



Tunesische Republik  
Ministerium für Hochschulbildung und wissenschaftliche  
Forschung  
Universität von Tunis  
Nationale Hochschule für Ingenieurwesen in Tunis



---

# Projektbericht

## Schätzung der Gebrauchtwagenpreise

**Bearbeiterin :** Nawres Zaidoun

**Studienabschnitt :** 1. Studienjahr der  
Ingenieurausbildung in Angewandter Mathematik  
und Modellierung

**Betreuer :** Prof. Soufiane Gasmi

**Studienjahr :** 2022-2023

---

# Danksagung

Ich möchte mich bei allen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt und ermutigt haben. Zunächst möchte ich mich bei Professor Soufiane Gasmi bedanken, der mein Abschlussprojekt im ersten Jahr betreut und bewertet hat. Seine hilfreichen Anregungen und konstruktive Kritik während der Arbeit an diesem Bericht waren von unschätzbarem Wert.

Ein besonderer Dank geht auch an das "Schulen: Partner der Zukunft" PASCH-Programm, das mir im Jahr 2018 ein Stipendium ermöglichte, um Deutsch in einer internationalen Gemeinschaft zu lernen und meine Deutschkenntnisse an der Birklehof-Schule zu vertiefen. Diese Erfahrung war bei der Erstellung dieses Berichts sehr hilfreich.

Zuletzt möchte ich meinen Eltern danken, die mir durch ihre Unterstützung mein Studium ermöglicht haben und stets aufmerksam zugehört haben.

# Abstract

Als Reaktion auf die steigenden Trends beim Kauf und Verkauf von Automobilen gibt es ein wachsendes Interesse an der Preisprognose für Gebrauchtwagen. Die Erschwinglichkeit von Gebrauchtfahrzeugen befeuert dieses Interesse besonders, da die Verbraucherpräferenzen zu vorgeeigneten Fahrzeugen tendieren. Dieses Projekt zielt in erster Linie darauf ab, die Preise für Gebrauchtwagen auf der Grundlage signifikanter Attribute zu schätzen, die mit unserem Zielwert, dem Autopreis, korrelieren.

Um dies zu erreichen, habe ich Data-Mining-Techniken und Machine-Learning-Algorithmen eingesetzt und null, redundante und fehlende Werte im Datensatz während der Vorverarbeitungsphase herausgefiltert. Meine Untersuchung zum überwachten Lernen experimentierte mit drei Regressionsmodellen: Random Forest Regressor, Lineare Regression und Gradient Boosting Regressor. Diese Modelle wurden sorgfältig trainiert, getestet und miteinander verglichen.

Die Ergebnisse aus unserer Reihe von Experimenten zeigten, dass der Random Forest Regressor die anderen Modelle übertraf und die höchste Genauigkeit zeigte. Mit Blick auf die Zukunft stellt sich das Projekt vor, die verfeinertsten Algorithmen für genauere Vorhersagen zu nutzen. Ich hoffe, dass das entwickelte Modell in Zukunft in eine mobile Anwendung oder Web-Schnittstelle integriert wird, um der Öffentlichkeit einen zuverlässigen Zugang zu Schätzungen für Gebrauchtwagenpreise zu ermöglichen.

**Schlüsselwörter :** Schätzung der Gebrauchtwagenpreise, überwachtes Lernen (supervised learning), RandomForestRegressor.

# Inhaltsverzeichnis

Danksagung.....	2
Abstract.....	3
Inhaltsverzeichnis.....	4
Abbildungsverzeichnis.....	5
Tabellenverzeichnis.....	5
<b>1. Kapitel 1</b>	<b>6</b>
1.1 Hintergrund.....	6
1.2 Problemstellung .....	7
1.3 Projektziele.....	7
1.4 Methodologie.....	8
1.5 Einschränkungen der Studie.....	9
<b>2. Kapitel 2 - Literaturübersicht</b>	<b>10</b>
<b>3. Kapitel 3 – Projektbeschreibung</b>	<b>12</b>
3.1 Deutsche Gebrauchtwagen .....	12
3.2 Maschinelles Lernen.....	13
3.3 Beschreibung des Datensatzes .....	14
3.4 Vorverarbeitung des Datensatzes .....	16
3.5 Explorative Datenanalyse (EDA) .....	21
3.6 Bewertungsparameter des Modells.....	27
<b>4. Kapitel 4 – Projektanalyse</b>	<b>28</b>
4.1 Experimentelle Ergebnisse und Analyse.....	28
4.1.1 Lineare Regression .....	28
4.1.2 Random-Forest-Regressor.....	29
4.1.3 Gradient-Boosting-Regressor.....	29
4.2 Ergebnisvergleich.....	31
<b>5. Kapitel 5</b>	<b>32</b>
5.1 Schlussfolgerung .....	32
5.2 Empfehlungen und zukünftige Arbeiten.....	32
<b>6. Bibliographie</b>	<b>33</b>

# Abbildungsverzeichnis

Abbildung 1: Vorgeschlagene Methodik.....	8
Abbildung 2: Kategorien des maschinellen Lernens.....	13
Abbildung 3: Techniken des maschinellen Lernens.....	14
Abbildung 4: Website, von der die Daten gescraped wurden. ....	15
Abbildung 5: Vorverarbeitung des Attributs "Erstzulassung".....	17
Abbildung 6: Vorverarbeitung des Attributs 'Erstzulassung': Entfernung der Konvention 'EZ'.....	17
Abbildung 7: Vorverarbeitung des Attributs "Preis".....	17
Abbildung 8: Vorverarbeitung des Attributs "Kilometerstand".....	18
Abbildung 9: Attributdatentypen.....	18
Abbildung 10: Vorverarbeitung des Attributs "Name".....	18
Abbildung 11: Vorverarbeitung des Kaggle-Datensatzes.....	19
Abbildung 12: Gebrauchtwagen-Datensatz.....	19
Abbildung 13: Statistiken des Datensatzes.....	20
Abbildung 14: Entfernung von duplizierten Zeilen.....	20
Abbildung 15: Fehlende Zellen.....	20
Abbildung 16: Balkendiagramm für fehlende Werte.....	21
Abbildung 17: Heatmap für fehlende Werte.....	21
Abbildung 18: Autojahr-Boxplot.....	22
Abbildung 19: Attribut Marke – Balkendiagramm.....	22
Abbildung 20: Attribut Marke – Kraftstoff.....	23
Abbildung 21: Kraftstoff vs Preise.....	23
Abbildung 22: Kilometerstand vs Preise.....	23
Abbildung 23: Attributverteilungen.....	24
Abbildung 24: Attributverteilungen .....	25
Abbildung 25: Autojahr-Verteilungsdiagramm.....	25
Abbildung 26: Regression.....	26
Abbildung 27: Kopf des Datensatzes mit Dummy-Variablen.....	26
Abbildung 28: RMSE-Gleichung.....	27
Abbildung 29: R_squared-Gleichung.....	27
Abbildung 30: Bewertungswerte des linearen Regressionsmodells.....	28
Abbildung 31: Bewertungswerte des Random Forest Regressor Modells.....	29
Abbildung 32: Bewertungswerte des Gradient Boosting Regressor Modells.....	30
Abbildung 33: Vorhersageauswertungstabelle.....	30

# Tabellenverzeichnis

Tabelle 1: Beschreibung des Datensatzes.....	16
Tabelle 2: Ergebnisvergleich.....	31

# Kapitel 1

## 1.1 Hintergrund

Heutzutage spielt die Transportindustrie eine wichtige Rolle in der Wirtschaft vieler Länder, wobei der Automobilsektor in den entwickelten Ländern oft als "Industrie der Industrien" bezeichnet wird. Deutschland ist keine Ausnahme, da es über eine der fortschrittlichsten und einflussreichsten Automobilindustrien der Welt verfügt. Deutsche Automobilmarken wie Mercedes-Benz, BMW, Audi und Volkswagen sind weltweit für ihre technischen Fähigkeiten bekannt und haben eine bedeutende Präsenz auf nationalen und internationalen Märkten.

Im Laufe der Jahre hat die deutsche Automobilindustrie ein stetiges Wachstum erlebt, wobei eine große Anzahl neuer und gebrauchter Autos gekauft und verkauft wird. Insbesondere der Gebrauchtwagenmarkt ist bei deutschen Verbrauchern aufgrund von Faktoren wie Erschwinglichkeit und wirtschaftlichen Bedingungen beliebt geworden. Die Vorhersage der Preise von Gebrauchtwagen in Deutschland ist eine komplexe Aufgabe, die Fachwissen und die Berücksichtigung verschiedener Faktoren und Merkmale erfordert.

Gebrauchtwagenpreise werden von einer Vielzahl von Faktoren beeinflusst, wie dem Herstellungsjahr, der Art des Kraftstoffs, dem Zustand, den gefahrenen Kilometern, der Motorleistung, der Anzahl der Türen, der Häufigkeit von Lackierungen, Kundenbewertungen und dem Gewicht, unter anderem. Das Sammeln eines umfassenden Datensatzes, der alle relevanten Merkmale enthält, kann eine Herausforderung sein, und die Verfügbarkeit solcher Daten ist oft begrenzt.

In diesem Projekt habe ich mich auf den deutschen Gebrauchtwagenmarkt konzentriert und eine Benchmark-Datenbank mit allen wichtigen Merkmalen erstellt. Der Datensatz wurde einer gründlichen Vorverarbeitung und Transformation unterzogen, bevor er in Modelle eingeführt wurde. Dies beinhaltete statistische Analysen, Identifizierung und Handhabung von fehlenden, duplizierten und Nullwerten sowie die Extraktion und Auswahl von Merkmalen.

Das Prognoseproblem für Gebrauchtwagenpreise in Deutschland kann als Regressionsproblem angesehen werden und gehört zum Bereich des überwachten Lernens. In diesem Projekt habe ich drei bekannte Regressionsmodelle trainiert und verglichen: Lineare Regression, Gradient-Boosting-Regressor und Random-Forest-Regressor. Der Random-Forest-Regressor hat in ähnlichen Projekten vielversprechende Ergebnisse gezeigt und war daher meine erste Wahl für das Hauptalgorithmusmodell. Durch die präzise Vorhersage der Gebrauchtwagenpreise in Deutschland mit maschinellem Lernen möchte ich sowohl Käufern als auch Verkäufern ein intelligentes System zur Verfügung stellen, dass eine effiziente und fundierte Entscheidungsfindung auf dem Gebrauchtfahrzeugmarkt ermöglicht.

## 1.2 Problemstellung

Das Ziel dieser Forschung besteht darin, die Preise für gebrauchte Autos zu untersuchen und einen datengetriebenen Ansatz zur Vorhersage des Werts von Gebrauchtwagen zu entwickeln. Wir werden Methoden des maschinellen Lernens einsetzen, um Daten aus deutschen Websites, die gebrauchte Autos anbieten, zu analysieren. Durch die Untersuchung verschiedener Aspekte und Faktoren, die den tatsächlichen Preis eines gebrauchten Autos beeinflussen, wird unser Modell es Verbrauchern ermöglichen, den Wert ihrer aktuellen oder gewünschten Fahrzeuge anhand einer Reihe von Merkmalen zu schätzen. Insgesamt zielt diese Studie darauf ab, die Transparenz und Genauigkeit der Gebrauchtwagenpreise auf dem deutschen Markt zu verbessern.

## 1.3 Projektziele

Das Ziel dieses Projekts besteht darin, Modelle zur Preisvorhersage zu entwickeln, die der Allgemeinheit zugänglich gemacht werden. Sie bieten wertvolle Einblicke für Personen, die den Autokauf oder -verkauf in Erwägung ziehen, und erleichtern den oft schwierigen Prozess beim Händler. Die Studie zielt darauf ab, Verbraucher mit den notwendigen Werkzeugen zu versorgen, um unethischen Händlertaktiken nicht zum Opfer zu fallen und so ein besseres Einkaufserlebnis zu ermöglichen. Neben der Unterstützung von Verbrauchern strebt das Projekt die Erforschung neuer Methoden zur Bewertung von Gebrauchtwagenpreisen an und vergleicht deren Genauigkeit mit früheren Forschungsergebnissen. Es handelt sich um ein

fortlaufendes, interessantes Forschungsthema. Die Studie hofft, auf der Grundlage früherer Arbeiten signifikante Ergebnisse zu erzielen und fortgeschrittene Methoden einzusetzen.

## 1.4 Methodologie

Dieses Projekt konzentriert sich auf den Markt für Gebrauchtwagen in Deutschland und Europa insgesamt. Der Datensatz wurde zunächst durch das Scraping von 12gebrauchtwagen.de erlangt und später haben wir einen zusätzlichen Datensatz von Kaggle.com integriert, um seine Größe zu erhöhen und die Genauigkeit unseres intelligenten Modells zu verbessern. Die Methodik des Projekts ist wie folgt strukturiert :

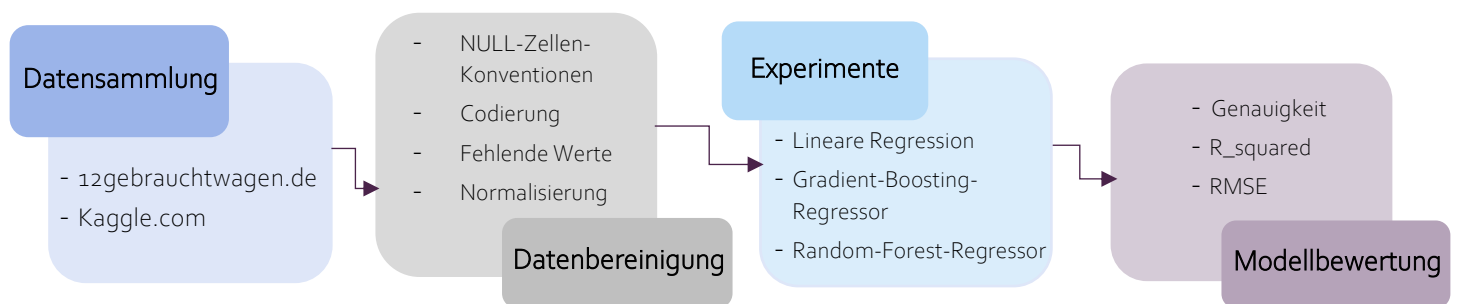


Abbildung 1: Vorgeschlagene Methodik

Nach der Datensammlung wurde der Datensatz einer Vorverarbeitung unterzogen, um fehlende Werte zu beseitigen, kategoriale Variablen in numerische umzuwandeln und irrelevante Attribute zu eliminieren. Gleichzeitig wurden Inkonsistenzen in den Einheiten korrigiert. Eine fundierte Datenkenntnis und adäquate Vorbereitung sind für den Aufbau eines effektiven Modells von entscheidender Bedeutung. Sie liefern Einsichten in erforderliche Anpassungen vor der Modellentwicklung und -implementierung. Dies umfasst eine vorläufige Datenanalyse, die die Identifizierung von Ausreißern und Verzerrungen sowie statistische Untersuchungen kategorialer und numerischer Variablen einschließt. Nachdem die Daten organisiert und für maschinelles Lernen optimiert wurden, wurden drei Modelle zur Vorhersage von Gebrauchtwagenpreisen entwickelt: Lineare Regression, Random Forest Regression und Gradient-Boosting-Regression. Die Modellbewertung basierte auf dem Root-Mean-Square-Error (RMSE) und dem R-Quadrat. Dabei erzielte der Random Forest Regressor die besten Ergebnisse aller drei Modelle.



## 1.5 Einschränkungen der Studie

Die Pandemie, welche einen Halbleitermangel auslöste, hat in den letzten Jahren tiefgreifende Veränderungen in der Automobilindustrie bewirkt. Dadurch sind die Preise für Gebrauchtwagen gestiegen und die Kosten für Neuwagen haben sich rasant verändert. Diese Entwicklung könnte die Methoden zur Vorhersage zukünftiger Autopreise beeinflussen. Daher kann der aktuelle Datensatz keinen fairen Marktpreis für Fahrzeuge liefern. Eine mobile App mit einem Echtzeit-Datenmodell könnte eine optimale Lösung für die Öffentlichkeit darstellen, um dieses Problem zu bewältigen.

# Kapitel 2

## Literaturübersicht

Zahlreiche Studien haben verschiedene Ansätze und Methoden verwendet, um Gebrauchtwagenpreise weltweit vorherzusagen. Die erzielten Genauigkeitswerte reichen von fünfzig Prozent bis neunzig Prozent. Um die Mauritius-Gebrauchtwagenpreise zu prognostizieren, verwendete Pudaruth (2014) verschiedene maschinelle Lerntechniken. Basierend auf historischen Zeitungsdaten verwendete er Entscheidungsbäume, K-Nearest-Neighbours, Multiple Regression und Naive-Bayes-Algorithmen. Die Genauigkeit lag zwischen 60 und 70 %. Um die Genauigkeit zu erhöhen, wurde empfohlen, ausgefeiltere Modelle und Algorithmen einzusetzen und die Datenbasis zu erweitern.

Monburinon et al. (2018) prognostizierten Gebrauchtwagenpreise anhand von Daten einer deutschen E-Commerce-Website mit 304.133 Zeilen und elf Attributen. Der Gradient Boosted Regression Tree wurde verwendet, um den geringsten mittleren absoluten Fehler (MAE) von 0,28 zu erreichen. Die Autoren schlugen vor, die Parameter in zukünftigen Arbeiten zu ändern und One-Hot-Encoding anstelle von Label-Encoding zu verwenden, um eine realistischere Interpretation kategorialer Daten zu ermöglichen. Gegic et al. (2019) von der International Burch University in Sarajevo nutzten Support Vector Machine, Random Forest und Artificial Neural Network als maschinelle Lernverfahren. Ihre Ergebnisse zeigten, dass die Kombination der Algorithmen mit einer Random Forest-Preisgruppierung eine Genauigkeit von bis zu 87,38% erzielen konnte. Noor und Jan (2017) erzielten mithilfe von multiplen linearen Regressionsmodellen eine hohe Genauigkeit von 98 % bei der Preisvorhersage von Gebrauchtwagen. Pak Wheels, eine Website, die Gebrauchtwagen verkauft, wurde von ihnen verwendet. K.Samruddhi und Kumar (2020) verwendeten den K-Nearest-Neighbour-Algorithmus und eine Kreuzvalidierung, um eine Genauigkeit von bis zu 85 % zu erreichen.

Gongqi et al. (2011) schlugen vor, ein künstliches Neuronales Netzwerk (KNN) zu verwenden, und erzielten genaue Vorhersagewerte mit einem kombinierten Ansatz aus

BP-Neuronalem Netzwerk und nichtlinearer Kurvenanpassung. Support Vector Machines (SVMs) wurden von Listiani (2009) verwendet, um Leasingwagenpreise zu bewerten. Er fand heraus, dass SVMs bei großen Datensätzen mit hoher Dimensionalität genauer waren als ML-Regression.

Kuiper (2008) sammelte Daten von General Motors über Autos, die im Jahr 2005 hergestellt wurden, und verwendete auch eine Variable Auswahltechnik, um die am meisten relevante Eigenschaft in seinem Modell zu berücksichtigen. Er empfahl die Verwendung eines multivariaten Regressionsmodells. Forscher nutzten Random Forest, eine überwachte Lernstrategie, um den Preis von Gebrauchtwagen vorherzusagen (Nabarun Pal, 2018). Der Datensatz von Kaggle wurde verwendet, um den Preis von Gebrauchtwagen vorherzusagen. Der Preiseinfluss jeder Eigenschaft wurde durch eine sorgfältige explorative Datenanalyse ermittelt. Random Forest trainierte 500 Entscheidungsbäume. Random Forest wird normalerweise für die Klassifikation verwendet, aber in diesem Fall wurde das Problem zu einem äquivalenten Regressionsproblem umgewandelt. Die experimentellen Ergebnisse zeigten eine 95,82 %ige Trainingsgenauigkeit und eine 83,63 %ige Testgenauigkeit. Das Modell konnte den Autopreis genau vorhersagen, indem es die am stärksten korrelierenden Merkmale ausgewählt hatte.

Eine andere Gruppe von Forschern hat sich auf dieses Thema spezialisiert (Jian Da Wu, 2017) und versucht, ein System zu entwickeln, das aus drei Komponenten besteht: einer Leistungsanalyse, einem Preisprognosealgorithmus und einem Datenerfassungssystem. Der vorgeschlagene ANFIS und ein herkömmliches künstliches neuronales Netzwerk (ANN) wurden aufgrund ihrer adaptiven Lernfähigkeit verglichen. ANFIS enthält sowohl adaptive neuronale Netzwerkfähigkeiten als auch qualitative Fuzzy-Logik-Approximation. Im Experiment zeigte die Verwendung von ANFIS als Expertensystem zur Vorhersage von Gebrauchtwagenpreisen bessere Ergebnisse. Die Experimente haben gezeigt, dass das vorgeschlagene System genaue und einfache Preisprognosen liefern kann. Der Kunde kann genaue und einfache Informationen über den Kaufpreis von Gebrauchtwagen über eine grafische Benutzeroberfläche erhalten.

Es scheint, dass viele Forscher derzeit die Vorhersage des Preises von Gebrauchtwagen ein wichtiges Thema behandeln, wie aus allen Literaturübersichten hervorgeht. Bisher hat die Random-Forest-Technik die beste Genauigkeit von 83,63 % auf Kaggles Datensatz erzielt. Die Forscher haben eine Reihe von Regressoren getestet, und das letzte, das sie gefunden haben, ist ein lineares Regressionsmodell.

# Kapitel 3

## Projektbeschreibung

### 3.1 Deutsche Gebrauchtwagen

Der deutsche Gebrauchtwagenmarkt wurde im Jahr 2021 auf einen Wert von 113,2 Milliarden US-Dollar geschätzt und wird voraussichtlich bis zum Jahr 2027 einen Wert von 171,03 Milliarden US-Dollar erreichen. Dies entspricht einer durchschnittlichen jährlichen Wachstumsrate (CAGR) von 7,12% während des Prognosezeitraums (2022 - 2027).

Aufgrund der globalen COVID-19-Pandemie waren Länder gezwungen, vorübergehend die Nachfrage nach sowohl alten als auch neuen Fahrzeugen zu reduzieren. Darüber hinaus hat die globale wirtschaftliche Verlangsamung die Verbraucher beeinflusst und zu einem Rückgang der Autoverkäufe geführt. Aufgrund des Infektionsrisikos und des Wunsches nach sozialer Distanz haben Menschen auf der ganzen Welt jedoch begonnen, den öffentlichen Verkehr zu meiden. Laut dem KBA ist der Gebrauchtwagenmarkt in Deutschland im ersten Halbjahr 2021 um 6,8 Prozent geschrumpft. Allerdings durften Autohäuser im Land erst im März 2021 unter bestimmten Bedingungen wieder öffnen. Dies hatte natürlich einen größeren Einfluss auf den Neuwagenmarkt, mit einem Rückgang der Neuzulassungen um 24,8 Prozent in der ersten Hälfte des Jahres. Im Dezember 2020 prognostizierte Schwacke, dass im Jahr 2021 etwas mehr als sieben Millionen Gebrauchtwagen verkauft würden, etwa genauso viele wie im Jahr 2020. Darüber hinaus würde dieses Transaktionsvolumen um weniger als 200.000 Einheiten unter den 7,2 Millionen Fahrzeugen liegen, die vor der Pandemie den Besitzer wechselten. Preisunterschiede sind häufig und irreführende Preise auf jeder anderen Website, daher besteht die Notwendigkeit eines Tools, um den Preis von Gebrauchtwagen basierend auf realen Daten von lokalen Websites vorherzusagen und Verbrauchern eine genaue Bewertung der Fahrzeuge zu ermöglichen. Diese Studie wird eine einfache Benutzeroberfläche entwickeln, die für Verbraucher genau genug ist, um den Preis von Autos für Verkaufs- oder Kaufzwecke zu bewerten.

## 3.2 Maschinelles Lernen

Das Hauptziel des maschinellen Lernens (ML) besteht darin, Computern das Lernen zu ermöglichen, ohne dass ihnen ausdrücklich Anweisungen gegeben werden müssen. Dies geschieht durch die Verwendung mathematischer Modelle, um Daten zu analysieren. Künstliche Intelligenz (KI) ist ein Teilgebiet des maschinellen Lernens, das auf diesen Prinzipien aufbaut. Durch den Einsatz von Algorithmen werden Muster in den Daten identifiziert, die dann zur Erstellung von Modellen zur Vorhersage genutzt werden. Ähnlich wie Menschen verbessert sich das maschinelle Lernen mit zunehmender Menge an verfügbaren Daten und Erfahrung.

Ein großer Vorteil des maschinellen Lernens besteht darin, dass es sich an veränderliche Daten, sich ändernde Anforderungen oder Aufgabenstellungen anpassen kann, während das Programmieren einer Lösung nicht immer realisierbar ist.

Es gibt drei wichtige Kategorien des maschinellen Lernens :

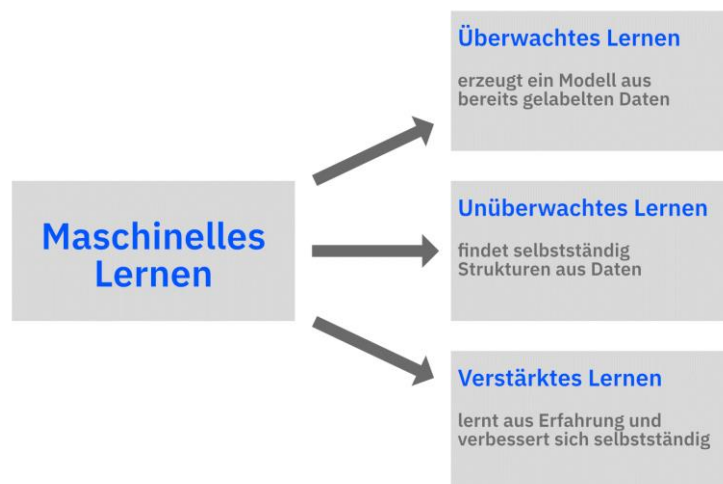


Abbildung 2: Kategorien des maschinellen Lernens

Überwachtes und unüberwachtes Lernen sind weit verbreitete Arten des maschinellen Lernens. Beim überwachten Lernen wird das Modell mit gekennzeichneten Daten trainiert, wobei die gewünschten Ausgaben bereitgestellt werden. Beim unüberwachten Lernen geht es hingegen darum, Muster oder Strukturen in nicht gekennzeichneten Daten ohne explizite Anleitung zu entdecken. Eine weitere wichtige Kategorie ist das verstärkte Lernen, das auf sequenzieller Entscheidungsfindung beruht. Dabei lernt ein Agent durch Versuch und Irrtum auf der Grundlage von Belohnungen oder Bestrafungen. Es ist jedoch wichtig zu

beachten, dass Maschinen auch nach Mai 2019 immer noch Training benötigen und ohne vorheriges Lernen keine autonomen Entscheidungen treffen können (Matthew Botvinick). Die Hauptkategorien des überwachten und unüberwachten maschinellen Lernens sind wie folgt:

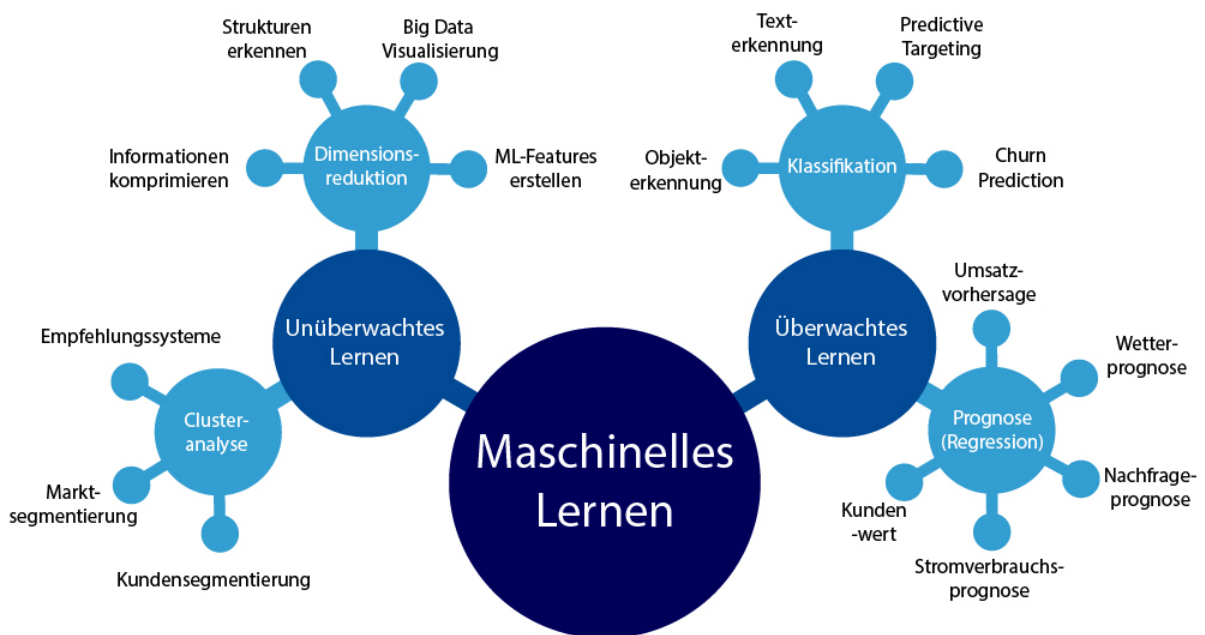


Abbildung 3: Techniken des maschinellen Lernens

### 3.3 Beschreibung des Datensatzes

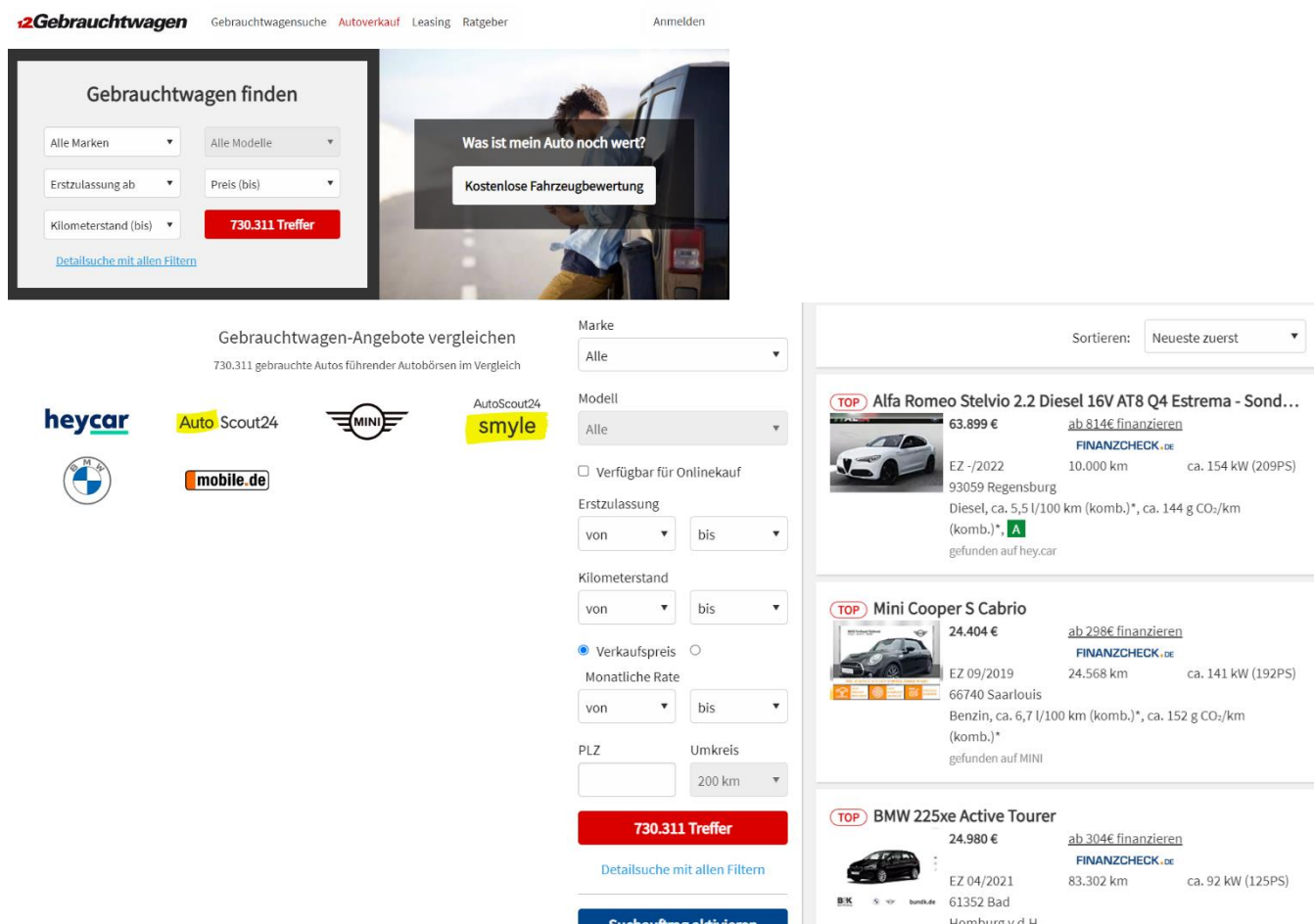
In diesem Projekt werden wir den Preis von Gebrauchtwagen anhand verschiedener Merkmale vorhersagen, wie Marke, Kilometerstand, Kraftstoff, Preis und Erstzulassung.

Wir werden Daten von einer deutschen Auto-Website scrapen. Wir verwenden BeautifulSoup, die Requests Library und Pandas Dataframe. Die Ergebnisse werden in einer Excel-Datei gespeichert. BeautifulSoup ist eine Python-Bibliothek, die speziell für das Parsen und Extrahieren von Daten aus HTML- und XML-Dokumenten entwickelt wurde. Mit BeautifulSoup können Webseiten nach relevanten Informationen durchsucht werden, indem der HTML-Code analysiert wird. In diesem Web-Scraping-Projekt wurde BeautifulSoup verwendet, um strukturierte Daten aus den gescrapten Webseiten zu extrahieren.

Die Requests-Bibliothek ist ebenfalls eine Python-Bibliothek, die das Senden von HTTP-Anfragen an Webseiten ermöglicht. Sie ermöglicht das Herunterladen von Webseiteninhalten, das Senden von POST- oder GET-Anfragen und die Verwaltung von

Cookies und Sitzungen. In diesem Web-Scraping-Projekt wurde die Requests-Bibliothek verwendet, um die Webseiteninhalte von den URLs abzurufen, die gescrapt werden sollten. Pandas ist eine leistungsstarke Python-Bibliothek, die die Arbeit mit Datenstrukturen und -analysen erleichtert. Der Pandas Dataframe ist eine zentrale Datenstruktur, die es ermöglicht, Daten in tabellarischer Form zu organisieren und zu manipulieren. Im Rahmen dieses Web-Scraping-Projekts wurde der Pandas Dataframe verwendet, um die gescrapteten Daten zu speichern, zu strukturieren und weiter zu analysieren. Er bietet Funktionen zum Filtern, Gruppieren, Sortieren und Zusammenführen der Daten, um sie besser zu verstehen und zu visualisieren.

Durch mehrere Durchläufe und Iterationen konnten wir erfolgreich 2.358 Datensätze mit 5 Variablen von der Website 12gebrauchtwagen.de sammeln. Diese Website ist eine Plattform für den Kauf und Verkauf von gebrauchten Fahrzeugen und bietet eine umfangreiche Auswahl an Autos verschiedener Marken, Modelle und Preisklassen.



The screenshot displays the 12Gebrauchtwagen website interface. At the top, navigation links include 'Gebrauchtwagensuche', 'Autoverkauf', 'Leasing', 'Ratgeber', and 'Anmelden'. The main search area, titled 'Gebrauchtwagen finden', features filters for 'Alle Marken', 'Alle Modelle', 'Erstzulassung ab', 'Preis (bis)', and 'Kilometerstand (bis)', with a red button indicating '730.311 Treffer'. A sidebar on the right offers a 'Kostenlose Fahrzeugbewertung' service. Below the search section, a comparison tool 'Gebrauchtwagen-Angebote vergleichen' is shown, listing logos for 'heycar', 'Auto Scout24', 'MINI', 'AutoScout24', 'smyle', and 'mobile.de'. The central part of the page contains detailed filters for 'Marke', 'Modell', 'Verfügbar für Onlinekauf', 'Erstzulassung', 'Kilometerstand', 'Verkaufspreis', 'Monatliche Rate', 'PLZ', and 'Umkreis', with a red button for '730.311 Treffer'. On the right, a list of car offers is displayed, including 'Alfa Romeo Stelvio 2.2 Diesel 16V AT8 Q4 Estrema - Sond...', 'Mini Cooper S Cabrio', and 'BMW 225xe Active Tourer', each with its price, financing options, and specifications.

Abbildung 4: Website, von der die Daten gescrapt wurden.

Um unseren Datensatz zu erweitern und die Genauigkeit unseres Modells zu verbessern, haben wir beschlossen, unseren "Web-gescrapten" Datensatz mit einem anderen Datensatz zu kombinieren, der von Kaggle stammt. Kaggle ist eine Online-Plattform, die eine vielfältige Sammlung von Datensätzen für Data Science und Maschinelles Lernen bereitstellt. Dieser zusätzliche Datensatz ist ein deutscher Datensatz von gebrauchten Autos.

Vor der Kombination der Datensätze haben wir eine Vorverarbeitung für jedes Attribut einzeln durchgeführt, um die Daten zu korrigieren und die Datentypen umzuwandeln. Die folgende Tabelle liefert Details und Beschreibungen des Datensatzes:

Nummer	Spaltenname	Datentyp	eindeutige Werte
1	Marke	object	76
2	Kilometerstand	float64	1745
3	Kraftstoff	object	7
4	Preis	float64	1680
5	Erstzulassung	float64	74

*Tabelle 1: Beschreibung des Datensatzes*

Basierend auf den oben genannten Informationen lässt sich feststellen, dass der Datensatz viele kategoriale Merkmale enthält, die in Ganzzahlen oder Gleitkommazahlen umgewandelt werden müssen. Darüber hinaus müssen redundante Daten entfernt werden. Als Folge davon ist eine Vorverarbeitung erforderlich.

## 3.4 Vorverarbeitung des Datensatzes

Die Vorverarbeitung ist eine Data-Mining-Technik, bei der rohe Daten in ein verständliches Format umgewandelt werden. Oftmals fehlen spezifische Aktivitäts- oder Trenddaten, und viele ungenaue Fakten sind in realen Daten enthalten. Dies kann zu einer Datensammlung von schlechter Qualität führen und wiederum zu Modellen von schlechter Qualität, die aus den Daten erstellt werden. Solche Probleme können durch die Vorverarbeitung der Daten gelöst werden. Die Vorverarbeitung im maschinellen Lernen ist der Prozess der Modifizierung oder Codierung von Daten, damit die Maschine sie leichter analysieren kann. Dadurch kann der Algorithmus die Daten nun richtig interpretieren.



In diesem Projekt wurden folgende Schritte zur Vorverarbeitung des Datensatzes durchgeführt.

- 1- Die Attribute "Erstzulassung" haben Werte im VARCHAR-Format und müssen in INT oder FLOAT umgewandelt werden.

```
In [42]: gw['Erstzulassung']

Out[42]: 0      1997
         1      2018
         2      2019
         3      2018
         4      2003
         ...
        2377    2007
        2378    2020
        2379    1999
        2380    2013
        2381    2009
        Name: Erstzulassung, Length: 2382, dtype: object

In [44]: gw = gw[gw.Erstzulassung != '-'].copy()
         gw['Erstzulassung'] = gw['Erstzulassung'].astype(str).astype(float)
```

Abbildung 5: Vorverarbeitung des Attributs "Erstzulassung"

Am Anfang waren die Werte im Format "EZ 12/2013". Es wurde beschlossen, die Konvention "EZ" zu entfernen, nur das Jahr beizubehalten und den Datentyp in FLOAT zu ändern.

Erstzulassung
EZ 11/1997
EZ 08/2018
EZ 07/2019
EZ 02/2018
EZ 01/2003
...
EZ 03/2007
EZ 02/2020
EZ 04/1999
EZ 12/2013
EZ 03/2009

```
for i in range(len(Gebrauchtwagen)):
    m = Gebrauchtwagen['Erstzulassung'][i].split('/')[1]
    Gebrauchtwagen['Erstzulassung'][i]=m
```

Erstzulassung
1997
2018
2019
2018
2003
...
2007
2020
1999
2013
2009

Abbildung 6: Vorverarbeitung des Attributs 'Erstzulassung': Entfernung der Konvention 'EZ'

- 2- Der Preis war ebenfalls im Format "22.500 €". Der Preis wurde in float umgewandelt, nachdem das €-Symbol entfernt wurde.

Preis
2.700 €
23.780 €
34.950 €
22.500 €
12.490 €
...
5.650 €
19.888 €
45.900 €
8.490 €
4.980 €

```
def remove2(string):
    for i in range(len(string)):
        if string[i] == '€':
            return string[:i]
    return string

for i in range(len(Gebrauchtwagen)):
    m = remove2(Gebrauchtwagen['Preis'][i])
    Gebrauchtwagen['Preis'][i]=m

gw['Preis'] = gw['Preis'].str.replace('.', '').astype(float)
```

Preis
2700.0
23780.0
34950.0
22500.0
12490.0
...
5650.0
19888.0
45900.0
8490.0
4980.0

Abbildung 7: Vorverarbeitung des Attributs "Preis"

3- Der Kilometerstand war ebenfalls im Format "95.064 km". Diese Spalte wurde umgewandelt, indem der String nach Entfernung des "km" in einen Float-Wert konvertiert wurde.

```
Gebrauchtwagen['Kilometerstand'] = Gebrauchtwagen['Kilometerstand'].apply(lambda x: x.strip('km'))
gw['Kilometerstand'] = pd.to_numeric(gw['Kilometerstand'], errors='coerce')
```

Abbildung 8: Vorverarbeitung des Attributs "Kilometerstand"

4- Aus Abbildung 7 lässt sich schließen, dass nun alle Attribute im geeigneten Format und Datentyp vorliegen.

```
gw.dtypes

Name          object
Kilometerstand float64
Kraftstoff     object
Preis          float64
Erstzulassung  float64
dtype: object
```

Abbildung 9: Attributdatentypen

Nun sind alle Attribute bereit für weitere Verarbeitungsschritte.

5- In der Spalte Name stehen 836 eindeutige Namen. Das ist wirklich schwer zu implementieren und eine Regression würde mehr als 300 Dummies bedeuten. Lassen Sie uns an der Marke arbeiten, nicht am Namen des Autos.

```
In [63]: def erstesWort(string):
          return string.split()[0]
          gw['Name'] = gw['Name'].apply(lambda x: erstesWort(x))

In [64]: gw['Name'].unique()

Out[64]: array(['Mercedes-Benz', 'Ford', 'VW', 'Peugeot', 'BMW', 'Audi', 'Hyundai',
               'Volvo', 'Porsche', 'Opel', 'Jaguar', 'Renault', 'Jeep', 'Seat',
               'Citroën', 'Fiat', 'Alfa', 'Saab', 'Mazda', 'Cadillac', 'Dacia',
               'Rolls-Royce', 'Lotus', 'Mini', 'Skoda', 'Smart', 'Nissan',
               'Daewoo', 'Maserati', 'Land', 'Corvette', 'Mitsubishi', 'Infiniti',
               'Ferrari', 'Toyota', 'Lexus', 'Alpina', 'Chevrolet', 'Suzuki',
               'Maybach', 'Bentley', 'Kia', 'Lada', 'Lamborghini', 'Fisker',
               'Chrysler', 'Hummer', 'Abarth', 'Honda', 'MG', 'Dodge', 'Aston',
               'Pontiac', 'Cupra', 'Lincoln', 'KTM', 'Others', 'Oldtimer',
               'Lancia', 'Subaru', 'Daihatsu', 'Austin', 'Ligier',
               'StreetScooter', 'SsangYong', 'Trabant', 'NSU', 'Moskvich', '9ff',
               'Caravans-Wohnm', 'AC', 'Morgan', 'IVECO', 'VAZ', 'GMC',
               'Wartburg', 'GAZ'], dtype=object)

In [65]: def replace(string):
          first_word = string.split()[0]
          if first_word.lower() == 'aston':
              first_word = 'Aston Martin'
          elif first_word.lower() == 'morgan':
              first_word = 'Morgan Motor Company'
          elif first_word.lower() == 'alfa':
              first_word = 'Alfa Romeo'
          elif first_word.lower() == 'vw':
              first_word = 'Volkswagen'
          elif first_word.lower() == 'mini':
              first_word = 'BMW'
          return first_word
          gw['Name'] = gw['Name'].apply(lambda x: replace(x))

In [68]: gw['Name'].unique()

Out[68]: array(['Mercedes-Benz', 'Ford', 'Volkswagen', 'Peugeot', 'BMW', 'Audi',
               'Hyundai', 'Volvo', 'Porsche', 'Opel', 'Jaguar', 'Renault', 'Jeep',
               'Seat', 'Citroën', 'Fiat', 'Alfa Romeo', 'Saab', 'Mazda',
               'Cadillac', 'Dacia', 'Rolls-Royce', 'Lotus', 'Skoda', 'Smart',
               'Nissan', 'Daewoo', 'Maserati', 'Land', 'Corvette', 'Mitsubishi',
               'Infiniti', 'Ferrari', 'Toyota', 'Lexus', 'Alpina', 'Chevrolet',
               'Suzuki', 'Maybach', 'Bentley', 'Kia', 'Lada', 'Lamborghini',
               'Fisker', 'Chrysler', 'Hummer', 'Abarth', 'Honda', 'MG', 'Dodge',
               'Aston Martin', 'Pontiac', 'Cupra', 'Lincoln', 'KTM', 'Others',
               'Oldtimer', 'Lancia', 'Subaru', 'Daihatsu', 'Austin', 'Ligier',
               'StreetScooter', 'SsangYong', 'Trabant', 'NSU', 'Moskvich', '9ff',
               'Caravans-Wohnm', 'AC', 'Morgan Motor Company', 'IVECO', 'VAZ',
               'GMC', 'Wartburg', 'GAZ'], dtype=object)

In [69]: gw = gw.rename(columns={'Name': 'Marke'})
```

Abbildung 10: Vorverarbeitung des Attributs "Name"

6- Um unseren Datensatz zu erweitern und unser Modell genauer zu machen, kombinieren wir unseren "web-scraped" Datensatz mit einem anderen Datensatz aus Kaggle. Zuerst entfernen wir bestimmte Spalten, die nicht mit unserem ersten Datensatz übereinstimmen. Anschließend werden die verbleibenden Spalten umbenannt, um eine einheitliche Benennung zu gewährleisten. Danach führen wir die Verknüpfung der beiden Datensätze durch. Eine weitere Verarbeitung erfolgt, indem wir bestimmte Werte in der Spalte "Kraftstoff" ersetzen, um eine einheitliche Kategorisierung zu erreichen.

```
gw2 = pd.read_csv("car_price_data1.csv")
gw2
```

	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year	Model
0	BMW	4200.0	sedan	277	2.0	Petrol	yes	1991	320
1	Mercedes-Benz	7900.0	van	427	2.9	Diesel	yes	1999	Sprinter 212
2	Mercedes-Benz	13300.0	sedan	358	5.0	Gas	yes	2003	S 500
3	Audi	23000.0	crossover	240	4.2	Petrol	yes	2007	Q7
4	Toyota	18300.0	crossover	120	2.0	Petrol	yes	2011	Rav 4
...	...	...	...	...	...	...	...	...	...
4340	Mercedes-Benz	125000.0	sedan	9	3.0	Diesel	yes	2014	S 350
4341	BMW	6500.0	sedan	1	3.5	Petrol	yes	1999	535
4342	BMW	8000.0	sedan	194	2.0	Petrol	yes	1985	520
4343	Toyota	14200.0	sedan	31	NaN	Petrol	yes	2014	Corolla
4344	Volkswagen	13500.0	van	124	2.0	Diesel	yes	2013	T5 (Transporter)

4345 rows x 9 columns

```
gw2 = gw2.drop(["Body", "EngineV", "Model", "Registration"], axis=1)

gw2 = gw2.rename(columns={'Brand': 'Marke'})
gw2 = gw2.rename(columns={'Price': 'Preis'})
gw2 = gw2.rename(columns={'Mileage': 'Kilometerstand'})
gw2 = gw2.rename(columns={'Engine Type': 'Kraftstoff'})
gw2 = gw2.rename(columns={'Year': 'Erstzulassung'})

gw2
```

	Marke	Preis	Kilometerstand	Kraftstoff	Erstzulassung
0	BMW	4200.0	277	Petrol	1991
1	Mercedes-Benz	7900.0	427	Diesel	1999
2	Mercedes-Benz	13300.0	358	Gas	2003
3	Audi	23000.0	240	Petrol	2007
4	Toyota	18300.0	120	Petrol	2011
...	...	...	...	...	...
4340	Mercedes-Benz	125000.0	9	Diesel	2014
4341	BMW	6500.0	1	Petrol	1999
4342	BMW	8000.0	194	Petrol	1985
4343	Toyota	14200.0	31	Petrol	2014
4344	Volkswagen	13500.0	124	Diesel	2013

4345 rows x 5 columns

Abbildung 11: Vorverarbeitung des Kaggle-Datensatzes

Dies ist die endgültige Form des Datensatzes, der vorverarbeitet wird.

```
df = pd.concat([gw1, gw2])

df
```

	Marke	Kilometerstand	Kraftstoff	Preis	Erstzulassung
0	Mercedes-Benz	278.000	Benzin	2700.0	1997.0
1	Ford	95.064	Diesel	23780.0	2018.0
2	Volkswagen	11.312	Benzin	34950.0	2019.0
3	Peugeot	80.675	Diesel	22500.0	2018.0
4	BMW	165.000	Benzin	12490.0	2003.0
...	...	...	...	...	...
4340	Mercedes-Benz	9.000	Diesel	125000.0	2014.0
4341	BMW	1.000	Petrol	6500.0	1999.0
4342	BMW	194.000	Petrol	8000.0	1985.0
4343	Toyota	31.000	Petrol	14200.0	2014.0
4344	Volkswagen	124.000	Diesel	13500.0	2013.0

6703 rows x 5 columns

Abbildung 12: Gebrauchtwagen-Datensatz

6- Ich habe die "dataprep" Bibliothek installiert und die Funktion "create\_report" importiert, um einen Bericht für den DataFrame "df" mit dem Titel "mein Bericht" zu generieren. Die "dataprep" Bibliothek ist eine Python-Bibliothek, die Tools zur Datenverarbeitung bietet und Datenverarbeitungsaufgaben vereinfacht sowie umfassende Berichte über Datensätze generiert. Die Funktion "create\_report" generiert in diesem Fall einen detaillierten Bericht, der Einblicke in die Datenqualität liefert. Im generierten Bericht wird festgestellt, wie in Abbildung 13 dargestellt, dass es 4,6% doppelte Zeilen und 0,5% fehlende Zellen gibt, was auf potenzielle Datenprobleme hinweist, die behoben werden müssen.

**Overview**

**Dataset Statistics**

Number of Variables	5
Number of Rows	8703
Missing Cells	175
Missing Cells (%)	0.5%
Duplicate Rows	309
Duplicate Rows (%)	4.6%
Total Size in Memory	992.1 KB
Average Row Size in Memory	151.6 B
Variable Types	Categorical: 2 Numerical: 3

Abbildung 13: Statistiken des Datensatzes

Die in Abbildung 14 gezeigten Schritte wurden unternommen, um doppelte Zeilen zu entfernen und bessere Ergebnisse zu erzielen.

- Identifizierung doppelter Zeilen

```
: duplicate_rows = df[df.duplicated()]
```

- Berechnung des Prozentsatzes der zu entfernenden doppelten Zeilen

```
: num_duplicate_rows = len(duplicate_rows)
percent = 4.6
num_to_remove = int(num_duplicate_rows * percent / 100)
```

- Entfernung der doppelten Zeilen

```
: df1 = df.drop(duplicate_rows.sample(num_to_remove).index)
```

Abbildung 14: Entfernung von duplizierten Zeilen

Wir haben uns entschieden, alle fehlenden Werte zu entfernen, obwohl dies nicht immer empfohlen wird. In diesem Fall war es jedoch akzeptabel, da weniger als 5 % der Daten betroffen waren. Wir haben die Funktion dropna() verwendet, um die fehlenden Werte zu entfernen, und schließlich haben wir die Ergebnisse erhalten, wie in Abbildung 15 gezeigt.

```
df_no_mv.isna().sum()
Marke      0
Kilometerstand  0
Kraftstoff  0
Preis      0
Erstzulassung  0
dtype: int64
```

Abbildung 15: Fehlende Zellen

## 3.4 Explorative Datenanalyse (EDA)

Explorative Datenanalyse besteht aus den folgenden Schritten:

1- Um die fehlenden Werte zu visualisieren, wurde der Datensatz mithilfe eines Balkendiagramms für fehlende Werte und einer Heatmap für fehlende Werte dargestellt.

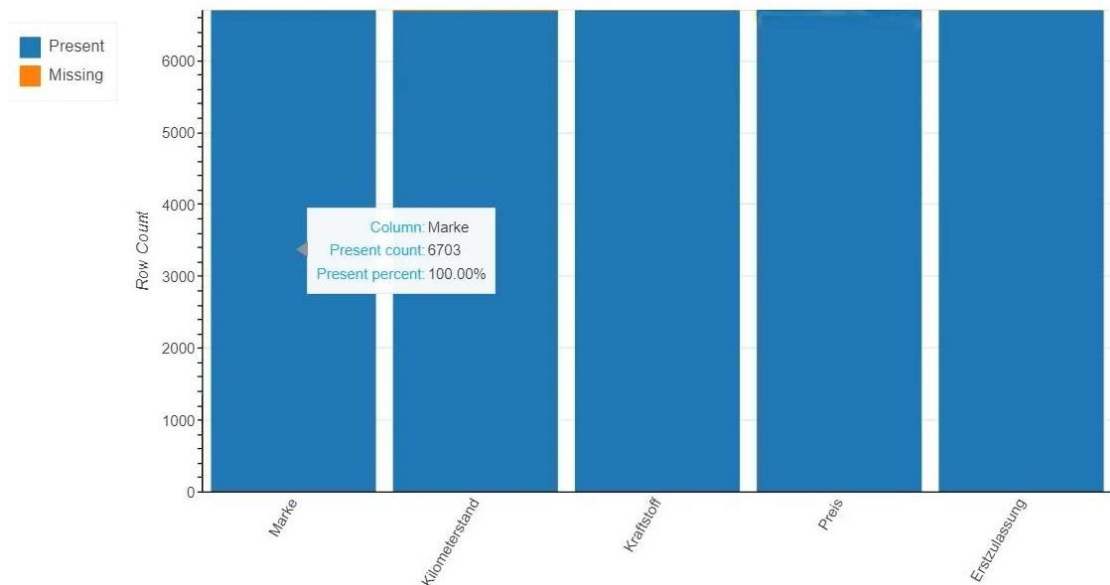


Abbildung 16: Balkendiagramm für fehlende Werte

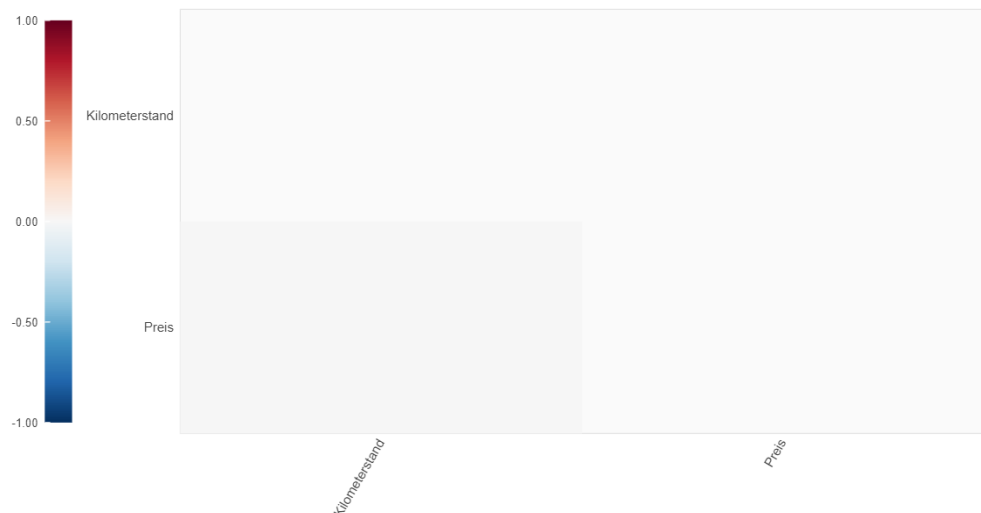


Abbildung 17: Heatmap für fehlende Werte

Aus Abbildung 4 und Abbildung 5 lässt sich schließen, dass der Datensatz keine Beispiele enthält, die Null- oder fehlende Werte enthalten.

2- Der Boxplot wird verwendet, um Ausreißer bei Integer-Attributen zu identifizieren. Daher wird das Attribut "Erstzulassung", das das Autojahr darstellt, visualisiert.

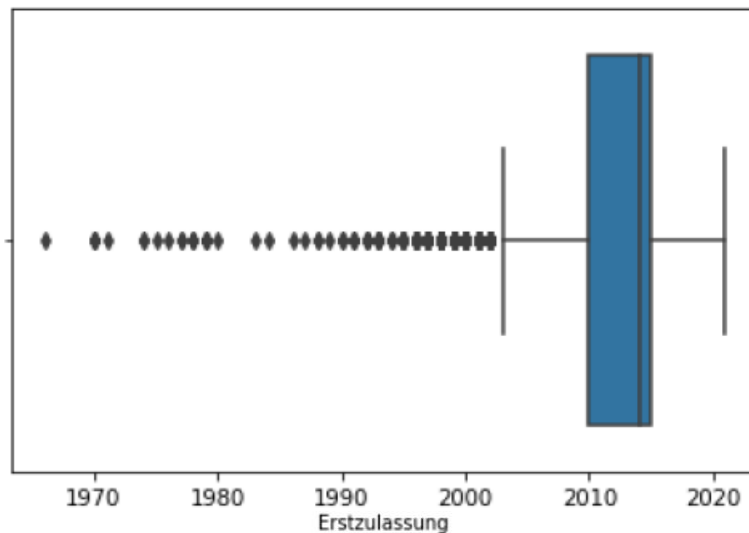
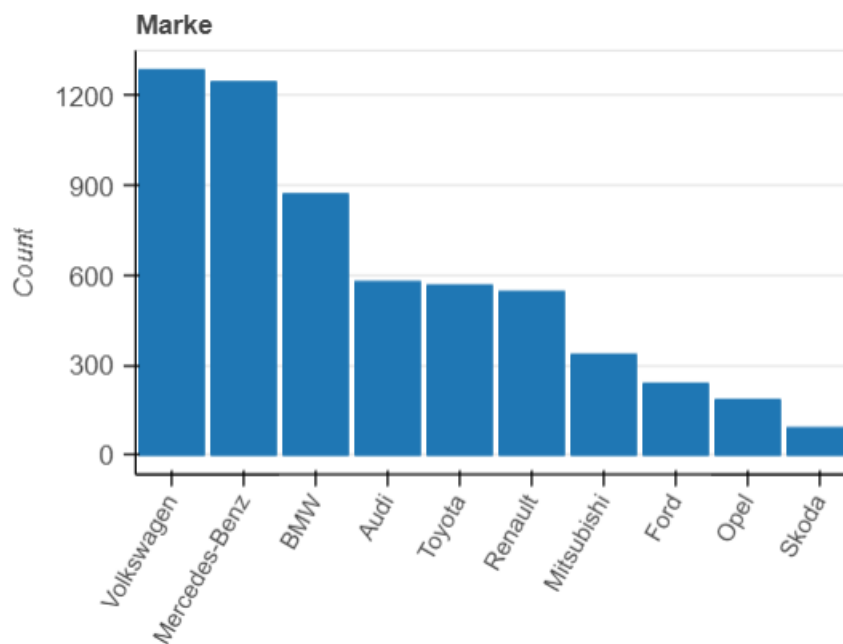


Abbildung 18: Autojahr-Boxplot

Wie in Abbildung 6 dargestellt, liegen die meisten Daten im Zeitraum von 2004 bis 2020. Ich habe mich entschieden, die Ausreißer beizubehalten, um einen Überblick über alle verschiedenen klassischen Autos zu erhalten.

3- Aus Abbildung 7 geht hervor, dass Volkswagen, die am häufigsten vorkommende Marke im Datensatz ist.



Top 10 of 76 Marke

Abbildung 19: Attribut Marke - Balkendiagramm

4- Abbildung 8 zeigt, dass Diesel, der am häufigsten vorkommende Kraftstoff ist.

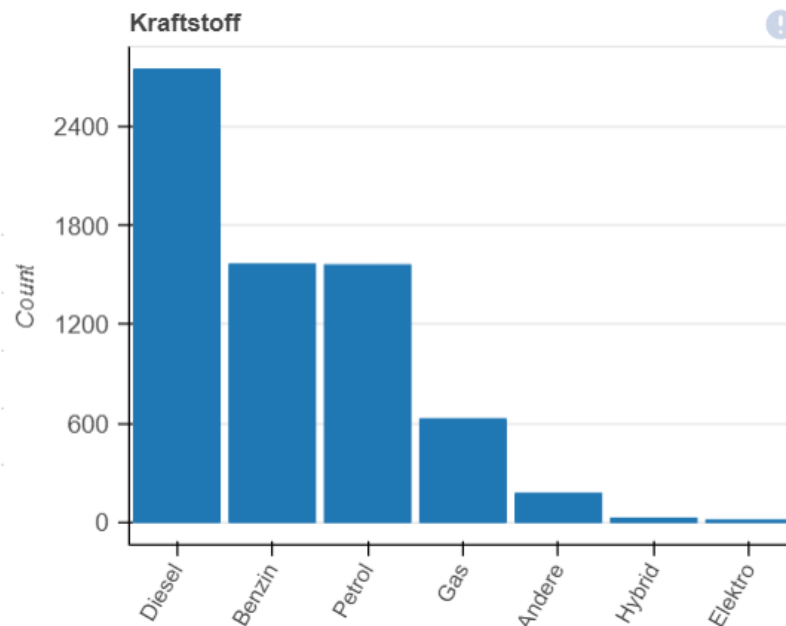


Abbildung 20: Attribut Marke - Kraftstoff

5- Beim Vergleich der Kraftstofftypen und Preise wurden folgende Ergebnisse erzielt

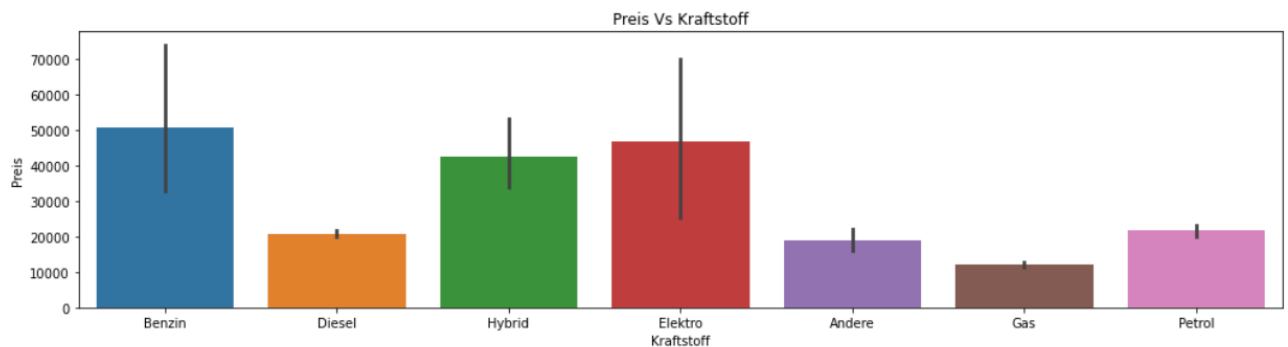


Abbildung 21: Kraftstoff vs Preise

Der Verkaufspreis von Fahrzeugen mit den Kraftstofftypen Hybrid, Elektro und Diesel ist höher als bei den anderen Typen.

Die Abbildung unten zeigt die Beziehung zwischen dem Kilometerstand und dem Preis. Je geringer die gefahrenen Kilometer eines Autos sind, desto höher ist der Preis.

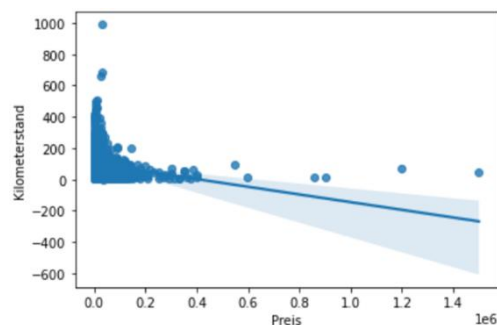


Abbildung 22: Kilometerstand vs Preise

6- Der letzte Schritt bei der Visualisierung besteht darin, die Schiefe der Attribute zu überprüfen, um festzustellen, ob sie normalverteilt sind oder nicht

Die Wahrscheinlichkeitsverteilung gibt uns Aufschluss darüber, wie eine Variable verteilt ist. Dadurch können Anomalien wie Ausreißer leicht erkannt werden.

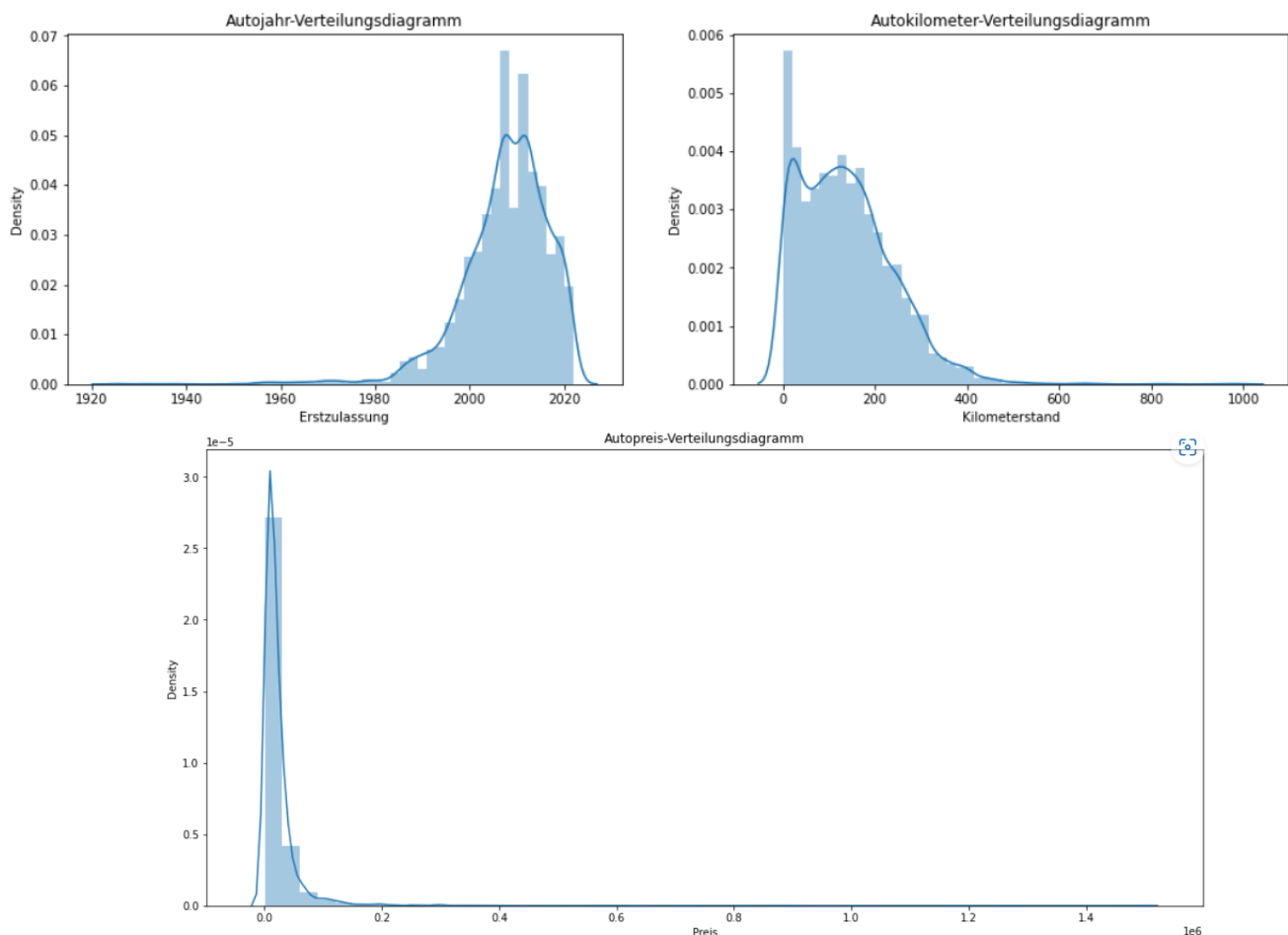


Abbildung 23: Attributverteilungen

Es ist klar, dass einige Ausreißer bei den numerischen Variablen Preis, Jahr und Kilometerstand auftreten. Hier befinden sich die Ausreißer um die höheren Preise herum (rechte Seite des Diagramms). Wir können das Problem leicht lösen, indem wir 0,5 % oder 1 % der problematischen Stichproben entfernen. Da es sich um einen Datensatz über Gebrauchtwagen handelt, kann man sich vorstellen, dass 1000000 ein überhöhter Preis ist.



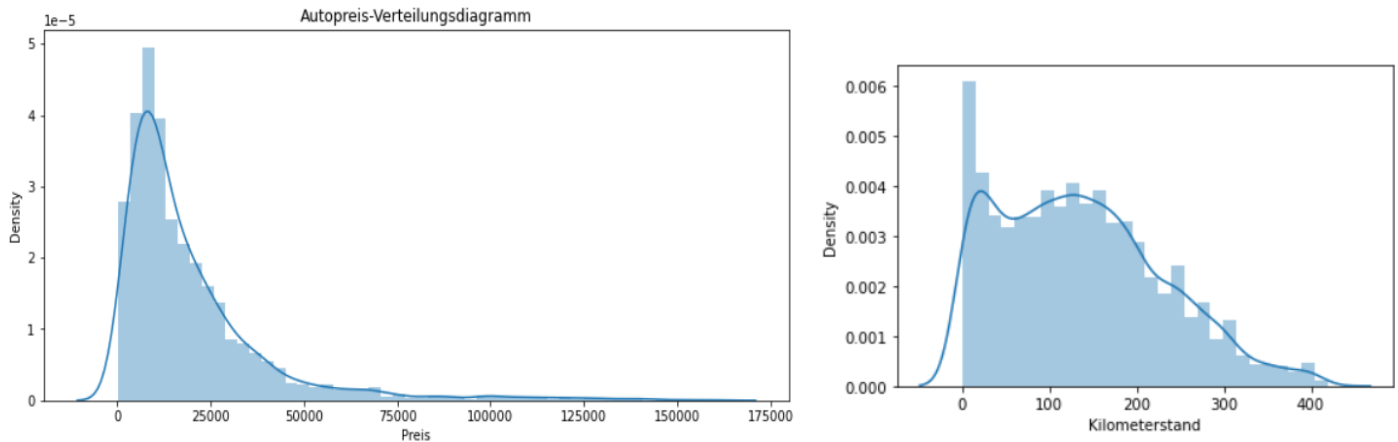


Abbildung 24: Attributverteilungen

In der Spalte "Erstzulassung" gibt es Ausreißer am unteren Ende. Um diese Ausreißer zu behandeln, wurde der Schwellenwert  $q$  berechnet, der dem 1. Perzentil (Quantile) entspricht. Anschließend wurde ein neuer Datensatz erstellt, der nur die Daten enthält, bei denen die "Erstzulassung" größer als dieser Schwellenwert  $q$  ist. Dadurch wurden die Ausreißer entfernt und der neue Datensatz enthält nur noch die relevanten Daten.

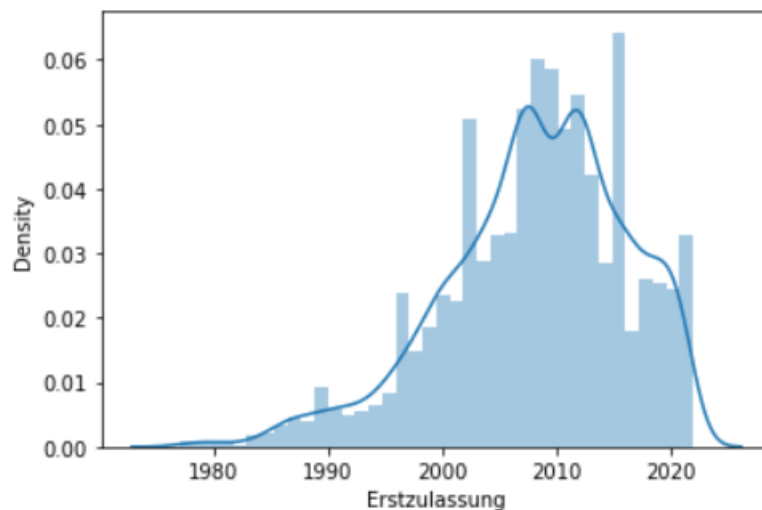


Abbildung 25: Autojahr-Verteilungsdiagramm

7- Ich habe die Variable "Preis" mittels einer logarithmischen Transformation transformiert, um die Regressionsanalyse zu verbessern. Diese Transformation skaliert die Datenpunkte auf eine logarithmische Skala, was zu einer verbesserten Symmetrie und Linearität führt. Darüber hinaus hilft sie, den Einfluss von Ausreißern zu mindern und eine stabilere Beziehung zwischen der abhängigen Variable (Preis) und den unabhängigen Variablen herzustellen. Die logarithmische Transformation trägt außerdem dazu bei, Heteroskedastizität zu reduzieren

und den Annahmen der linearen Regression gerecht zu werden. Durch die Beobachtung der Regression auf der transformierten Skala, wie in Abbildung 26 dargestellt, gewinnen wir ein besseres Verständnis der Muster und Beziehungen in den Daten, was zu einer verbesserten Identifizierung und Interpretation führt.

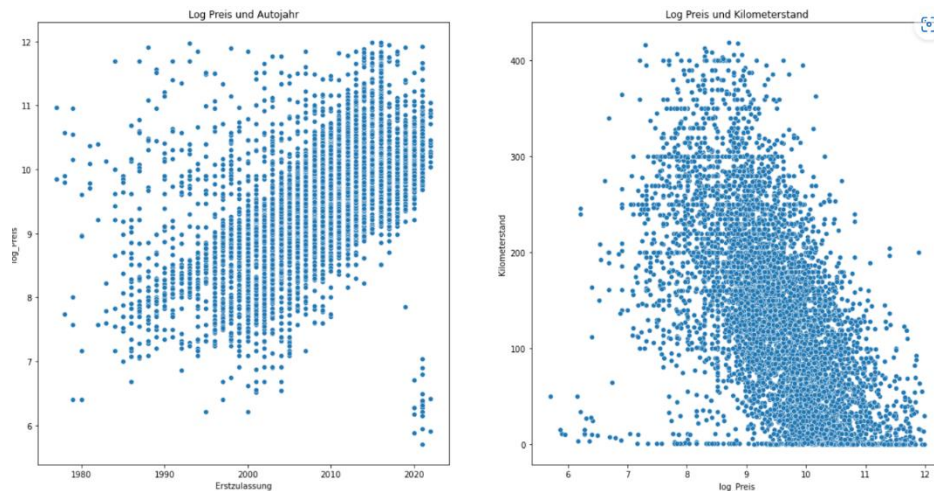


Abbildung 26: Regression

8- Die Kodierungstechnik wird verwendet, um kategoriale Daten in numerisches Format umzuwandeln, da maschinelle Lernalgorithmen in der Regel nicht direkt mit kategorialen Daten umgehen können. Aus diesem Grund habe ich Dummy-Variablen mithilfe der Funktion `pd.get_dummies()` erstellt. Diese Funktion erzeugt binäre Indikatorvariablen für jede eindeutige Kategorie im Datensatz. Durch die Einstellung `drop_first=True` wird eine Kategorie ausgelassen, um Multikollinearitätsprobleme zu vermeiden. Diese Transformation ermöglicht eine effektive Nutzung der kategorialen Daten in maschinellen Lernmodellen.

	Kilometerstand	Erstzulassung	log_Preis	Marke_Abarth	Marke_Alfa Romeo	Marke_Alпина	Marke_Aston Martin	Marke_Audi	Marke_BMW	Marke_Bentley	...	Marke_Trab
0	278.000	1997.0	7.901007	0	0	0	0	0	0	0	0 ...	
1	95.064	2018.0	10.078600	0	0	0	0	0	0	0	0 ...	
2	11.312	2019.0	10.461674	0	0	0	0	0	0	0	0 ...	
3	80.675	2018.0	10.021271	0	0	0	0	0	0	0	0 ...	
4	165.000	2003.0	9.432684	0	0	0	0	0	1	0	0 ...	

Abbildung 27: Kopf des Datensatzes mit Dummy-Variablen

Nach Abschluss dieser Vorverarbeitungsschritte ist der Datensatz nun bereit für die Vorhersagemodelle.

## 3.5 Bewertungsparameter des Modells

Das Regressionsmodell kann anhand der folgenden Parameter evaluiert werden:

### 1- Wurzel des mittleren quadratischen Fehlers (Root Mean Square Error, RMSE):

Die Wurzel des quadratischen Mittelwertfehlers (RMSE) ist eine Bewertungsmetrik, die auch die durchschnittliche Größenordnung des Fehlers misst. Es ist die Quadratwurzel des durchschnittlichen quadrierten Unterschieds zwischen Vorhersage und tatsächlicher Beobachtung. Die RMSE wird mit der folgenden Gleichung berechnet :

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{t=n} (y' - y)^2}$$

Abbildung 28: RMSE-Gleichung

Dabei ist  $y'$  der prognostizierte Wert und  $y$  der wahre Wert.  $n$  ist die Gesamtzahl der Werte in der Testmenge.

### 2- R\_squared

R\_squared ( $R^2$ ) ist eine statistische Metrik, die den Anteil der Varianz der abhängigen Variable erklärt, der durch das Regressionsmodell erklärt wird. Es misst die Güte der Anpassung des Modells an die Datenpunkte und gibt an, wie gut die beobachteten Werte durch die unabhängigen Variablen des Modells erklärt werden können. Ein höherer R\_squared-Wert deutet auf eine bessere Anpassung hin, wobei 1 den besten möglichen Wert darstellt, der angibt, dass das Modell die Daten perfekt erklärt. Ein R\_squared-Wert von 0 bedeutet hingegen, dass das Modell keine Verbesserung gegenüber einer einfachen Durchschnittsprognose bietet.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}}$$

Abbildung 29: R\_squared-Gleichung

Dabei steht  $SS_{residual}$  für die Summe der quadrierten Residuen (Differenz zwischen den beobachteten und vorhergesagten Werten) und  $SS_{total}$  für die Gesamtsumme der quadrierten Abweichungen der beobachteten Werte vom Durchschnittswert.

# Kapitel 4

## Projektanalyse

### 4.1 Experimentelle Ergebnisse und Analyse

Folgende Machine Learning-Klassifikatoren wurden implementiert:

- 1- Lineare Regression
- 2- Random Forest Regressor
- 3- Gradient Boosting Regressor

#### 4.1.1 Lineare Regression

Durch Anwendung einer linearen Gleichung auf die beobachteten Daten versucht die lineare Regression, die Beziehung zwischen zwei Variablen zu veranschaulichen. Es sollte eine unabhängige Variable und eine abhängige Variable geben. Ein Beispiel dafür ist das Gewicht und die Größe einer Person, die linear miteinander zusammenhängen. Daher besteht eine lineare Beziehung zwischen Gewicht und Größe einer Person. Mit zunehmender Größe nimmt auch das Gewicht zu. Es ist keine Voraussetzung, dass eine Variable eine andere verursacht, aber es gibt einige wichtige Zusammenhänge zwischen den beiden Variablen. In solchen Fällen verwenden wir ein Streudiagramm, um die Anzahl der Beziehungen zwischen den Variablen darzustellen. Wenn es keine Korrelation oder Verbindung zwischen den Variablen gibt, zeigt das Streudiagramm keine erkennbaren Muster von Zunahme oder Abnahme. Daher ist die lineare Regression in solchen Fällen für die gegebenen Daten nicht geeignet.

In diesem Projekt wurde die lineare Regression mit der Eigenschaft des Achsenabschnitts trainiert. Zur Bewertung des Modells wurden R\_squared und RMSE Fehler verwendet. Die folgenden Ergebnisse wurden erzielt.

```
# Model Building
from sklearn.linear_model import LinearRegression
lr = LinearRegression()

# Training Model
lr.fit(x_train,y_train)

# Model Summary
y_pred_lr = lr.predict(x_test)

r_squared = r2_score(y_test,y_pred_lr)
rmse = np.sqrt(mean_squared_error(y_test,y_pred_lr))
print("R_squared :",r_squared)
print("RMSE :",rmse)

R_squared : 0.5945180188068379
RMSE : 0.6191185022608608
```

Abbildung 30: Bewertungswerte des linearen Regressionsmodells

### 4.1.2 Random-Forest-Regressor

Normalerweise werden Random Forests oder zufällige Entscheidungswälder für Klassifizierung, Regression und andere Aufgaben verwendet. Dabei werden während des Trainings eine Vielzahl von Entscheidungsbäumen erstellt, und am Ende wird die Klasse ausgegeben, die den Modus der Klassen (Klassifizierung) oder den Durchschnitt der Vorhersagen (Regression) der einzelnen Bäume darstellt. Ein Random Decision Forest korrigiert die Tendenz von Entscheidungsbäumen, ihr Trainingsset zu überanpassen. Obwohl Random Forests im Allgemeinen besser als Entscheidungsbäume sind, sind sie nicht so genau wie Gradient Boosted Trees. Allerdings werden sie von den Eigenschaften der Daten beeinflusst.

In diesem Projekt wurde der Random Forest Regressor mit der Eigenschaft des Achsenabschnitts trainiert. Zur Bewertung des Modells wurden R\_squared und RMSE Fehler verwendet. Die folgenden Ergebnisse wurden erzielt.

```
: # Model Building
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()

# Training Model
rf.fit(x_train,y_train)

# Model Summary
y_pred_rf = rf.predict(x_test)

r_squared = r2_score(y_test,y_pred_rf)
rmse = np.sqrt(mean_squared_error(y_test,y_pred_rf))
print("R_squared :",r_squared)
print("RMSE :",rmse)

R_squared : 0.6664186652559813
RMSE : 0.5615505722709605
```

Abbildung 31: Bewertungswerte des Random Forest Regressor Modells

### 4.1.2 Gradient Boosting Regressor

Der Gradient Boosting Regressor ist ein leistungsstarker Regressionsalgorithmus, der für die Vorhersage von kontinuierlichen numerischen Werten eingesetzt wird. Er basiert auf der Idee, iterativ schwache Regressionsmodelle zu erstellen und zu verbessern, indem die Fehler des vorherigen Modells analysiert und minimiert werden. Durch die Kombination mehrerer schwacher Modelle kann der Gradient Boosting Regressor komplexe nichtlineare Zusammenhänge in den Daten erfassen und präzise Vorhersagen liefern. Er ist bekannt für seine Fähigkeit, mit hoher Genauigkeit auf verschiedenste Arten von Daten und Problemstellungen zu arbeiten. Der Gradient Boosting Regressor kann jedoch auch

empfindlich auf Überanpassung reagieren, weshalb eine sorgfältige Parameterabstimmung und Modellvalidierung erforderlich ist, um optimale Ergebnisse zu erzielen.

In diesem Projekt wurde der Gradient Boosting Regressor mit der Eigenschaft des Achsenabschnitts trainiert. Zur Bewertung des Modells wurden R\_squared und RMSE Fehler verwendet. Die folgenden Ergebnisse wurden erzielt.

```
: # Model Building
from sklearn.ensemble import GradientBoostingRegressor
gbt = GradientBoostingRegressor()

# Training Model
gbt.fit(x_train,y_train)

# Model Summary
y_pred_gbt = gbt.predict(x_test)

r_squared = r2_score(y_test,y_pred_gbt)
rmse = np.sqrt(mean_squared_error(y_test,y_pred_gbt))
print("R_squared :",r_squared)
print("RMSE :",rmse)

R_squared : 0.6586694899970147
RMSE : 0.5680356042270435
```

Abbildung 32: Bewertungswerte des Gradient Boosting Regressor Modells

Um die Genauigkeit der Vorhersagen des Modells zu bewerten, wurde ein manueller Vergleich zwischen den vorhergesagten und tatsächlichen Preisen durchgeführt. Zur Erleichterung dieser Analyse wurde ein DataFrame namens df\_ev erstellt. Die vorhergesagten Preise wurden erzeugt, indem die vom Modell vorhergesagten Werte exponentiert wurden. Die tatsächlichen Preise wurden aus einem separaten Datensatz gewonnen und zum DataFrame hinzugefügt. Um die Diskrepanzen zu bewerten, wurden die Residuen als Variance zwischen den tatsächlichen und vorhergesagten Preisen berechnet. Darüber hinaus wurde der prozentuale Unterschied zwischen den beiden Werten ermittelt. Um die Ergebnisse effektiver zu präsentieren, wurde der DataFrame basierend auf dem prozentualen Unterschied sortiert, und die letzten 30 Zeilen wurden angezeigt.

```
df_ev = pd.DataFrame(np.exp(y_pred_rf), columns=['Predicted Price'])

y_test = y_test.reset_index(drop=True)
df_ev['Actual Price'] = np.exp(y_test)

df_ev['Residual'] = df_ev['Actual Price'] - df_ev['Predicted Price']
df_ev['Difference%'] = np.absolute(df_ev['Residual']/df_ev['Actual Price']*100)

pd.set_option('display.float_format', lambda x: '%.2f' % x)
df_ev.sort_values(by=['Difference%'])

df_ev.tail(30)
```

	Predicted Price	Actual Price	Residual	Difference%
1234	2897.82	4400.00	1502.18	34.14
1235	11937.41	9200.00	-2737.41	29.75
1236	3283.37	8900.00	5616.63	63.11
1237	10025.21	7100.00	-2925.21	41.20
1238	3889.81	4000.00	110.19	2.75
1239	11302.53	9800.00	-1502.53	15.33
1240	29934.94	32000.00	2065.06	6.45
1241	13275.70	12700.00	-575.70	4.53
1242	6569.21	6800.00	230.79	3.39
1243	36439.36	26800.00	-9639.36	36.99
1244	31915.08	39500.00	7584.92	19.20
1245	9030.54	9000.00	-30.54	0.34

Abbildung 33: Vorhersageauswertungstabelle

## 4.2 Ergebnisvergleich

Der Vergleich aller Experimente wird in Tabelle 2 dargestellt.

Nummer	Algorithmus	Genauigkeit	RMSE
1	Lineare Regression	0.59	0.619
2	Random-Forest-Regressor	0.66	0.561
3	Gradient Boosting Regressor	0.65	0.568

*Tabelle 2: Ergebnisvergleich*

Basierend auf den Bewertungsparametern schnitt der Random Forest Regressor mit der höchsten Genauigkeit und dem geringsten Fehler bei allen drei Bewertungsparametern besser ab als die anderen beiden Algorithmen. Der Gradient Boosting Regressor belegte in Bezug auf die Genauigkeit den zweiten Platz und erreichte eine Genauigkeit von 65% sowie einen ähnlichen Fehlerparameter wie der Random Forest Regressor. Der Linear Regression Algorithmus war am ungenauesten, mit einer Genauigkeit von 59% und dem höchsten Fehlerwert.

# Kapitel 5

## 5.1 Schlussfolgerung

Durch den Einsatz von Data-Mining- und maschinellen Lernansätzen hat dieses Projekt einen skalierbaren Rahmen für die Preisvorhersage von Gebrauchtwagen in Deutschland vorgeschlagen. Zur Sammlung der Benchmark-Daten wurde die Webseite [12gebrauchtwagen.de](https://12gebrauchtwagen.de) gescraped. Ein effizientes maschinelles Lernmodell wurde erstellt, indem drei Machine-Learning-Regressoren - der Random Forest Regressor, die Lineare Regression und der Gradient Boosting Regressor - trainiert, getestet und bewertet wurden. Als Resultat der Vorverarbeitung und Transformation hat sich der Random Forest Regressor mit einer Genauigkeit von 66% als Spitzenreiter erwiesen, gefolgt vom Gradient Boosting Regressor mit 65%. Jedes Experiment wurde in Echtzeit in der Jupyter-Notebook-Umgebung durchgeführt.

## 5.2 Empfehlungen und zukünftige Arbeiten

In der Zukunft plane ich, mehr Daten mit verschiedenen Web-Scraping-Techniken zu sammeln und tiefe Lernklassifikatoren zu testen. Algorithmen wie Quantile Regression, ANN und SVM werden in Erwägung gezogen, um eine höhere Genauigkeit zu erreichen. Anschließend habe ich vor, mein intelligentes Modell in web- und mobilbasierte Anwendungen zu integrieren, die für die breite Öffentlichkeit zugänglich sind.



# Bibliographie

'Künstliche Intelligenz – Das richtige Vorgehen bei Machine Learning'. Bitfactory - Partner für App Entwicklung und digitale Produkte, <https://www.bitfactory.io/de/blog/kuenstliche-intelligenz-das-richtige-vorgehen-bei-machine-learning/>.

D, Erika. (2019) 'Looking at R-Squared'. Medium, <https://medium.com/@erika.dauria/looking-at-r-squared-721252709098>.

Fehlermaße: Wie Sie die Güte Ihrer Forecasts auswerten. 6 July 2021, <https://www.jedox.com/de/blog/fehlermasze-guete-von-forecasts-ermitteln/>.

Wuttke, Laurenz.(2022) 'Machine Learning: Definition, Algorithmen, Methoden und Beispiele'. datasolut GmbH, <https://datasolut.com/was-ist-machine-learning/>.

Germany Used Car Market Size & Share Analysis - Industry Research Report - Growth Trends. <https://www.mordorintelligence.com/industry-reports/germany-used-car-market>. Accessed 26 May 2023.

Kuiper, S. (2008). Introduction to Multiple Regression: How Much Is Your Car Worth? Journal of Statistics Education. doi:10.1080/10691898.2008.11889579

Cerrottia, M. (2019). Unsupervised machine learning in atomistic simulations, between predictions and understanding. 150-155.

Jian Da Wu, C.-c. H.-C. (2017). "An expert system of price forecasting for used cars using adaptive. ELSEVEIR, 16, 417-957.

Kuiper, S. (2008). Introduction to Multiple Regression: How Much Is Your Car Worth? Journal of Statistics Education. doi:10.1080/10691898.2008.11889579

Swaminathan, S. (2018, March 15). Logistic regression - detailed overview. (towards Data science) Retrieved October 27, 2020, from <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>

Nabarun Pal, P. A. (2018). How much is my car worth? A methodology for predicting used cars prices using Random Forest. Future of Information and Communications Conference (FICC) 2018, 1-6.

Monburinon, N., Chertchom, P., Kaewkiriya, T., Rungpheung, S., Buya, S., & Boonpou, P. (2018). Prediction of Prices for Used Car by Using Regression Models. 5th International Conference on Business and Industrial Research (ICBIR), (pp. 115-119). Bangkok.

Noor, K., & Jan, S. (2017). Vehicle Price Prediction System using Machine Learning Techniques. International Journal of Computer Applications, 27-31.

Pudaruth, S. (2014). Predicting the Price of Used Cars using Machine Learning. International Journal of Information & Computation Technology, 754-764.

Alle Gebrauchtwagen-Angebote Im Netz Vergleichen – 12Gebrauchtwagen.De.  
<https://www.12gebrauchtwagen.de/>.