens 'sample_df'

# Verwenden Sie den tatsächlichen Namen Ihres DataFrames, falls dieser anders ist.

# Definiere die Bin-Grenzen bins = pd.cut(sample_df['Bevölkerung pro km2'], bins=15, labels=False)

# Definiere die Bin-Grenzen und die Reihenfolge für die Bins bin_labels =
[f'{int(sample_df["Bevölkerung

```

```

km2"].groupby(bins).min()[i])}{int(sample_df["Bevölkerung pro km2"].groupby(bins).max()[i])} Einwohner pro km2' for i in range(15)]
bin_labels = bin_labels[::-1] # Invertiere die Reihenfolge der Beschriftungen

# Weise die Bin-Grenzen und die Reihenfolge zu bins =
pd.cut(sample_df['Bevölkerung pro km2'], bins=15, labels=False, right=False) sample_df['Bevölkerung pro km2 Bins'] = bins

bev_dichte = sample_df.groupby('Bevölkerung pro km2 Bins')['Anzahl_PV_Anlagen'].mean() bev_dichte = bev_dichte.sort_index(ascending=False)

plt.figure(figsize=(8, 6)) bev_dichte.plot(kind='barh', color='lightseagreen') plt.xlabel('Durchschnittliche Anzahl PV-Anlagen pro Gemeinde', fontsize=12) plt.ylabel('Bevölkerung pro km2', fontsize=12) plt.title('Durchschnittliche Anzahl PV-Anlagen nach Einwohner / km2', fontsize=14)

# Verwende die definierten Bin-Grenzen und Reihenfolge für die Beschriftungen der YAchse plt.yticks(range(len(bev_dichte)), bin_labels)

plt.show()

```

Durch das Festlegen der Bin-Grenzen und der Reihenfolge für die Bins sollten die

Bins im Diagramm in der gewünschten Reihenfolge angezeigt werden.

**query\_19 = """ SELECT \* FROM marktstammdaten LEFT JOIN einspeiseverguetung ON marktstammdaten.[Registrierungsdatum] = einspeiseverguetung.[Datum] LIMIT 5 """ Für marktstammdaten.[Registrierungsdatum] liegt ein konkretes Datum vor. Für einspeiseverguetung.[Datum] liegen jeweils Datumsangaben vor, die bis zum Datum der nächsten Zeile gelten - manchmal ein Jahr, manchmal ein Monat**

In diesem Fall, wo die Daten in einspeiseverguetung.[Datum] bis zum Datum der nächsten Zeile gelten und unterschiedliche Zeitintervalle haben können (manchmal ein Jahr, manchmal ein Monat), wird der Join etwas komplexer. Um das korrekt umzusetzen, kannst du die Bedingung für den Join so anpassen, dass sie die entsprechende nächste Zeile in einspeiseverguetung findet, deren Datum größer oder gleich dem Datum in marktstammdaten ist.

Eine Möglichkeit, dies in SQL zu erreichen, ist die Verwendung einer Unterabfrage oder einer Window Function, um das nächste Datum in einspeiseverguetung zu finden, das größer oder gleich dem Datum in marktstammdaten ist. Leider kann SQL allein manchmal Schwierigkeiten haben, dies elegant zu lösen, insbesondere wenn die Datenbank keinen direkten Zugriff auf den Index oder die Zeilennummerierung bietet.

Hier ist eine allgemeine Herangehensweise, die mit einer Unterabfrage arbeitet:

```
SELECT *
FROM marktstammdaten
LEFT JOIN (
    SELECT *,
```

```

        LEAD([Datum]) OVER (ORDER BY [Datum]) AS Next_Date
      FROM einspeiseverguetung
    ) AS evg
    ON marktstammdaten.[Registrierungsdatum] >= evg.[Datum]
    AND (marktstammdaten.[Registrierungsdatum] < evg.Next_Date OR
    evg.Next_Date IS
    NULL)
  LIMIT 5

```

In dieser Abfrage wird eine Unterabfrage verwendet, um eine zusätzliche Spalte Next\_Date zu erstellen, die das nächste Datum in der Spalte einspeiseverguetung.[Datum] enthält. Dann wird der Join so konfiguriert, dass er die Zeilen verknüpft, wenn das marktstammdaten.[Registrierungsdatum] größer oder gleich dem einspeiseverguetung.[Datum] ist und kleiner als das nächste Datum in einspeiseverguetung, oder wenn es kein nächstes Datum gibt.

Bitte beachte, dass dies ein allgemeiner Ansatz ist und je nach spezifischer Struktur der Datenbank und der Anforderungen an die Join-Bedingungen Anpassungen erforderlich sein könnten.

Datum	Einspeiseverguetung_kleine_PV_Anlagen	0	2005-01-01	54.53								
1	200601-01	51.80	2	2007-01-01	49.21	3	2008-01-01	46.75	4	2009-01-01	43.01	...
...	...	139	2023-03-01	7.65	140	2023-04-01	7.65	141	2023-05-01	7.65	142	
		2023-06-01	7.65	143	2023-07-01	7.65	144	rows × 2 columns	Um die			

**Tabelle später besser Joinen zu können, möchte ich die fehlenden Daten ergänzen, so dass für jedes Datum zwischen 01.01.2005 und 01.07.2023 eine Zeile generiert wird. Die angegebene Einspeisevergütung\_kleine\_PV\_Anlagen gilt jeweils bis zum nächsten Datum. Ab da gilt die neue Angabe.**

Um eine vollständige Zeitreihe mit allen Daten zwischen dem 01.01.2005 und dem 01.07.2023 zu erstellen und die Einspeisevergütung bis zum nächsten Datum beizubehalten, kannst du die fehlenden Daten auffüllen und die fehlenden Werte interpolieren. Hier ist eine Möglichkeit, das mit Python und Pandas zu erreichen:

Angenommen, dein DataFrame heißt dataframe:

```
import pandas as pd

# Erstelle ein Date Range-Objekt mit allen Daten zwischen dem ersten und letzten Datum in deinem DataFrame date_range = pd.date_range(start='2005-01-01', end='2023-07-01', freq='D')

# Setze das Datum als Index
dataframe = dataframe.set_index('Datum')

# Erstelle eine neue Spalte mit dem Datum-Index und fülle die fehlenden Werte durch Vorwärtstfüllen (forward fill)
filled_data = dataframe.reindex(date_range).ffill() # Setze das Datum zurück in eine Spalte
filled_data['Datum'] = filled_data.index

# Zurücksetzen des Indexes, um das DataFrame zu vervollständigen
filled_data.reset_index(drop=True, inplace=True)

# Ergebnis mit allen Daten und beibehaltener Einspeisevergütung bis zum nächsten Datum

print(filled_data)
```

Das Skript erstellt eine vollständige Zeitreihe mit allen Daten zwischen dem frühesten und dem spätesten Datum in deinem DataFrame und füllt die fehlenden Werte in der Spalte Einspeisevergütung\_kleine\_PV\_Anlagen durch Vorwärtsfüllen auf, was bedeutet, dass der letzte bekannte Wert bis zum nächsten bekannten Datum übernommen wird.

```
Jahr_Monat Anzahl_Volleinspeisung Anzahl_Teileinspeisung Einspeisevergütung Strompreis 0 2005-01 2154 37 54.53 31.81 1 2005-02 1246 18 54.53 30.46 2 2005-03 3427 38 54.53 30.46 3 2005-04 5842 76 54.53 30.46 4 2005-05 5420 78 54.53 30.46 ... ... ... ... 217 2023-02 1848 66384 7.65 46.27 218 2023-03 2859 88120 7.65 30.46 219 2023-04 3240 82478 7.65 46.27 220 2023-05 3694 92547 7.65 46.27 221 2023-06 3624 81258 7.65 31.81 222 rows x 5 columns Ich möchte diese Tabelle plotten um den zeitlichen Verlauf darstellen, vorschäge?
```

Das klingt nach einer großartigen Visualisierungsmöglichkeit! Ein zeitlicher Verlauf könnte mittels eines Liniendiagramms oder eines Bereichsdigramms dargestellt werden, um die Veränderungen im Laufe der Zeit für die verschiedenen Daten darzustellen. Hier ist ein Beispiel, wie du diese Daten mit Python und der Bibliothek Matplotlib visualisieren könntest:

```
import matplotlib.pyplot as plt

# Hier sind einige Beispieldaten als DataFrame
dargestellt # Ersetze diese Daten mit deinem DataFrame, z. B. abfrage_41 data = {
    'Jahr_Monat': ['2005-01', '2005-02', '2005-03', '2005-04', '2005-05',
    '2023-02', '2023-03', '2023-04', '2023-05', '2023-06'],
    'Anzahl_Volleinspeisung': [2154, 1246, 3427, 5842, 5420, 1848, 2859,
    3240, 3694, 3624],
    'Anzahl_Teileinspeisung': [37, 18, 38, 76, 78, 66384, 88120, 82478,
    92547, 81258],
```

```

'Einspeisevergütung': [54.53, 54.53, 54.53, 54.53, 54.53, 7.65, 7.65,
7.65, 7.65, 7.65],

'Strompreis': [31.81, 30.46, 30.46, 30.46, 30.46, 46.27, 30.46, 46.27,
46.27, 31.81] }

# DataFrame
erstellen df =
pd.Data-
Frame(data)

# Um die Zeitkomponente zu berücksichtigen, könnten wir das Jahr-Mo-
nat-Format in ein datetime-Format umwandeln df['Jahr_Monat'] =
pd.to_datetime(df['Jahr_Monat'] + '-01')

# Plot erstellen plt.fi-
gure(figsize=(10, 6))

# Schleife über die verschiedenen Spalten, um sie
zu plotten for column in df.columns[1:]:

    plt.plot(df['Jahr_Monat'], df[column], marker='o', la-
bel=column) plt.xlabel('Datum') plt.ylabel('Werte')
    plt.title('Zeitlicher Verlauf der Daten') plt.legend()
    plt.grid(True) plt.tight_layout()

# Plot anzeigen
plt.show()

Stelle sicher, dass du den entsprechenden DataFrame für die Visualisie-
rung verwendest, indem du die Daten aus deiner abfrage_41 verwendest
und sie in das oben gezeigte Skript einfügst. Dieses Skript verwendet Mat-
plotlib, um die verschiedenen Werte über die Zeit zu plotten.

```

**Das passt so nicht. Anzahl\_Volleinspeisung und Anzahl\_Teileinspeisung ergeben zusammen 100% und könnten als stacked barchart geplottet werden.**

**Einspeisevergütung Strompreis werden als Linien angezeigt**

Verstanden, du möchtest die Summe von **Anzahl\_Volleinspeisung** und **Anzahl\_Teileinspeisung** als gestapeltes Säulendiagramm darstellen und **Einspeisevergütung** sowie **Strompreis** als Linien auf dem gleichen Diagramm.

Hier ist ein Beispiel, wie du das mit Matplotlib umsetzen könntest:

```
import matplotlib.pyplot as plt

# Hier sind einige Beispieldaten als DataFrame
dargestellt # Ersetze diese Daten mit deinem Da-
taFrame, z. B. abfrage_41 data = {
    'Jahr_Monat': ['2005-01', '2005-02', '2005-03', '2005-04', '2005-05',
    '2023-02', '2023-
    03', '2023-04', '2023-05', '2023-06'],
    'Anzahl_Volleinspeisung': [2154, 1246, 3427, 5842, 5420, 1848, 2859,
    3240, 3694, 3624],
    'Anzahl_Teileinspeisung': [37, 18, 38, 76, 78, 66384, 88120, 82478,
    92547, 81258],
    'Einspeisevergütung': [54.53, 54.53, 54.53, 54.53, 54.53, 7.65, 7.65,
    7.65, 7.65, 7.65],
    'Strompreis': [31.81, 30.46, 30.46, 30.46, 30.46, 46.27, 30.46, 46.27,
    46.27, 31.81] }

# DataFrame
erstellen df =
pd.Data-
Frame(data)
```

```

# Um die Zeitkomponente zu berücksichtigen, könnten wir das Jahr-Monat-Format in ein datetime-Format umwandeln df['Jahr_Monat'] = pd.to_datetime(df['Jahr_Monat'] + '-01')

# Plot erstellen fig, ax1 =
plt.subplots(figsize=(10, 6))

# Säulen für Anzahl_Volleinspeisung und Anzahl_Teileinspeisung (als gestapeltes Säulendiagramm) ax1.bar(df['Jahr_Monat'], df['Anzahl_Volleinspeisung'], label='Anzahl_Volleinspeisung', width=20)
ax1.bar(df['Jahr_Monat'], df['Anzahl_Teileinspeisung'], label='Anzahl_Teileinspeisung', width=20, bottom=df['Anzahl_Volleinspeisung'])

# Linien für Einspeisevergütung und Strompreis ax2 = ax1.twinx()
ax2.plot(df['Jahr_Monat'], df['Einspeisevergütung'], color='red', label='Einspeisevergütung')
ax2.plot(df['Jahr_Monat'], df['Strompreis'], color='green', label='Strompreis')

# Achsenbeschriftungen ax1.set_xlabel('Datum') ax1.set_ylabel('Anzahl Einspeisungen') ax2.set_ylabel('Preis') # Legenden anzeigen
ax1.legend(loc='upper left') ax2.legend(loc='upper right') plt.title('Einspeisungen und Preise über die Zeit')
plt.tight_layout() # Plot anzeigen
plt.show()

```

Dieses Skript nutzt Matplotlib, um ein gestapeltes Säulendiagramm für Anzahl\_Volleinspeisung und Anzahl\_Teileinspeisung zu erstellen und Linien für Einspeisevergütung und Strompreis darzustellen. Bitte ersetze die Beispieldaten im DataFrame df mit deinen eigenen Daten aus der Abfrage.

```
# Anzahl der Folds für die Cross-Validierung n_splits = 5 random_state = 42 # Initialisierung KFold kfold = KFold(n_splits=n_splits, shuffle=True,
```

```

random_state=random_state) cv_scores = [] # Iterieren über die Folds for train_index, test_index in kfold.split(scaled_data): train_data, test_data =
scaled_data[train_index], scaled_data[test_index] # Daten in Sequenzen und Ziele umwandeln train_sequences, train_targets = prepare_data(train_data, n_steps)
test_sequences, test_targets = prepare_data(test_data, n_steps) # Modell erstellen und trainieren dropout_model = Sequential() dropout_model.add(LSTM(units=50, activation='relu', input_shape=(n_steps, 1))) dropout_model.add(Dropout(0.1)) # Dropout mit 10% Auslassung von Neuronen dropout_model.add(Dense(units=1)) dropout_model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])

drop-

```

out\_model.fit(train\_sequences, train\_targets, epochs=100, batch\_size=32, verbose=0) # Bewertung des Modells auf dem Testdatensatz scores = dropout\_model.evaluate(test\_sequences, test\_targets, verbose=0) print(f"Fold Score (MSE): {scores}") cv\_scores.append(scores) # Durchschnittliche Bewertung über alle Folds average\_score = np.mean(cv\_scores) print(f"Average Score (MSE): {average\_score}") Dieses modell hat am besten performt.

Ich möchte nun monatliche vorhersagen treffen vom 01.07.2023 bis 31.12.2030. als ausgabe möchte ich sowohl die vorhergesagten werte und konfidenzintervalle als tabelle als auch einen plot der daten, beginnend ab 2019.

Um monatliche Vorhersagen von Juli 2023 bis Dezember 2030 zu treffen und diese mit Konfidenzintervallen anzuzeigen, benötigst du eine Funktion, die kontinuierliche Vorhersagen erstellt und die Ergebnisse in einer Tabelle und einem Plot darstellt.

Hier ist ein allgemeines Beispiel, wie du das umsetzen könntest:

```
# Skalierung der gesamten Daten
scaled_data = scaler.fit_transform(df)
```

```
# Funktion zur Erstellung von Vorhersagen und Konfidenzintervallen def make_predictions(model, data, scaler, n_steps, start_date, end_date): # Initialisierung der Listen
```

```

für Vorhersagen und Konfidenzintervalle    predictions = []
lower_confidence = []    upper_confidence = []    dates =
pd.date_range(start=start_date, end=end_date, freq='M')

# Vorhersagen für jeden Monat
treffen    for date in dates:
    # Vorbereiten der Eingangsdaten für die Vorhersage
    data_for_prediction = data[-n_steps:] # Die letzten n_steps Datenpunkte
    prepared_data = np.array(data_for_prediction).reshape(1, n_steps, 1)

    # Vorhersage treffen und inverse Skalierung anwenden
    prediction = model.predict(prepared_data)
    prediction = scaler.inverse_transform(prediction)[0][0]

# Konfidenzintervall berechnen (hier als Platzhalter)
    lower = prediction - 0.1 * prediction
    upper = prediction + 0.1 * prediction

    # Daten für die nächste Vorhersage aktualisieren
    data = np.append(data, prediction)
    predictions.append(prediction)
    lower_confidence.append(lower)
    upper_confidence.append(upper)

return predictions, lower_confidence, upper_confidence, dates

```

```

# Festlegen des Startdatums für die Vorher-
sage start_date = '2023-07-01'

# Vorhersagen und Konfidenzintervalle erstellen predictions, lower_confi-
dence, upper_confidence, dates = make_predictions(dropout_model,
scaled_data, scaler, n_steps, start_date, '2030-12-31')

# Tabelle mit Vorhersagen und Konfidenzintervallen erstellen
predictions_df = pd.DataFrame({
    'Date': dates,
    'Predicted_Value': predictions,
    'Lower_Confidence': lower_confidence,
    'Upper_Confidence': upper_confidence
})

# Filtern der Daten ab 2019 plot_data = predictions_df[predictions_df['Date'] >= '2019-01-01']

# Plot der Vorhersagen und Konfidenzintervalle plt.figure(figsize=(10, 6))
plt.plot(predictions_df['Date'], predictions_df['Predicted_Value'], la-
bel='Predicted Value') plt.fill_between(predictions_df['Date'], predictions_df['Lower_Confidence'], predictions_df['Upper_Confidence'], al-
pha=0.2) plt.xlabel('Date') plt.ylabel('Value') plt.title('Monthly Predictions
from July 2023 to December 2030') plt.legend()

plt.show()

```

```
# Anzeige der Tabelle print(predictions_df)
```

Du müsstest die Platzhalter für die Berechnung der Konfidenzintervalle entsprechend deiner Anforderungen ersetzen. Diese zeigen lediglich ein Beispiel mit einem festen Intervall um die Vorhersagen an. Dieser Code erstellt eine Tabelle mit den Vorhersagen und Konfidenzintervallen sowie einen Plot, der die Vorhersagen und Intervalle visualisiert.

**passe den code an meine daten timeseries\_df an:**

```
# resample with replacement each row boot_slopes = [] boot_interc = [] n_boots = 100 plt.figure() for _ in range(n_boots): # sample the rows, same size, with replacement sample_df = data_df.sample(n=n_points, replace=True) # fit a linear regression ols_model_temp = sm.ols(formula = 'y ~ x', data=sample_df) results_temp = ols_model_temp.fit() #      append      coefficients boot_interc.append(results_temp.params[0]) boot_slopes.append(results_temp.params[1]) # plot a greyed out line y_pred_temp = ols_model_temp.fit().predict(sample_df['x']) plt.plot(sample_df['x'], y_pred_temp, color='grey', alpha=0.2)# add data points plt.scatter(x, y) plt.plot(x, y_pred, linewidth=2) plt.grid(True) plt.xlabel('x') plt.ylabel('y') plt.title('x vs y') plt.show()  GesamtBruttoleistung Anzahl_Volleinspeisung Anzahl_Teileinspeisung  
Einspeisevergütung Strompreis EEG_Umlage PV_Kosten_Euro_pro_kWp Jahr_Monat 2000-01-01 42308.897 6944 2620 50.62 14.92 0.2 8100 2000-02-01 757.749 99 25 50.62  
14.92 0.2 8100 2000-03-01 698.163 163 21 50.62 14.92 0.2 8100 2000-04-01 3479.965  
871 153 50.62 14.92 0.2 8100 2000-05-01 3285.015 709 162 50.62 14.92 0.2 8100  
... ...
```

```
... ... ... ... ... 2023-08-01 2304090.367 3807 98267 7.65 46.27 0.0 1250 2023-09-
01 1358857.379 3443 89448 7.65 46.27 0.0 1250 2023-10-01 2101186.830 3008
77553 7.65
46.27 0.0 1250 2023-11-01 1581038.041 2221 66049 7.65 46.27 0.0 1250 2023-12-
01 1162517.318 1597 36900 7.65 46.27 0.0 1250 288 rows × 7 columns
```

Hier ist eine generische Funktion, die das Bootstrapping für alle Spalten von 'GesamtBruttoleistung' in deinem DataFrame durchführt:

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot
as plt

def bootstrap_regression(data_df, response_col, n_boots=100):
    boot_results = {}

    for predictor_col in data_df.columns:
        if predictor_col != response_col:
            boot_slopes = []
            boot_interc = []

            for _ in range(n_boots):
                # Perform bootstrapping here
                # ...

                # Append results to lists
                boot_slopes.append(slope)
                boot_interc.append(interc)

            # Calculate summary statistics
            # ...

            boot_results[predictor_col] = {
                "slopes": boot_slopes,
                "interc": boot_interc
            }

    return boot_results
```

```

# sample the rows, same size, with replacement
sample_df = data_df.sample(n=len(data_df), replace=True)

# fit a linear regression

ols_model_temp      = sm.OLS(sample_df[response_col],
sm.add_constant(sample_df[predictor_col]))           results_temp =
ols_model_temp.fit()

# append coefficients          boot_interc.append(re-
sults_temp.params[0])          boot_slopes.append(re-
sults_temp.params[1])

boot_results[predictor_col] = {'intercepts': boot_interc, 'slopes':
boot_slopes}

return boot_results

```

```
# Beispieldruck boot_results = bootstrap_regression(timeseries_df,
'GesamtBruttolistung', n_boots=100)
```

Diese Funktion gibt ein Dictionary zurück, das für jede unabhängige Variable (ausgenommen der abhängigen Variable 'GesamtBruttolistung') eine Liste von

Bootstrap-Intercept- und -Slope-Werten enthält. Du kannst dann die Verteilung der Steigungen und der y-Achsenabschnitte für jede Variable visualisieren und analysieren.

### **bitte den code zur visualisierung**

Natürlich, hier ist ein Beispielcode, um die Verteilung der Steigungen und y-Achsenabschnitte für jede Variable zu visualisieren:

```
import seaborn as sns
```

```
def visualize_bootstrap_results(boot_results):    for predictor_col, values in boot_results.items():  
    plt.figure(figsize=(12, 6))  
  
    # Verteilung der Intercepts      plt.subplot(1, 2, 1)  
    sns.histplot(values['intercepts'], bins=30, kde=True)  
    plt.title(f'Bootstrap Verteilung der Intercept für {predictor_col}')    plt.xlabel('Intercept')    plt.ylabel('Häufigkeit')  
  
    # Verteilung der Slopes        plt.subplot(1, 2, 2)  
    sns.histplot(values['slopes'], bins=30, kde=True)  
    plt.title(f'Bootstrap Verteilung der Slope für {predictor_col}')    plt.xlabel('Slope')    plt.ylabel('Häufigkeit')
```

```
plt.tight_layout()  
plt.show()  
  
# Beispielaufruf visualize_bootstrap_re-  
sults(boot_results)
```

Dieser Code verwendet **seaborn** für die Histogramme. Du kannst die Funktion **visualize\_bootstrap\_results** mit dem Ergebnis deiner **bootstrap\_regression** Funktion aufrufen, um die Verteilung der Intercept- und Slope-Werte für jede unabhängige Variable zu sehen.

**Berechne und visualisiere die 95% Konfidenzintervalle um den zentralen wert und plotte die Ergebnisse für jeden predictor.**

Um Konfidenzintervalle (CIs) für die Intercepts und Slopes aus den Bootstrapverteilungen zu ziehen, kannst du die Quantile der Verteilung verwenden. Hier ist ein Beispielcode, wie du das machen kannst:

```
def bootstrap_regression(data_df, response_col, n_boots=100):
```

```
    boot_results = {}  
  
    for predictor_col in data_df.columns:  
        if predictor_col != response_col:  
            boot_slopes = []  
            boot_interc = []
```

```

for _ in range(n_boots):

    # sample the rows, same size, with replacement
    sample_df = data_df.sample(n=len(data_df), replace=True)

    # fit a linear regression

    ols_model_temp      = sm.OLS(sample_df[response_col],
sm.add_constant(sample_df[predictor_col]))
    results_temp = ols_model_temp.fit()

    # append coefficients
    boot_interc.append(results_temp.params[0])
    boot_slopes.append(results_temp.params[1])

    boot_results[predictor_col] = {'intercepts': boot_interc, 'slopes': boot_slopes}

return boot_results

```

boot\_results = bootstrap\_regression(timeseries\_df, 'GesamtBruttoleistung', n\_boots=100)

# Berechne die 95% Konfidenzintervalle ci\_interc = {predictor: np.percentile(results['intercepts'], [2.5, 97.5]) for predictor, results in boot\_results.items()}

```

ci_slopes = {predictor: np.percentile(results['slopes'], [2.5, 97.5]) for predictor, results in boot_results.items()}

# Visualisiere die Bootstrap-Verteilungen und Konfidenzintervalle num_cols = 3 num_rows =
int(np.ceil(len(boot_results) / num_cols))

plt.figure(figsize=(15, 5 * num_rows))

for i, (predictor, results) in enumerate(boot_results.items(), 1):

    plt.subplot(num_rows, num_cols, i) plt.hist(results['intercepts'],
bins=30, edgecolor='black', color='skyblue', alpha=0.7) plt.axvline(ci_interc[predictor][0], color='red', linestyle='dashed', linewidth=2, label='2.5% CI') plt.axvline(ci_interc[predictor][1], color='green', linestyle='dashed', linewidth=2, label='97.5% CI') plt.title(f'Bootstrap Verteilung des Intercepts für {predictor}') plt.xlabel('Intercept') plt.ylabel('Häufigkeit')
plt.legend()

plt.tight_layout()
plt.show()

```

In diesem Code erstellen wir Histogramme für die Bootstrap-Verteilungen von

Intercept und Slope und zeichnen dann die 95% Konfidenzintervalle als gestrichelte Linien ein. Du kannst die Prozenteile (hier 2.5 und 97.5) je nach gewünschtem Konfidenzniveau anpassen.

## **Ich möchte Punktschätzungen der Koeffizienten sowie deren Standardfehler und p-Werte untersuchen**

Wenn du die Punktschätzungen der Koeffizienten, ihre Standardfehler und pWerte für deine lineare Regression analysieren möchtest, kannst du dies direkt aus den Ergebnissen deines OLS-Modells in Statsmodels tun. Hier sind die

Schritte, wie du das tun kannst:

```
# Beispiel für lineare Regression mit Stats-
models
OLS ols_model = sm.OLS(y,
sm.add_constant(X))
results = ols_mo-
del.fit()
```

```
# Punktschätzungen der Koeffizienten
coefficients = results.params
print("Punktschätzungen der Koeffi-
zienten:")
print(coefficients)
```

```
# Standardfehler der Koeffizienten stan-
dard_errors = results.bse
print("\nStandardfehler der Koeffizienten:")
print(standard_errors)
```

```
# p-Werte der Koeffizienten
p_values = results.pvalues
print("\np-Werte der Koeffizien-
ten:")
print(p_values)
```

Dieser Code zeigt dir die Punktschätzungen der Koeffizienten (coefficients), die Standardfehler der Koeffizienten (standard\_errors) und die p-Werte der Koeffizienten (p\_values).

Interpretation:

Punktschätzungen der Koeffizienten: Diese repräsentieren die geschätzten Veränderungen in der abhängigen Variable für eine Einheit Veränderung in der entsprechenden unabhängigen Variable, wenn alle anderen Variablen konstant gehalten werden.

Standardfehler der Koeffizienten: Diese geben an, wie genau die Punktschätzungen sind.

Kleinere Standardfehler deuten auf präzisere Schätzungen hin.

p-Werte der Koeffizienten: Diese liefern Informationen darüber, ob die geschätzten Koeffizienten signifikant von Null abweichen. Ein kleiner p-Wert (typischerweise kleiner als 0,05) deutet darauf hin, dass der Koeffizient signifikant ist.

Du kannst diese Informationen verwenden, um festzustellen, welche Koeffizienten signifikant sind und welche möglicherweise nicht. Wenn der p-Wert für einen Koeffizienten hoch ist, könnte dies darauf hinweisen, dass dieser Koeffizient nicht signifikant von Null abweicht und möglicherweise aus dem Modell entfernt werden könnte.