



**TrendAuto
2030 plus**

Transformationsnetzwerk für die
Fahrzeug- und Zulieferindustrie in der
Region Aachen-Köln-Bonn-Gummersbach

Gefördert durch:



Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

September 2024



KI in der Entwicklung & Produktion

Ein Schritt-für-Schritt-Leitfaden zur Programmierung Künstlicher Intelligenz

Prof. Dr. Margot Ruschitzka
Martin de Fries

Marcus Irmer
Alexander Nüßgen



Transformationsnetzwerk für die Fahrzeug- und Zulieferindustrie in der Region Aachen-Köln-Bonn-Gummersbach

Was ist TrendAuto2030plus?

Wir sind das „**Tr**ansformationsnetzwerk für eine **e**lektrische, **n**achhaltige und **d**igitale **Automobilindustrie2030plus**“ (kurz **TrendAuto2030plus**).

Wir sind Partner der Automobil- und Zulieferbranche und unterstützen Unternehmen in der Region Aachen, Bonn, Köln und Gummersbach bei der Identifizierung ihrer Transformationsbedarfe.

Wir bestärken Unternehmen, Potenziale zu entfalten und solide Transformationsstrategien zu entwickeln und sorgen so nachhaltig für zukunftssichere Beschäftigung.

Wir bieten praxisnahe Lösungen, fördern Wissenstransfer und Kollaboration, um Unternehmen für den globalen und lokalen Wandel vorzubereiten.

Projektkonsortium

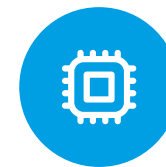
**Technology
Arts Sciences
TH Köln**



Strategie

Das Projekt konkretisiert sich in vier Handlungsfeldern:

Vor dem Hintergrund von Digitalisierung und Nachhaltigkeitswende ist die Automotive Industry zur Transformation angehalten. Wie kann ich mein Unternehmen wandlungsfähiger gestalten? Wie kann ich Potenziale des Wandels nutzen? Wie entwickelt sich die Branche in der Region? Unsere Teams **Agilität und Transformationspotenziale** und **Strategieentwicklung-2030plus** beschäftigen sich intensiv mit diesen Fragen und entwickeln eine regionale Strategie für die Zukunft.



Technologie

Technologien sind oft der Schlüssel zu Innovationen und erfolgreicher Transformation. Daher bieten unsere Teams **Benchmarking und Good Practices, Entwicklungssystematik und kollaboratives Arbeiten** sowie **Produktionssystematik und Industrie 4.0** ihre Expertise in diesem Bereich. Sie befassen sich mit aktuellen Trends wie Elektromobilität, Digitalisierung, Künstlicher Intelligenz oder Kreislaufwirtschaft.



Geschäftsmodell

Nicht zuletzt das geplante Verbrenner-Aus könnte dazu führen, dass einige Geschäftsmodelle in der Branche vollständig entfallen. Unser Team **Neue Technologien und Geschäftsmodelle** bietet umfassende Unterstützung bei der (Weiter-)entwicklung von Geschäftsmodellen. Insbesondere legt das Team dabei einen Fokus auf technologische Kompetenzen sowie relevante Technologietrends wie Künstliche Intelligenz und Wasserstoff.



Qualifikation

Mit Blick auf aktuelle Herausforderungen in der Autoindustrie ermitteln wir gemeinsam mit Unternehmen die resultierenden Weiterbildungsbedarfe. Unser Team **Weiterbildungsbedarfe und Qualifizierungsformate** teilt Best Practices für Qualifizierungsmaßnahmen in Unternehmen, fördert die Vernetzung unterschiedlicher Akteure im Weiterbildungsbereich und informiert über existierende Qualifizierungsangebote. Ergänzende Weiterbildungsformate werden prototypisch entwickelt.

Kontakt

Transformationsnetzwerk TrendAuto2030plus
info@trendauto2030.de
Gustav-Heinemann-Ufer 54
50968 Köln



trendauto2030plus.de

Autoren



Margot Ruschitzka

Prof. Dr. rer. nat. Dr. h.c.

CAD CAM Center Cologne (4C)
Technische Hochschule Köln
Betzdorfer Straße 2

Kontakt



Marcus Irmer

Mechatronik, M.Sc.

CAD CAM Center Cologne (4C)
Technische Hochschule Köln
Betzdorfer Straße 2

Kontakt



Martin de Fries

Mechatronik, M.Sc.

CAD CAM Center Cologne (4C)
Technische Hochschule Köln
Betzdorfer Straße 2

Kontakt



Alexander Nüßgen

Automotive Engineering, M.Sc.

CAD CAM Center Cologne (4C)
Technische Hochschule Köln
Betzdorfer Straße 2

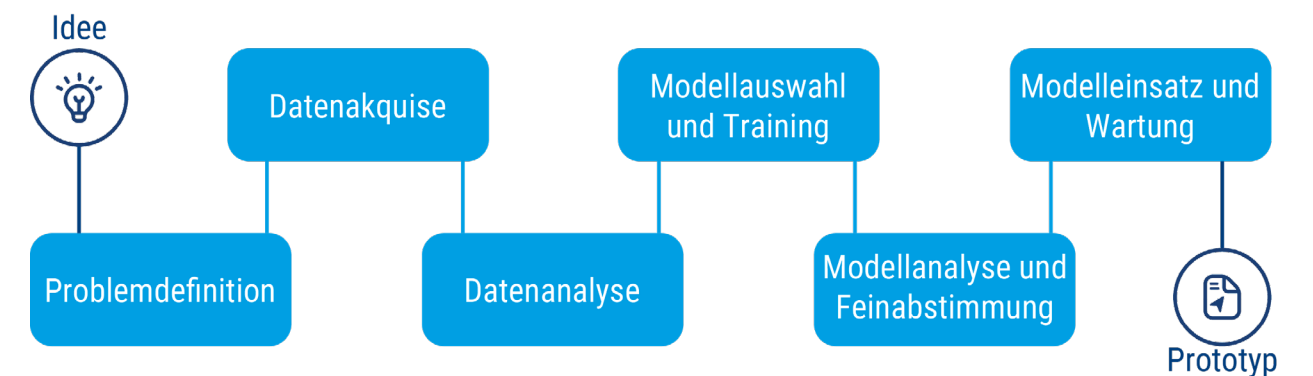
Kontakt

Vorwort

Künstliche Intelligenz (KI) hat sich in den letzten Jahrzehnten als eines der faszinierendsten und umwälzendsten Felder in der Technologie und Wissenschaft etabliert. Von einfachen algorithmischen Verfahren bis hin zu komplexen neuronalen Netzwerken, die menschenähnliche Lernfähigkeiten besitzen, erstreckt sich die Bandbreite dessen, was unter dem Begriff KI summiert wird. Die signifikanten Fortschritte in der Rechenleistung, Datenspeicherung und -analyse haben die Grundlage für diese bemerkenswerte Ära der Innovation bereitet. Heute sind die Anwendungsgebiete von KI allgegenwärtig - von der Verbesserung der Kundeninteraktion in der Geschäftswelt, der Erhöhung der Effizienz in der Produktion, dem Vorantreiben wissenschaftlicher Forschung bis hin zur Verbesserung der Lebensqualität durch personalisierte Dienstleistungen.

Das Ziel dieses Artikels ist es, den Leserinnen und Lesern einen fundierten und schrittweisen Einblick in die Entwicklung einer Künstlichen Intelligenz zu gewähren sowie eine praxisnahe und detaillierte Schritt-für-Schritt-Anleitung zur Entwicklung einer Künstlichen Intelligenz (KI) zur Verfügung zu stellen. Im Zentrum steht dabei der gesamte Prozess von der initialen Idee bis hin zur erfolgreichen Implementierung und Wartung eines KI-Modells. Dies umfasst die grundlegenden Phasen der Problemdefinition, Datenakquise, Datenanalyse, Auswahl und Training des Modells, Analyse und Feinabstimmung des Modells sowie schließlich den Modelleinsatz und die fortlaufende Wartung. Die einzelnen Phasen sind dabei inklusive exemplarische Implementierung eines einfachen KI-Modells in den folgenden Kapiteln beschrieben.

Unser besonderer Dank geht an die Hans Berg GmbH & Co. KG, insbesondere den Leiter Musterbau und Verfahrensentwicklung Thomas Woitzik, für die Bereitstellung relevanter Daten.



Inhalt

Problemdefinition	8
1.1 Problemerkennung	9
1.2 Problemverständnis	9
1.3 Definition des Problems	10
1.4 Beispiel	10
Datenakquise	13
2.1 Identifikation der Datenanforderungen	14
2.2 Datenerhebung	14
2.3 Datenprüfung	14
2.4 Datenintegration	15
2.5 Beispiel	15
Datenanalyse	17
3.1 Datenexploration	18
3.2 Datenbereinigung	18
3.3 Datenanreicherung und -transformation	19
3.4 Beispiel	19
Modellauswahl und Training	22
4.1 Datenvorbereitung	23
4.2 Modellauswahl	23
4.3 Training des Modells	24
4.4 Beispiel	24
Modellanalyse und Feinabstimmung	26
5.1 Leistungsbewertung	27
5.2 Modelloptimierung	28
5.3 Beispiel	29
Modelleinsatz und Wartung	32
6.1 Implementierung des Modells	33
6.2 Überwachung	33
6.3 Aktualisierung und Anpassung	34
6.4 Dokumentation und Compliance	34
6.5 Beispiel	34

1

Problemdefinition

Identifizieren Sie das Problem, das mit KI gelöst werden soll.

Die erste Phase der Problemdefinition legt den Grundstein für die Entwicklung einer KI, indem sie präzise den Rahmen absteckt, innerhalb dessen die KI-Applikation Lösungen bieten soll. Ein gründliches Verständnis des zu lösenden Problems ist entscheidend für den Erfolg des gesamten Projekts.

Die richtige Identifizierung, das Verständnis und die genaue Definition eines Problems, das mittels KI gelöst werden soll, stellen den Ausgangspunkt für die Entwicklung jeder KI-Lösung dar. Diese Phase ist daher entscheidend, da sie die Grundlage für alle folgenden Entwicklungsschritte bildet. Die Phase der Problemdefinition lässt sich dabei in die folgenden Schritte gliedern:

Erkennen – Verstehen – Definieren

1.1 Problemerkennung

- **Informationsbeschaffung:** Sammeln Sie grundlegende Informationen über das Gebiet oder den Bereich, in dem das Problem auftritt. Verstehen Sie Kontext, Dynamik und Theorie des Bereichs.
- **Stakeholder-Analyse:** Bestimmen Sie, wer von dem Problem und seiner Lösung betroffen ist. Identifizieren Sie die Hauptakteure, ihre Bedürfnisse und ihre Erwartungen an die KI-Lösung.
- **Zielklärung:** Verstehen Sie, was mit der Lösung erreicht werden soll. Klare, messbare Ziele helfen, den Erfolg der KI-Lösung später zu bewerten.

1.2 Problemverständnis

- **Problemzerlegung:** Zerlegen Sie das Gesamtproblem in kleinere, handhabbare Teile. Dies kann helfen, komplexe Probleme besser zu verstehen und zu bearbeiten.
- **Priorisierung:** Bewerten und priorisieren Sie die identifizierten Teilprobleme basierend auf Faktoren wie ihrem Einfluss, ihrer Dringlichkeit und den verfügbaren Ressourcen.
- **Ursachenanalyse:** Identifizieren Sie die Ursachen des Problems. Techniken wie das 5-Why-Verfahren können dabei helfen, die Wurzelursachen tiefgehend zu verstehen.

1.3 Definition des Problems

- **Problemformulierung:** Formulieren Sie das Problem und die daraus resultierende Aufgabe so genau wie möglich, z.B. Klassifizierung, Clustering oder Regression. Eine gute Problemformulierung beinhaltet eine klare Beschreibung des aktuellen Zustands sowie des Ziels, das erreicht werden soll.
- **Machbarkeitsanalyse:** Bewerten Sie, ob KI eine geeignete Lösung für das Problem ist. Dies umfasst die Sichtung der verfügbaren Daten, der technologischen Anforderungen und möglicher Einschränkungen.
- **Spezifizierung der Anforderungen:** Legen Sie die spezifischen Anforderungen fest, die die KI-Lösung erfüllen muss. Dazu gehören funktionale Anforderungen, Leistungsziele und ethische Überlegungen.
- **Zielformulierung:** Definieren Sie klare, messbare Ziele für Ihre KI-Lösung. Diese Ziele sollten auf das übergeordnete Problem abgestimmt sein und als Grundlage für die spätere Bewertung der Modellleistung dienen.

Durch die genaue Befolgung dieser Schritte wird eine solide Grundlage für die Entwicklung effektiver und zielgerichteter KI-Lösungen, die tatsächlich in der Lage sind, spezifische Probleme zu lösen, gelegt. Wichtig ist dabei, den Fokus sowohl auf technische Machbarkeit als auch auf die Bedürfnisse und Erwartungen der Stakeholder zu legen.

1.4 Beispiel

Klassifikation, Clustering und Regression

Ein Großteil von Aufgabenstellungen lässt sich in Klassifikations-, Clusters- oder Regressionsaufgaben einteilen. Klassifikationsaufgaben werden häufig durch Überwachtes Lernen gelöst und zielen darauf ab, anhand gegebener Informationen Objekte in diskrete, vordefinierte Kategorien oder Klassen einzuteilen. Typische Beispiele sind

- Klassifikation von E-Mails als „Spam“ oder „Nicht-Spam“.
- Klassifikation von Objekten in Bildern, zum Beispiel das Unterscheiden zwischen Menschen, Hindernissen, Umgebung oder vieles mehr zur Entwicklung autonomer Funktionen.

- Klassifikation von Produkten als „fehlerhaft“ oder „einwandfrei“ zur Qualitätsüberwachung und Prozessoptimierung.

Daneben gehören Clustering-Aufgaben zum unüberwachten Lernen und bestehen darin, gegebene Datenpunkte in Gruppen oder Cluster zu segmentieren, ohne dass es vorher festgelegte Kategorien gibt. Beispiele dafür sind

- Identifizierung von Gruppen ähnlicher Kunden basierend auf Kaufverhalten, Demografie oder Interessen zur Kundensegmentierung
- Identifikation von Produktionsmustern, die zu ähnlichen Fehlerarten führen, um Wartungspläne zu optimieren
- Identifizierung und Gruppierung von ähnlichen Informationen in großen Datenbanken zum intelligenten Informationsmanagement

Regressionsaufgaben hingegen befassen sich mit der Vorhersage kontinuierlicher Zielvariablen, anstelle diskreter Klassen oder Gruppen. Beispiele hierfür sind

- Schätzung der Ankunftszeit basierend auf Verkehrsdaten.
- Vorhersage von Umsätzen und Gewinnen zur bestmöglichen Unternehmenssteuerung
- Schätzung der Restlebensdauer von Maschinen oder Werkzeugen basierend auf deren Nutzung und Belastung zur Optimierung der Produktion

Binäre Klassifizierungsaufgabe zur optimalen Klassifizierung eines Bauteils

Ein anschauliches Beispiel für die praktische Umsetzung einer Klassifikationsaufgabe zeigt eine Herausforderung der Hans Berg GmbH & Co. KG, welche prototypisch mittels eines KI-Ansatzes gelöst werden konnte. Die identifizierte Herausforderung dient als gutes Beispiel, um die Potentiale von KI in der Fertigungsindustrie zu demonstrieren. Im Zentrum steht der Prozess des Kondensatorentladungsschweißens zur Fertigung eines Bauteils, welches in Abbildung 2 exemplarisch dargestellt ist. Die Firma Hans Berg nutzt dazu flexible Fertigungszellen mit einem umfangreichen Informationssystem zur Sammlung von Messdaten für die Prozessüberwachung und die Gütebewertung der Schweißung. Der Schweißprozess selbst, basierend auf der schlagartigen Entladung von Kondensatoren, hängt von den Einstellparametern Kraft und Energie ab, deren Messwerte entscheidende Gütekriterien für die Qualität des fertigen Bauteils sind. Trotz eines etablierten Prozesses zur

1 Problemdefinition

maschinellen Bewertung der Güte der Schweißung, stand die Firma Hans Berg vor der Herausforderung, dass die Maschine viele Bauteile produziert, die als nicht in Ordnung (NOK) klassifiziert wurden, obwohl eine metallographische Prüfung sie als in Ordnung (OK) bestätigte, da starre Toleranzgrenzen zur Bewertung verwendet wurden. Dieses Problem weist auf eine Diskrepanz zwischen der maschinellen Bewertung und der tatsächlichen Teilequalität



Abbildung 2: Bauteil „Haubenaufsteller“ mit Schweißung aus dem Automotive-Segment der Firma Hans Berg

hin, was zu unnötigem Material- und Ressourcenverlust führt. Die Aufgabenstellung besteht daher darin, den Ausschuss durch eine optimierte Bewertung der Teilegüte zu reduzieren. Der Kern der Herausforderung liegt somit in der Verbesserung des bestehenden Bewertungsprozesses, um eine präzisere Unterscheidung zwischen OK- und NOK-Teilen zu ermöglichen.

Die Schritte zur Entwicklung eines KI-Ansatzes zur Lösung dieser Aufgabenstellung können werkzeuggestützt durchgeführt werden. Hier wurde dafür die Programmiersprache Python sowie die Programmierumgebung Colab genutzt. Python verfügt über eine Vielzahl an Funktionalitäten und Bibliotheken zum automatisierten Erstellen, Trainieren und Evaluieren von KI-Modellen. Colab bietet darüber hinaus den Vorteil, dass es eine einfache Umgebung darstellt, welche browserbasiert ohne jegliche Installation ausgeführt werden kann. Die Implementierung der oben beschriebenen Aufgabenstellung kann hier eingesehen und ganz einfach selbst erprobt werden:



2

Datenakquise

Sammeln und erfassen Sie relevante Daten, die zur Entwicklung des KI-Modells benötigt werden.

KI-Systeme benötigen Daten, um zu lernen. Die Beschaffung von qualitativ hochwertigen, relevanten und vielfältigen Daten ist daher entscheidend für die Entwicklung einer leistungsfähigen KI. Der Prozess der Datenbeschaffung wird auch Datenakquise genannt und gliedert sich in die folgenden Schritte:

Identifizieren – Erheben – Prüfen – Integrieren

2.1 Identifikation der Datenanforderungen

- **Bedarfsanalyse:** Identifizieren Sie, welche Daten rund um das Problem bereits erhoben wurden und welche Daten zusätzlich benötigt werden könnten, um das Problem effektiv zu lösen. Berücksichtigen Sie die Merkmale wie Datentyp (z.B. numerisch, kategorisch), Datenstruktur (z.B. tabellarisch, textbasiert, bildbasiert) und spezifische Attribute der Daten.
- **Datenquellenidentifikation:** Bestimmen Sie potenzielle Datenquellen und bewerten Sie ihre Qualität sowie Verfügbarkeit. Daten können aus internen Quellen (z.B. Unternehmensdatenbanken) oder externen Quellen (z.B. öffentlich zugängliche Datensätze, Datenkauf, soziale Medien) bezogen werden.

2.2 Datenerhebung

- **Datensammlung:** Sammeln Sie die erforderlichen Daten aus den identifizierten Quellen.
- **Datenerfassung:** In einigen Fällen, besonders wenn keine existierenden Datenquellen verfügbar sind, kann es notwendig sein, Daten durch Experimente, Messungen oder Umfragen selbst zu erfassen.

2.3 Datenprüfung

- **Qualitätsbewertung:** Beurteilen Sie die Qualität und Repräsentativität der gesammelten Daten. Überprüfen Sie, wie die Daten vorliegen (Dateiformat, Datentyp, ...) und wie viele Daten vorhanden sind. Kontrollieren Sie zudem, ob die Daten genau, vollständig und relevant sind.

- **Ethische Überprüfung:** Überprüfen Sie die ethischen Aspekte der Datenverwendung, vor allem im Hinblick auf Datenschutz, Persönlichkeitsrechte und die Einhaltung zugehöriger Vorschriften.

2.4 Datenintegration

- **Datenzusammenführung:** Wenn Daten aus verschiedenen Quellen stammen, ist es oft erforderlich, diese zu kombinieren. Die Datenintegration umfasst die Vereinheitlichung von Datenformaten, das Abgleichen von Datensätzen und das Zusammenführen in eine gemeinsame Datenbank oder ein Datenset.
- **Datensicherung:** Implementieren Sie geeignete Maßnahmen zum Schutz der Daten vor Verlust und unbefugtem Zugriff. Entscheiden Sie, wo und wie die Daten gespeichert werden sollen. Dies können lokale Serversysteme sein, cloudbasierte Speicherlösungen oder spezialisierte Datenbanken, abhängig von den Anforderungen an die Sicherheit, Zugänglichkeit und Skalierbarkeit.

Die Datenakquise legt die Grundlage für die Trainings- und Testdaten, die für das Lernen und die Bewertung von KI-Systemen erforderlich sind.

2.5 Beispiel

Die Maschine von Hans Berg zur Fertigung des Bauteils besitzt bereits ein Informationssystem zur Sammlung von Messdaten. So werden automatisch Messwerte wie Kraft, Energie, Strom, Stromanstiegszeit, Schweißzeit und AbSENKweg bei der Fertigung eines Bauteils aufgenommen. Exemplarisch wurden 82 dieser Datensätze gesammelt und in einer Excel-Datei gespeichert. Diese Daten wurden eingehend überprüft und durch eine umfassende metallographische Prüfung zur wahren Klassifizierung der Teile als OK oder NOK ergänzt. Ein Auszug der Daten ist in Abbildung 3 zu sehen.

Teilekontrolle [mm]	Kraft [kN]	Energie [kJ]	Strom [kA]	tp [ms]	th [ms]	Absenkweg [mm]	Bewertung
-0,0160	14,8000	5,4000	55,7000	7,2000	14,0000	0,7060	OK
-0,0100	19,4000	5,5000	58,8000	7,5000	13,7000	0,7100	OK
-0,0480	19,4000	4,9000	55,4000	7,3000	13,5000	0,6180	NOK
-0,0400	21,2000	4,2000	51,3000	7,1000	14,1000	0,4880	NOK
-0,0540	21,3000	4,2000	51,5000	7,7000	14,2000	0,4920	NOK
-0,0260	21,2000	4,2000	51,2000	7,4000	14,2000	0,5300	NOK
-0,0300	21,3000	4,2000	51,9000	7,6000	14,5000	0,5680	NOK
-0,0080	21,1000	4,2000	51,7000	7,8000	14,6000	0,5520	NOK
0,0080	23,1000	3,7000	48,7000	7,6000	14,6000	0,4760	NOK
-0,1600	23,0000	3,6000	48,8000	7,8000	15,1000	0,4900	NOK
0,0580	13,0000	5,9000	59,1000	7,4000	13,4000	0,8220	NOK
0,0520	13,0000	5,9000	58,8000	7,1000	13,2000	0,8160	NOK
0,0580	12,9000	5,9000	59,0000	7,2000	13,1000	0,8240	NOK
0,0600	13,0000	5,9000	59,1000	7,2000	13,0000	0,8120	NOK
0,0300	13,0000	5,9000	58,9000	7,2000	12,9000	0,8060	NOK
0,0240	13,0000	5,9000	59,4000	6,9000	13,0000	0,8220	NOK

Abbildung 3: Auszug der während der Fertigung des Bauteils gesammelten Daten inkl. Bewertung des Bauteils

3

Datenanalyse

Bereiten Sie die gesammelten Daten vor und analysieren Sie diese.

Die im vorherigen Schritt gesammelten Daten müssen analysiert, bereinigt und vorverarbeitet werden, um z.B. Ungenauigkeiten, Fehler und Irrelevanzen zu entfernen sowie ein tieferes Systemverständnis zu erlangen. Dies ist Gegenstand der nachfolgenden Datenanalyse. Sie gewährt zudem tiefgehende Einblicke in die gesammelten Daten und verbessert das Problemverständnis. Dies bildet die Grundlage für effiziente Modellierung und präzise Vorhersagen. Die Datenanalyse umfasst mehrere Schritte, die im Folgenden beschrieben werden:

Explorieren – Bereinigen – Transformieren

3.1 Datenexploration

- **Verteilungsanalyse:** Analysieren Sie die Verteilung der Daten, um Anomalien, Muster und mögliche Verzerrungen zu erkennen. Histogramme, Box-Plots und Quantil-Quantil-Diagramme sind beispielsweise Werkzeuge, die in diesem Schritt häufig verwendet werden.
- **Deskriptive Statistik:** Berechnen Sie grundlegende statistische Maßzahlen wie Mittelwert, Median, Modus, Standardabweichung, Minima und Maxima, um ein grundlegendes Verständnis der Datensätze zu gewinnen.
- **Korrelationsanalyse:** Untersuchen Sie die Beziehungen zwischen Variablen. Korrelationskoeffizienten und Scatterplots können Aufschluss darüber geben, welche Variablen miteinander einhergehen und potenzielle Kandidaten für weitere Analysen darstellen.

3.2 Datenbereinigung

- **Behandlung fehlender Werte:** Identifizieren und behandeln Sie fehlende Daten, entweder durch Entfernen der entsprechenden Datensätze, das Ersetzen fehlender Werte mit geeigneten Imputationsmethoden oder durch Modellierung der fehlenden Daten als eine eigene Kategorie, falls angebracht.
- **Identifikation und Korrektur von Fehlern:** Identifizieren Sie Fehler oder Inkonsistenzen in den Daten und korrigieren oder entfernen Sie diese. Dies kann fehlerhafte Eingaben, falsch zugeordnete Kategorien oder nicht plausible Werte umfassen.

- **Entfernung von Ausreißern:** Identifizieren Sie Datenpunkte, die stark von den anderen Beobachtungen abweichen. Entscheiden Sie mit Vorsicht, ob diese entfernt oder gesondert analysiert werden sollen, basierend auf dem Kontext und Zweck Ihrer Analyse.
- **Beseitigung von Duplikaten:** Identifizieren und beseitigen Sie Duplikate innerhalb der Daten, um die Datenqualität zu verbessern und sicherzustellen, dass die Analyse- und Modellergebnisse nicht durch redundante Informationen verzerrt werden.

3.3 Datenanreicherung und -transformation

- **Datenformatierung:** Sofern die Daten noch nicht in einer einheitlichen Form vorliegen, die für eine KI lesbar ist, konvertieren Sie die Daten entsprechend, z.B. Text in numerische Daten.
- **Feature Engineering:** Entwickeln Sie neue Merkmale (Features) aus den vorhandenen Daten, um Ihr Verständnis zu verbessern oder um spezifische Hypothesen zu testen. Dieser Schritt kann das Zusammenfassen mehrerer Variablen, die Berechnung von Verhältnissen oder die Anwendung von Domänenwissen umfassen.
- **Normalisierung und Skalierung:** Normieren und skalieren Sie gegebenenfalls die Daten so, dass sie von den verwendeten Modellen besser verarbeitet werden können. Methoden wie Min-Max-Normalisierung oder Standardisierung werden oft verwendet, besonders bei Modellen, die sensitiv auf die Eingabeskala reagieren, wie z.B. Neuronale Netzwerke.

Durch sorgfältige Anwendung dieser Schritte wird sichergestellt, dass die Daten richtig für das Training von KI-Modellen vorbereitet sind, was einen entscheidenden Einfluss auf deren Leistungsfähigkeit und Genauigkeit besitzt.

3.4 Beispiel

Die zuvor gesammelten Daten müssen nun aufbereitet werden, damit sie von einem KI-System verarbeitet werden können. Hierbei wurden die Daten zunächst aus der Excel-Datei geladen und einer ersten Überprüfung unterzogen. Dabei wurde unter anderem die Bewertung (Text: „OK“ bzw. „NOK“) eines Bauteils in ein für den KI-Ansatz lesbares Format als 1 („OK“) bzw. 0 („NOK“) konvertiert. Anschließend können die Daten analysiert werden. In Abbildung 4 ist ein Auszug der Analyseergebnisse zu sehen. Dort sind die Bewertungen der Bauteile in Abhängigkeit der Kraft und des Stroms dargestellt. Anhand

dessen lässt sich beispielhaft im mittleren Bereich der Abbildung ein Prozessfenster erkennen lässt, dass die Grenzen definiert, innerhalb derer der Prozess ablaufen muss, um ein fehlerfreies Bauteil zu produzieren.

Die Datenanalyse kann werkzeuggestützt durchgeführt werden. Der folgende Code zeigt die dafür wesentlichen programmietechnischen Schritte. So wird zunächst die Bibliothek pandas geladen. Mit `pd.read_excel(file_path, engine='openpyxl')` werden dann die Daten aus einer Excel-Datei geladen und mit `data.head()` werden die ersten Zeilen der Daten angezeigt, um einen Überblick zu bekommen. Zusätzlich können hier beispielsweise Funktionalitäten der Bibliotheken seaborn und matplotlib zur weiteren graphischen Darstellung der Daten genutzt werden. Ferner wird der Datensatz in Eingangsvariablen X sowie Ausgangsvariablen y unterteilt. Dabei enthält X alle Spalten außer der Bewertung. y wird aus der Spalte Bewertung erzeugt, wobei die Werte wie zuvor beschrieben zu einer binären Klasse umgewandelt werden: 1 für „OK“ und 0 für „NOK“.

```
import pandas as pd

# Daten aus Excel-Datei laden
file_path = '/example/DATEN.xlsx'
data = pd.read_excel(file_path, engine='openpyxl')

# Erste Zeilen der Daten anzeigen
print("Erste Zeilen der Daten:")
print(data.head())

# Eingangs- und Ausgangsvariablen definieren
X = data.drop(['Bewertung'], axis=1)
y = data['Bewertung'].apply(lambda x: 1 if x == 'OK' else 0)
```

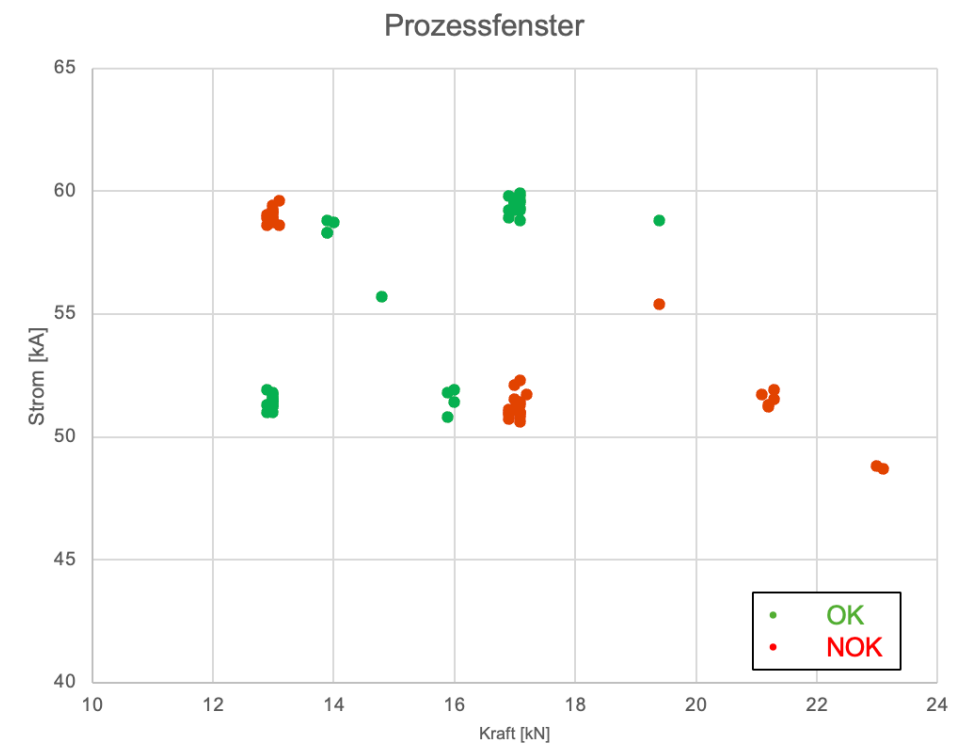


Abbildung 4: Bewertung des Bauteils in Abhängigkeit der Kraft und des Stroms

4

Modellauswahl und Training

Bestimmen Sie den Typ des KI-Modells, das am besten für Ihre Daten und Ihr Problem geeignet ist. Trainieren Sie anschließend das KI-Modell mit Ihren vorbereiteten Daten.

Basierend auf den zuvor definierten Zielen und der Art der verfügbaren Daten wird in einem nächsten Schritt ein KI-Modell ausgewählt und trainiert. Die Architektur des Modells sowie die Hyperparameter zum Training bestimmen maßgeblich die Leistung und Effektivität der endgültigen KI-Anwendung. Die einzelnen Unterschritte der Modellauswahl und des Trainings sind im Folgenden beschrieben:

Vorbereiten – Auswählen – Trainieren

4.1 Datenvorbereitung

- **Aufteilung in Trainings- und Testdaten:** Teilen Sie die erhobenen Daten in zwei Teile auf: einen Trainingsdatensatz inkl. Validierungsdatensatz sowie einen Testdatensatz. Die Aufteilung erfolgt üblicherweise in einem Verhältnis, das genug Daten für das Training lässt, während ebenfalls genügend Daten für die abschließende Bewertung übrigbleiben, z.B. Trainingsdatensatz 80 % und Testdatensatz 20 %.

4.2 Modellauswahl

- **Auswahl des Lernparadigmas:** Wählen Sie auf Grundlage des Problemverständnisses ein geeignetes Lernparadigma aus. Für klassische Probleme eignen sich oft bewährte Paradigmen wie Supervised Learning (Lernen durch Beobachtung und Deutung der Beobachtung sowie Korrektur der Deutung), Unsupervised Learning (Lernen durch Beobachtung und Deutung der Beobachtung ohne Korrektur der Deutung) oder Reinforcement Learning (Lernen durch Belohnung oder Bestrafung von Handlungen in einer definierten Umgebung).
- **Auswahl der Modellart:** Wählen Sie zudem auf Grundlage des Problemverständnisses eine geeignete Modellart aus. Die einfachste Modellart sind vorwärts gerichtete neuronale Netzwerke. Hier fließen die Informationen in nur eine Richtung von der Eingabeschicht bis hin zur Ausgabeschicht. Vorwärts gerichtete neuronale Netzwerke sind äußerst flexibel und können für eine Vielzahl von Problemen eingesetzt werden. Daneben gibt es für klassische Probleme eine Vielzahl bewährter Modelle wie Lineare Regression, Logistische Regression, Entscheidungsbäume, Support Vector Machines und viele mehr. Bei komplexeren Problemstellungen wie

Bild- oder Spracherkennung, können tiefgehende neuronale Netzwerke (Deep Learning) zum Einsatz kommen.

4.3 Training des Modells

- **Training:** Trainieren Sie das gewählte Modell mit den vorbereiteten Trainingsdaten. Hierbei lernt das Modell die Beziehung zwischen Merkmalen (Features) und Zielen (Labels) der Daten.
- **Hyperparameter-Tuning:** Die Leistung des Modells wird durch die Einstellung seiner Hyperparameter optimiert. Dieses Tuning kann systematisch erfolgen, z.B. durch Grid-Search, Random-Search oder fortgeschrittener Techniken wie Bayesian Optimization.

Die Schritte Modellauswahl und Training des Modells sind iterativ. Basierend auf den Erkenntnissen der Modellanalyse (siehe nächste Phase) können Sie im Prozess zurückgehen und z.B. verschiedene Modellarchitekturen oder Hyperparameter ausprobieren, um die Güte und Leistung des Modells zu verbessern. Durch das methodische iterative Durchlaufen dieser Schritte können Sie ein robustes und leistungsfähiges KI-Modell entwickeln, dass die spezifischen Anforderungen Ihres Problems erfüllt.

4.4 Beispiel

Bei dem vorliegenden Beispiel handelt es sich um eine klassische binäre Klassifizierungsaufgabe mit gelabelten Daten (Bewertung „OK“ oder „NOK“). Folglich kann als Lernparadigma Supervised Learning angewendet werden. Als Modellart wurde der Einfachheit halber ein vorwärts gerichtetes neuronales Netz mit zwei versteckten Schichten gewählt.

Auch das Erstellen sowie Trainieren des Modells kann werkzeuggestützt mithilfe des folgenden Codes durchgeführt werden. Dabei werden zunächst die nachfolgend genutzten Bibliotheken sowie Funktionen geladen. Anschließend wird der Datensatz in Trainings- (80 %) und Testdaten (20 %) mittels der Funktion `train_test_split` aufgeteilt. Danach wird mittels der Funktion `Sequential` ein sequenzielles Modell eines neuronalen Netzes erstellt. Dies ist ein Stapel von Schichten, wo jede Schicht genau eine Eingabe und eine Ausgabe hat. Mit der Funktion `Dense` werden anschließend dem Modell Schichten hinzugefügt. Eine Schicht des neuronalen Netzes benötigt dabei Spezifikationen wie die Anzahl der Neuronen und die Aktivierungsfunktion. Das hier definierte Modell hat drei Schichten: eine erste Eingabeschicht mit 64 Neuronen und ReLU-Aktivierung, eine zweite versteckte Schicht mit 32 Neuronen und ReLU-

Aktivierung sowie eine dritte Ausgabeschicht mit einem Neuron und Sigmoid-Aktivierung für die hier vorliegende binäre Klassifikationsaufgabe. Außerdem wird hier eine Verlustfunktion (`loss='binary_crossentropy'`), ein Optimierer (`optimizer='adam'`) und eine Metrik (`metrics='accuracy'`) angegeben, mit dem das Modell mittels der Funktion `model.compile` kompiliert wird. Nachfolgend kann das Modell des neuronalen Netzes mittels der Funktion `model.fit` mit den Trainingsdaten trainiert werden, hier über 50 Epochen (`epochs`) mit einer Batch-Größe von 10 (`batch_size`), wobei 20 % der Trainingsdaten für die Validierung reserviert werden (`validation_split`).

Die Anzahl Schichten, Neuronen pro Schicht sowie die Aktivierungsfunktion pro Schicht sind klassische Designparameter der Architektur eines neuronalen Netzwerks, welche zur Feinabstimmung des Modells variiert werden können.

Die Anzahl Epochen sowie die Batch-Größe sind ebenfalls klassische Designparameter beim Training eines neuronalen Netzwerks, welche zur Feinab-

```
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Train-Test-Split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2)

# Neuronales Netz erstellen
model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Modell kompilieren
model.compile(loss='binary_crossentropy', optimizer='adam',
              metrics=['accuracy'])

# Modell trainieren
history = model.fit(X_train, y_train, epochs=50, batch_size=8,
                   validation_split=0.2)
```

Haben Sie Fragen zur Auswahl des richtigen Modells zur Lösung Ihres Problems oder zur Wahl der Hyperparameter? Kontaktieren Sie uns. Wir beraten Sie gerne.



5

Modellanalyse und Feinabstimmung

Bewerten Sie, wie gut das KI-Modell auf Testdaten reagiert, die es während des Trainings nicht gesehen hat, und stimmen Sie das Modell ab, um seine Leistung zu verbessern.

Nachdem das Modell trainiert wurde, muss es evaluiert werden, um seine Leistung zu überprüfen. Hierbei werden verschiedene Metriken verwendet, um die Genauigkeit, Relevanz und andere Aspekte zu bewerten. Dies ist Gegenstand der Modellanalyse. Die anschließende Feinabstimmung des Modells zielt dann darauf ab, die Leistung des Modells zu optimieren und es effektiv auf spezifische Anwendungsfälle anzupassen. Die einzelnen Unterschritte dazu sind nachfolgend beschrieben:

Bewerten – Optimieren

5.1 Leistungsbewertung

- **Festlegung und Ermittlung von Evaluierungsmetriken:** Bestimmen Sie je nach Art des Problems (Klassifizierung, Regression usw.) passende Evaluierungsmetriken. Bei Klassifizierungsproblemen können dies beispielsweise sein:
 - Accuracy: Die Accuracy gibt den Anteil der korrekt gemachten Vorhersagen eines Modells an. Um die Accuracy zu erhalten, müssen Sie die Anzahl der korrekten Vorhersagen durch die Anzahl aller gemachten Vorhersagen teilen.
 - Precision: Die Precision gibt das Verhältnis der wahren positiven Ergebnisse an, die vom Modell korrekt vorhergesagt wurden. Die Precision versucht, die folgende Frage zu beantworten: Wie viele der positiven Ergebnisse wurden von dem Modell richtig vorhergesagt?
 - Recall: Der Recall, auch Sensitivität genannt, ist eine Metrik, die aussagt, wie gut das Modell in der Lage ist positive Ergebnisse zu identifizieren
 - F1-Score: Der F1-Score ist eine Bewertungsmetrik, die sowohl den Recall als auch die Precision eines Modells kombiniert. Genauer gesagt ist der F1-Score ein harmonisches Mittel aus beiden Metriken. Diese Metrik wurde eingeführt, um die Leistung eines Modells zu messen, ohne explizit Recall und Precision anzugeben, sondern um sie in einer einzigen Metrik zu kombinieren.

Bei Regressionsproblemen sind es häufig:

- mittlere absolute Fehler (MAE)
- mittlere quadratische Fehler (RMSE)
- R²-Wert.
- **Test:** Das Training des Modells wurde mit dem Trainingsdatensatz durchgeführt. Für die Bewertung des Modells nutzen Sie nun den separaten Testdatensatz, um eine unabhängige Evaluation der Leistung des Modells in einer realen Anwendungsumgebung zu bekommen.
- **Untersuchung von Fehlklassifizierungen:** Analysieren Sie bei Klassifizierungsproblemen die Fälle, in denen das Modell falsche Vorhersagen macht, um systematische Fehler oder Schwächen des Modells zu identifizieren.
- **Residuenanalyse:** Untersuchen Sie bei Regressionsproblemen die Residuen (Unterschiede zwischen den vorhergesagten und den tatsächlichen Werten), um Muster oder Trends aufzudecken, die auf Probleme mit dem Modell hinweisen könnten.

5.2 Modelloptimierung

- **Optimierung der Hyperparameter:** Experimentieren Sie mit verschiedenen Sets von Hyperparametern, um die beste Kombination für Ihr Modell zu finden. Zuvor erwähnte Methoden wie Grid-Search, Random-Search oder Bayesian Optimization können dabei helfen.
- **Optimierung des Feature Engineerings:** Überprüfen und optimieren Sie das Feature Engineering, indem Sie unnötige Merkmale entfernen, neue Merkmale erstellen oder vorhandene Merkmale transformieren, um die Modellleistung zu verbessern.

Die Modellentwicklung ist ein iterativer Prozess. Basierend auf den Ergebnissen der Modellanalyse und der Feinabstimmung können weitere Zyklen der Anpassung und Evaluierung notwendig sein, um die optimale Modellkonfiguration zu erreichen. Durch die gründliche Analyse und gezielte Feinabstimmung des Modells können Sie dessen Präzision und Effektivität maximieren und so sicherstellen, dass es robust und zuverlässig für den vorgesehenen Einsatzzweck funktioniert.

5.3 Beispiel

Der folgende Code bewertet das zuvor trainierte Modell. Dazu werden die Testdaten (`X_test`) verwendet und Vorhersagen des Modells für diesen berechnet (`y_pred`) sowie in eine binäre Klasse umgewandelt (größer als 0.5 wird zu 1, sonst zu 0). Anschließend wird die Konfusionsmatrix für das entwickelte Modell erstellt. Diese ist in Abbildung 5 dargestellt. Eine Konfusionsmatrix ist ein nützliches Hilfsmittel, um die Leistung eines KI-Modells zur Klassifikation zu bewerten, indem sie die tatsächlichen gegen die vorhergesagten Werte aufstellt. Darüber hinaus können an dieser Stelle auch gängige Metriken, wie die zuvor beschriebene Accuracy, berechnet werden und viele weitere Untersuchungen stattfinden.

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Vorhersagen
y_pred = (model.predict(X_test) > 0.5).astype("int32")

# Konfusionsmatrix erstellen
cm = confusion_matrix(y_test, y_pred)

# Konfusionsmatrix anzeigen
plt.figure(figsize=(7, 5))
sns.heatmap(cm, annot=True, cmap='Blues', xticklabels=['NOK', 'OK'],
            yticklabels=['NOK', 'OK'])
plt.xlabel('Vorhergesagte Klasse')
plt.ylabel('Tatsächliche Klasse')
plt.title('Konfusionsmatrix')
plt.show()
```

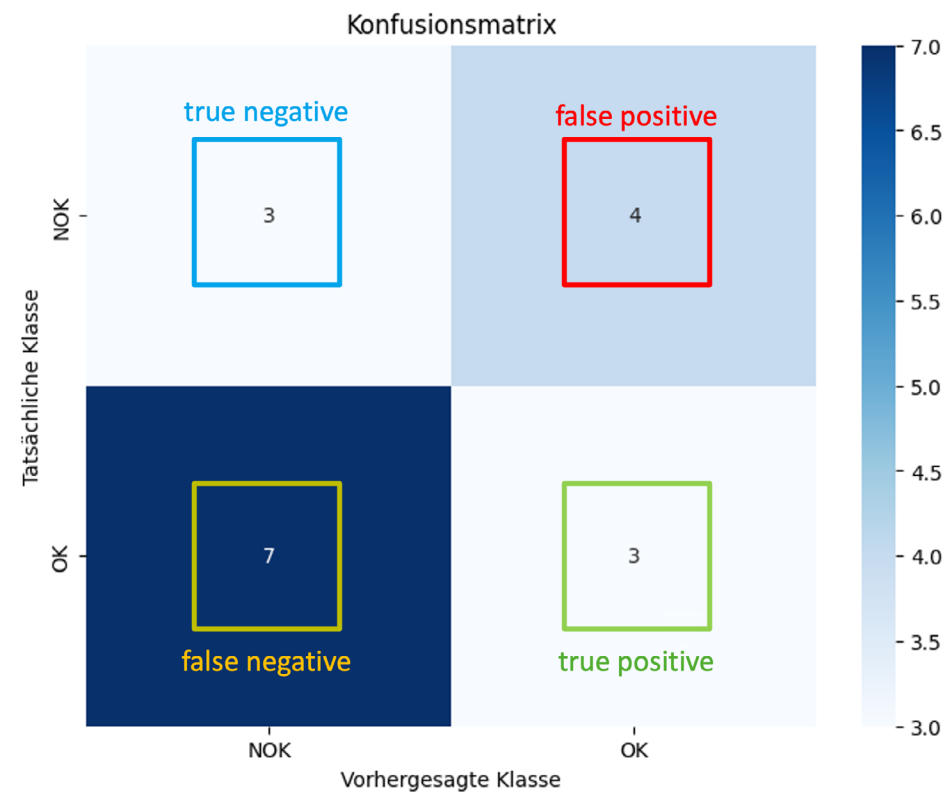


Abbildung 5: Konfusionsmatrix des entwickelten vorwärts gerichteten neuronalen Netzes

Anhand der Konfusionsmatrix ist ersichtlich, dass das initial entwickelte KI-Modell noch nicht in der Lage ist, mit einer ausreichenden Genauigkeit richtige Vorhersagen zu treffen. So werden gerade einmal 6 von 17 Testdaten richtig klassifiziert (3 true positive sowie 3 true negative). Daher wird die Modellarchitektur in einem nächsten Schritt optimiert. Hierfür wird ein Random Forest Classifier gewählt. Dieser bietet durch seine Kombination von Einfachheit, Vielseitigkeit und Effektivität eine leistungsfähige Lösung für binäre Klassifikationsaufgaben. Er gehört zur Gruppe der Ensemble-Lernmethoden, was bedeutet, dass er durch das Zusammenführen von Vorhersagen mehrerer Modelle (in diesem Fall Entscheidungsbäume) funktioniert.

Der folgende Code erstellt und trainiert wie zuvor beschrieben mittels der Funktion `model.fit` den Random Forest Classifier für denselben Datensatz und dieselbe binäre Klassifizierungsaufgabe. Der hier verwendete Random Forest Classifier besteht aus 10 einzelnen Entscheidungsbäumen (`n_estimators`).

Die Anzahl an Entscheidungsbäumen ist ein Designparameter des Random Forest Classifiers, welcher zur Feinabstimmung des Modells variiert werden kann.

```
from sklearn.ensemble import RandomForestClassifier

# Modelltraining
model = RandomForestClassifier(n_estimators=10)
model.fit(X_train, y_train)
```

Anschließend können erneut die Vorhersagen des Modells für den Testdatensatz wie zuvor beschrieben berechnet und die aktualisierte Konfusionsmatrix ermittelt werden (siehe Abbildung 6). Anhand der Konfusionsmatrix ist ersichtlich, dass durch die Wahl des Random Forest Classifiers als Architektur die Güte der Vorhersage des Modells stark verbessert werden konnte, da nun sämtliche Testdaten richtig klassifiziert werden (10 true positive sowie 7 true negative), sodass im nächsten Schritt das Modell realisiert werden kann.

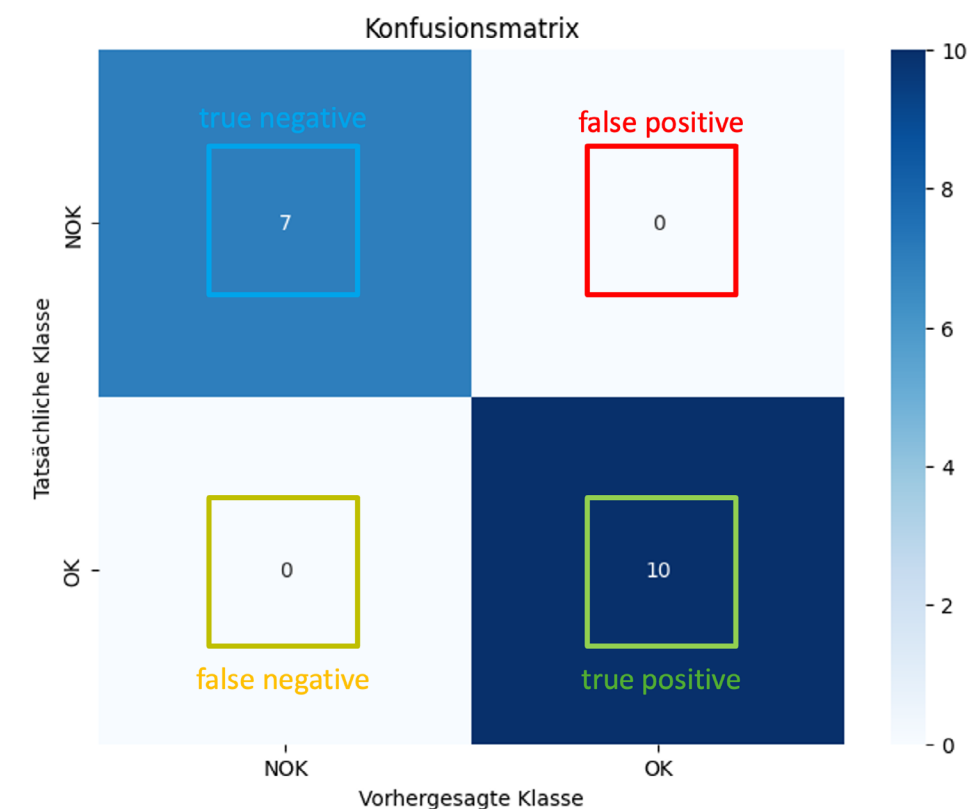


Abbildung 6: Konfusionsmatrix des entwickelten Random Forest Classifiers

6

Modelleinsatz und Wartung

Implementieren und integrieren Sie das KI-Modell in die endgültige Anwendung oder das System, innerhalb derer es eingesetzt werden soll. Überwachen Sie die Leistung des KI-Modells in der realen Welt und aktualisieren oder optimieren Sie es bei Bedarf.

Die letzte Phase beinhaltet die Integration und den Einsatz des KI-Modells in der realen Welt. Neben der eigentlichen Implementierung des Modells in der realen Umgebung sowie entsprechender Inbetriebnahme gehört dazu die kontinuierliche Überwachung und Wartung sowie Anpassung des Modells, um seine Relevanz und Genauigkeit über die Zeit sicherzustellen. Dabei werden die folgenden Schritte durchlaufen:

Implementieren – Überwachen – Aktualisieren – Dokumentieren

6.1 Implementierung des Modells

- **Vorbereitung:** Transformieren und optimieren Sie das Modell für die Produktionsumgebung. Dies kann das Kompilieren in ein effizienteres Format oder das Anpassen an die Besonderheiten der Zielplattform (z.B. Cloud, mobile Geräte, Edge-Geräte) beinhalten.
- **Implementierung:** Integrieren Sie das Modell in die vorhandene IT-Infrastruktur, was die Anbindung an Datenquellen, die Einrichtung von Inferenz-Diensten und die Verknüpfung mit Anwendungslogik einschließt, und nehmen Sie das Modell in Betrieb.
- **Automatisierung:** Stellen Sie Tools und Prozesse bereit, um Vorhersagen zu automatisieren und Benutzereingriffe so weit wie möglich zu reduzieren.

6.2 Überwachung

- **Leistungsüberwachung:** Überwachen Sie kontinuierlich die Leistung des Modells durch Analyse der Modellmetriken und Vergleich mit den Benchmark-Werten aus der Testphase.
- **Datenqualitätsüberwachung:** Überprüfen Sie regelmäßig die Qualität und das Format der Eingabedaten, da Veränderungen hier die Modelleleistung beeinflussen können.
- **Anomalieerkennung:** Richten Sie Überwachungssysteme ein, die automatisch Anomalien oder Abweichungen in der Modellperformance erkennen, um proaktiv Probleme anzugehen.

6.3 Aktualisierung und Anpassung

- **Regelmäßige Re-Evaluierung:** Planen Sie regelmäßige Überprüfungen des Modells, um sicherzustellen, dass es weiterhin die erwarteten Ergebnisse liefert, insbesondere, wenn sich die zugrundeliegenden Daten oder Anforderungen ändern.
- **Modellaktualisierungen:** Führen Sie notwendige Anpassungen oder komplette Neustrainings des Modells durch, wenn sich die Daten signifikant ändern oder die Modellperformance unter ein akzeptables Niveau fällt.
- **Inkrementelles Lernen:** Implementieren Sie Mechanismen für das Modell, um kontinuierlich aus neuen Daten zu lernen und sich entsprechend anzupassen, wenn dies von der Problemstellung her sinnvoll und realisierbar ist.

6.4 Dokumentation und Compliance

- **Dokumentation:** Halten Sie die wesentlichen Entscheidungen, Erkenntnisse sowie Ergebnisse jede der zuvor durchlaufenden Phasen des Lebenszyklus des Modells fest, einschließlich Entwicklung, Implementierung und Optimierung.
- **Datenschutz und Ethik:** Stellen Sie sicher, dass die Modellanwendung und Datenverarbeitungspraktiken den geltenden Datenschutzgesetzen und ethischen Richtlinien entsprechen.

Durch die sorgfältige Durchführung dieser Schritte können Sie die Langlebigkeit und Relevanz Ihres KI-Modells in der realen Welt maximieren sowie sicherstellen, dass es sich an verändernde Bedingungen anpasst.

6.5 Beispiel

Der entwickelte Random Forest Classifier zur Lösung der binären Klassifizierungsaufgabe besteht aus mehreren Entscheidungsbäumen, welche im letzten Schritt auf der Zielhardware realisiert und kombiniert werden. In Abbildung 7 ist ein exemplarischer Entscheidungsbaum dargestellt. Ein einzelner Entscheidungsbaum besteht dabei aus Knoten, die Entscheidungen repräsentieren, und Kanten/Ästen, die mögliche Pfade repräsentieren, die man abhängig von diesen Entscheidungen nehmen kann, sowie folgenden Elementen:

- **Wurzelknoten:** Der Startpunkt des Baumes. Anhand einer bestimmten metrischen Entscheidung (hier der sogenannten Gini-Unreinheit) wählt der Algorithmus das beste Feature und den entsprechenden Schwellenwert aus, um den Datensatz in Untergruppen aufzuteilen.
- **Interne Knoten:** Entscheidungspunkte unterhalb der Wurzel. Jeder interne Knoten stellt eine „Frage“ basierend auf einem Feature und seinem Schwellenwert (zum Beispiel, ob das Feature A größer als 5 ist). Abhängig von der Antwort auf diese Frage folgt man einem der Äste zum nächsten Knoten (ja links oder nein rechts).
- **Blätter (Endknoten):** Die Terminalpunkte des Baumes, die keine weiteren Verzweigungen haben. Jedes Blatt repräsentiert eine Vorhersage. Hier ist dies die Klasse „OK“ oder „NOK“.
- **Äste:** Die Verbindungen zwischen den Knoten, repräsentieren die Antwort auf die Frage im vorherigen Knoten. Sie führen zu weiteren Entscheidungspunkten oder zu Endpunkten (Blättern).

Ein Entscheidungsbaum wird dabei wie folgt gelesen:

- **Beginnen Sie an der Wurzel:** Betrachten Sie das erste Entscheidungskriterium (Feature und Schwellenwert).
- **Folgen Sie den Ästen:** Entscheiden Sie basierend auf den Feature-Werten Ihrer Eingabedaten, welchem Ast Sie folgen. Jeder Ast steht für eine mögliche Antwort (z.B. wahr/falsch oder ja/nein).
- **Wiederholen Sie den Prozess:** An jedem internen Knoten, treffen Sie eine weitere Entscheidung, bis Sie ein Blatt erreichen.

Interpretieren Sie das Blatt: Hier zeigt das Blatt die Klasse an, der die Daten zugeordnet sind.

Im Random Forest Classifier wird dieser Prozess für jeden Entscheidungsbaum durchgeführt (hier 10), und die Ergebnisse werden kombiniert, um die finale Vorhersage zu erzielen. Kommen für einen Datensatz beispielsweise 6 der Entscheidungsbäume zu dem Ergebnis „OK“ sowie die verbleibenden 4 zu dem Ergebnis „NOK“, so ist finale Vorhersage des Random Forest Classifiers „OK“ mit einer Wahrscheinlichkeit von 60 %.

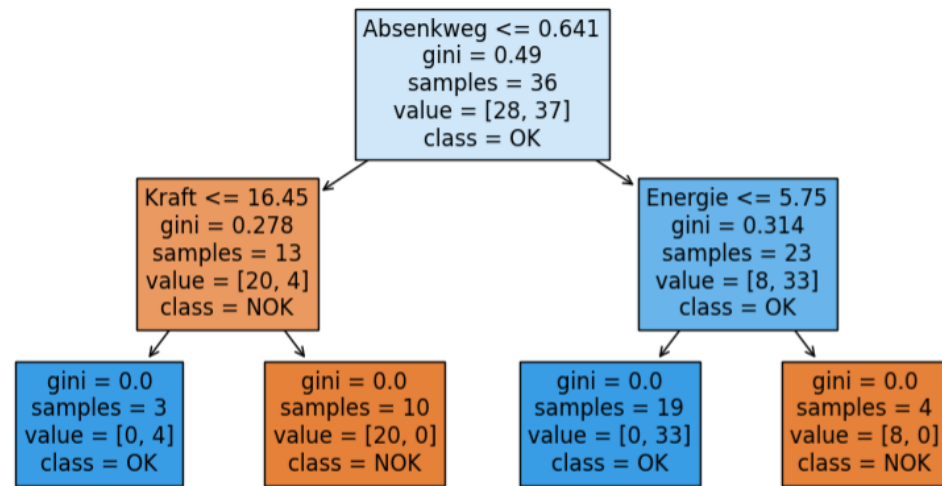


Abbildung 7: Exemplarischer Entscheidungsbaum des entwickelten Random Forest Classifiers

Haben Sie Fragen oder benötigen Sie Unterstützung?

Wir hoffen, dass unser Leitfaden Ihnen wertvolle Einblicke und Anweisungen zur Entwicklung Ihres eigenen KI-Modells gegeben hat. Sollten Sie diesbezüglich Fragen haben oder Unterstützung benötigen, zögern Sie nicht, uns zu kontaktieren.

Unser engagiertes Team steht Ihnen mit Expertenwissen und individueller Beratung zur Seite. Gemeinsam finden wir passende Lösungen und begleiten Sie Schritt für Schritt auf Ihrem Weg zum erfolgreichen KI-Projekt.



Bildnachweise

Titelseite: Unsplash - ZHENYU LUO

Seite 5: Pixabay - aditya859

Seite 12: Hans Berg GmbH & Co. KG

Seite 16: Hans Berg GmbH & Co. KG

Seite 21: Hans Berg GmbH & Co. KG

Seite 30: TH Köln

Seite 31: TH Köln

Seite 36: TH Köln

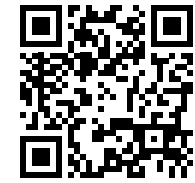
Rechtliche Hinweise

Diese Publikation ist urheberrechtlich geschützt. Die Vervielfältigung oder Bearbeitung der Inhalte, auch auszugsweise, ist ohne ausdrückliche schriftliche Genehmigung des Autors nicht gestattet, es sei denn, es handelt sich um eine Nutzung im Rahmen der gesetzlichen Bestimmungen. Für die Verwendung von in dieser Publikation enthaltenem Material, das von Dritten stammt und möglicherweise gesondert lizenziert ist, ist eine separate Genehmigung der jeweiligen Rechteinhaber erforderlich. Für die Verwendung von urheberrechtlich geschütztem Material, das in dieser Publikation enthalten ist und nicht durch diese Lizenz abgedeckt ist, müssen die Rechteinhaber direkt kontaktiert werden, um die Erlaubnis zur Weiterverwendung einzuholen. Der Autor übernimmt keine Haftung für Schäden, die aus der Nutzung der in dieser Publikation enthaltenen Informationen resultieren.

Kontakt

Transformationsnetzwerk TrendAuto2030plus
info@trendauto2030.de
Gustav-Heinemann-Ufer 54
50968 Köln

Layout und Satz: Rafael Rabe



trendauto2030plus.de

Impressum

TH Köln
Claudiusstr. 1
50678 Köln
Postanschrift
Gustav-Heinemann-Ufer 54
50968 Köln

T: + 49 221-8275-0
F: + 49 221-8275-3131
E: poststelle@th-koeln.de

Die TH Köln (Technische Hochschule Köln) ist eine Körperschaft des Öffentlichen Rechts. Sie wird durch die Präsidentin Prof. Dr. Sylvia Heuchemer gesetzlich vertreten.

Zuständige Aufsichtsbehörde:
Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen
Völklinger Str. 49
40221 Düsseldorf
T: +49 211-896-04
F: +49 211-896-4555
<http://www.wissenschaft.nrw.de/>
Umsatzsteuer-Identifikationsnummer gem. § 27a Umsatzsteuergesetz:
DE 122653679



TrendAuto
2030plus