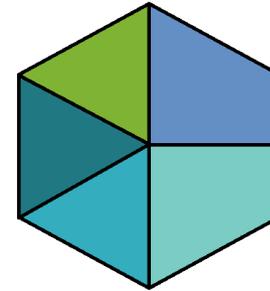




**Hochschule
Kaiserslautern**
University of
Applied Sciences

Angewandte
Ingenieurwissenschaften
Kaiserslautern



elab2go
Mobile
Engineering
Lab

Machine Learning mit Python und Jupyter Notebook

Prof. Dr. Eva Maria Kiss
M. Sc. Anke Welz

Maschinelles Lernen (Machine Learning, ML): Die Entwicklung lernfähiger Systeme

Formell:

Eine Maschine **lernt** aus der **Erfahrung E** hinsichtlich einer Klasse von **Aufgaben T** und dem **Performance-Maß P**, falls die Performance P hinsichtlich T mit E sich verbessert.

[Tom Mitchell, Machine Learning, 1997]

Informell:

Maschinelles Lernen ist ein Teilgebiet der Künstlichen Intelligenz, das es Systemen ermöglicht, auf Basis von Trainingsdaten automatisch zu lernen und hinzuzulernen.

Maschinelles Lernen befasst sich mit der Entwicklung lernfähiger Systeme und Algorithmen.

Machine Learning: Ziele und Anwendungen

Machine Learning-Algorithmen haben im weitesten Sinne das Ziel, aus Input-Daten **sinnvolle Zusammenhänge zu erkennen** und daraus Regeln abzuleiten. Z.B. Vorhersagen treffen, Trends erkennen, Daten nach bestimmten Kriterien gruppieren.

Anwendungsgebiete sind

- Vorausschauende Wartung: Fällt Maschine aus oder nicht?
- Mobile Assistenten (Siri, Alexa, Cortana, Bixby, ...)
- Betrugsaufdeckung: Ist E-Mail Spam oder nicht?
- Suchmaschinen, Produktempfehlungen, Clusteranalysen im Rahmen des gezielten Marketings

Machine Learning: Historische Entwicklung

Maschinelles Lernen hat seit seinen Anfängen um 1950 mehrere Auf-und-ab-Phasen erlebt. Klassisches datenzentrisches Maschinelles Lernen ist seit 1990 eine Disziplin der Datenanalysten und Statistiker. Seit ca. 2010 hat Deep Learning die Forschung beflügelt und zu einem neuen Hype geführt.

2010 Kaggle geht online, eine Plattform für Datenanalysten

2012 Google Brain kann mit Hilfe eines neuronalen Netzwerkes Katzen in YouTube-Videos identifizieren

2014 Facebooks DeepFace-Software kann Gesichter identifizieren

... Weiterentwicklung virtueller Assistenten ...

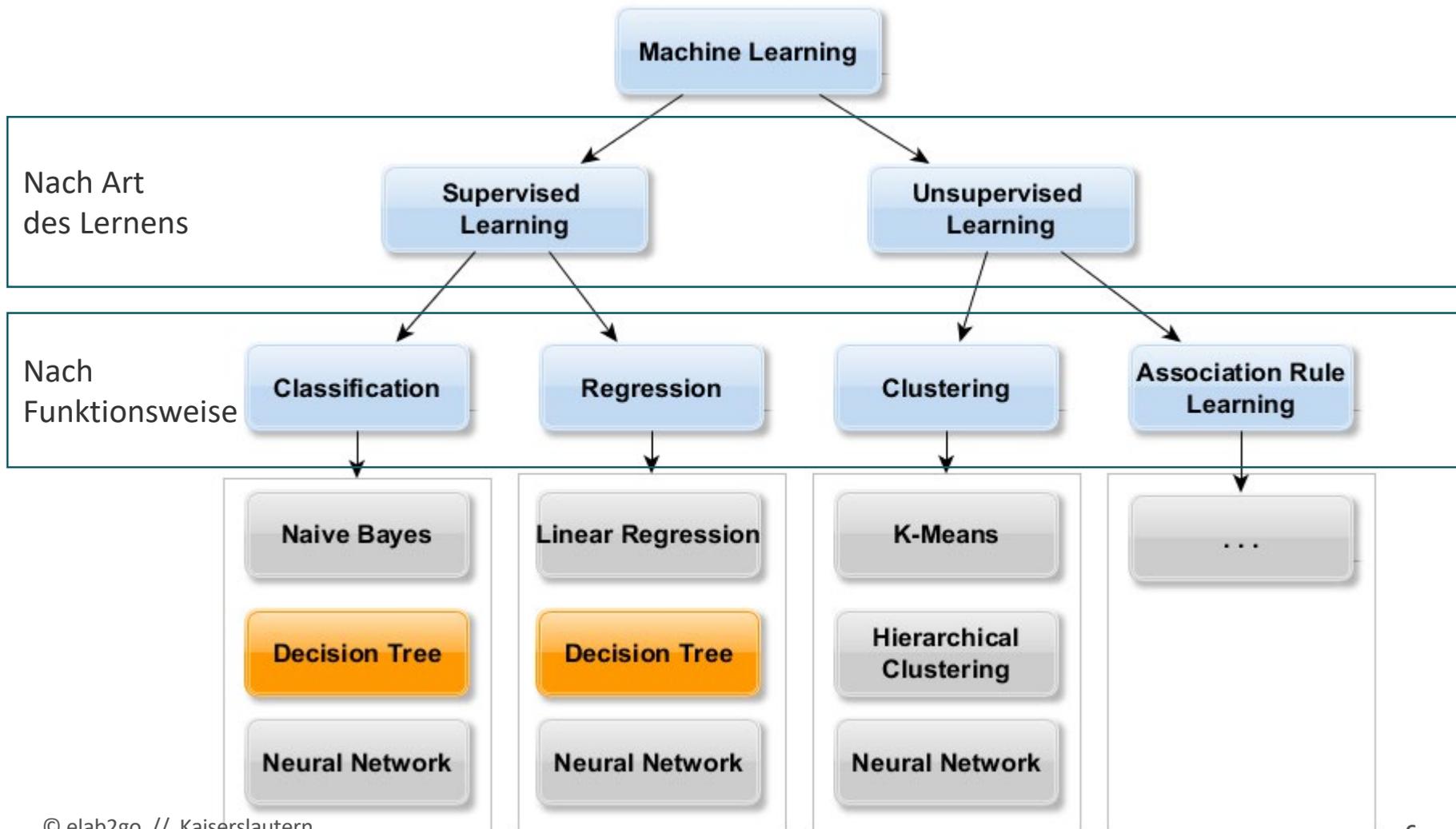
Machine Learning: Aktueller Hype bedingt durch Big Data

Durch die verstärkte Speicherung anonymisierter Nutzerdaten im Internet und die Digitalisierung in der Industrie (Industrie 4.0) sind riesige Datenspeicher entstanden, die als Basis für lernende Systeme verwendet werden können.

Der Bedarf, aus den schnelllebigem und semi-strukturierten Daten sinnvolle Informationen zu extrahieren (Spracherkennung, Bilderkennung) hat zur Weiterentwicklung der Sprachen und Bibliotheken für Maschinelles Lernen geführt.

Heute wird Angewandtes Maschinelles Lernen verstärkt auch von Informatikern und Ingenieuren betrieben.

Machine Learning: Methoden und Algorithmen



Machine Learning: Überwachtes Lernen

Beim überwachten Lernen liegt für jeden Datensatz der Input-Daten eine Bewertung vor, d.h. die Daten sind schon in Kategorien unterteilt. Der Ablauf ist wie folgt:

- Die Input-Daten werden in **Trainingsdaten** und **Testdaten** unterteilt.
- Auf Basis der Trainingsdaten wird mit Hilfe eines passenden Algorithmus ein **Vorhersagemodell** erstellt, dessen Güte mit Hilfe von **Performance-Kennzahlen** ermittelt wird.
- Der Trainingsprozess wird solange wiederholt, bis das Modell eine gewünschte Performance erreicht. Danach kann es für die Vorhersage auf neuen Datensätzen verwendet werden.

Tools und Sprachen für Maschinelles Lernen

Für die Entwicklung einer ML-Anwendung werden drei Sprachen mit den dazu passenden Entwicklungsumgebungen verwendet:

- R und RStudio → **elab2go Demo2, Demo3, Demo4**
 - Die klassische Lösung (open source, kostenlos)
 - Wird bevorzugt von Datenanalysten eingesetzt
- MATLAB → **elab2go Demo5**
 - Die kommerzielle Lösung
 - Wird bevorzugt von Ingenieuren eingesetzt
- Python → **elab2go Demo-PY3, Demo-PY4, Demo-PY5**
 - Die innovative Lösung (open source, kostenlos)
 - Wird im Internet-Umfeld eingesetzt, von Einsteigern und Programmierern bevorzugt

Demo-PY4: Predictive Maintenance mit scikit-learn

Gegeben: ein Automobildatensatz mit Messungen. Für einen Teil der Datensätze ist die Ausprägung der Zielvariablen „Ausfall“ bekannt.

Messungsnummer	Ausfall	Kuehlmitteltemperatur	EinspritzmengeKurzeit	Luftdruck	KatalysatorTemp	Kategorie
M_001	nein	2,6	,9	2,8		normal
M_002	ja	9,2	22,7	2,8		hoch
M_003	nein	3,2	28,5	,1		normal
M_004	ja	2,8	2,6	,2		normal
M_005	ja	4,9	14,4	3,0		hoch
M_006	nein	,8	5,5	2,9		hoch
M_007	nein	6,2	26,4	,1		normal
⋮						
Messungsnummer	Ausfall	Kuehlmitteltemperatur	EinspritzmengeKurzeit	EinspritzmengeLangzeit	Kraftstoffdruck	
M_137	?	2,6	,9	4,2	13,8	
M_138	?	9,2	22,7	15,3	8,6	
M_139	?	3,2	28,5	8,7	4,9	
M_140	?	2,8	2,6	5,9	7,4	

Gesucht: Prognose für neue Messungen, bei welcher Kombination von Sensorwerten ein Ausfall zu erwarten ist.

Ergebnis: Entscheidungsbaum-Vorhersagemodell

<https://www.elab2go.de/demo-py4/>

(1) Trainingsphase: Auf Basis der Trainingsdaten wird ein Entscheidungsbaum erstellt, dessen Güte mit Hilfe der Kennzahlen Accuracy (Vertrauenswahrscheinlichkeit), Precision (Genauigkeit) und Recall (Trefferquote) gemessen wird.

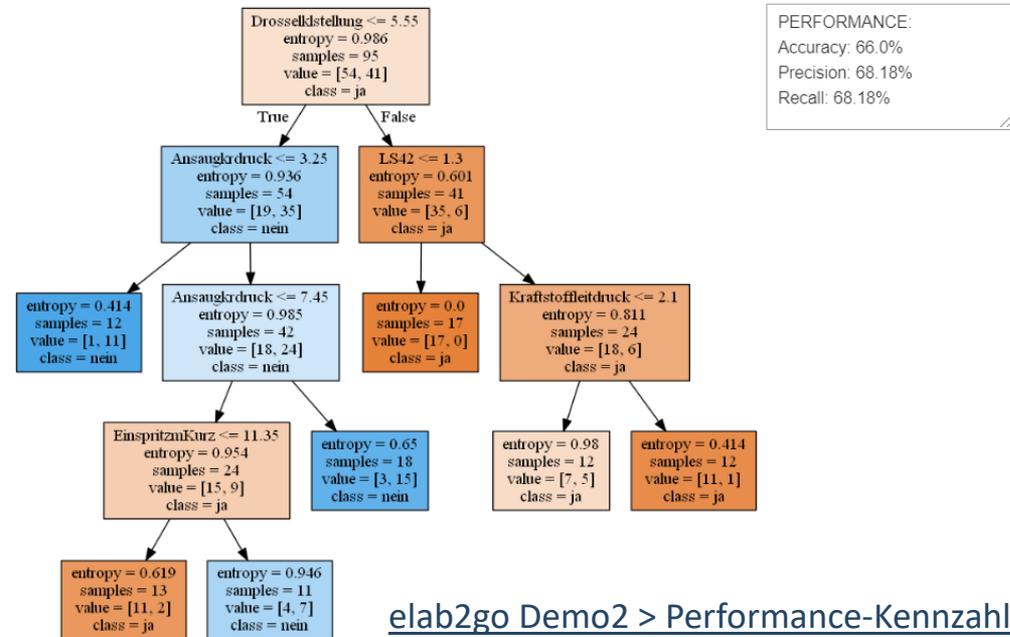
(2) Vorhersage:

Der optimierte Baum kann für die Prognose von Ausfällen eingesetzt werden.

Wenn am Ende eines Pfades "Ja" steht, dann führt der Pfad zu einem Ausfall des Motors andernfalls, d.h. bei "Nein", kommt es zu keinem Ausfall.

Interaktive Visualisierung des Entscheidungsbaums

Kriterium	entropy	Minsplit	<input type="range" value="0.15"/>	0.15
Aufteilung	best	Maxfeat	<input type="range" value="10"/>	10



[elab2go Demo2 > Performance-Kennzahlen](#)

Umsetzung des Demonstrators mit: Python, scikit-learn, Jupyter Notebook, Widgets

Für die Datenvorbereitung, die Erstellung des Entscheidungsbaum-Prognosemodells und die Vorhersage verwenden wir die Programmiersprache **Python**, die Bibliothek **scikit-learn** für Maschinelles Lernen, sowie **Jupyter Notebook** als Entwicklungsumgebung.

Um das beste Vorhersagemodell zu finden, erstellen wir in Jupyter Notebook ein **interaktives Widget**, das die Daten automatisch mit einer Benutzeroberfläche verbindet.

In dem interaktiven Widget können die wichtigsten **Konfigurationsparameter** des ML-Algorithmus eingestellt werden. Bei jeder Änderung eines Konfigurationsparameters wird ein neues Vorhersagemodell erstellt und visualisiert, und es werden die Performance-Kennzahlen automatisch berechnet und angezeigt.

Python als Programmiersprache für Datenanalyse

<https://www.elab2go.de/demo-py1/>

Die Programmiersprache **Python** wurde um 1990 in Amsterdam entwickelt. Die aktuelle Version Python 3.0 wird vor allem im Umfeld des Machine Learning verwendet, aber auch bei der Programmierung eingebetteter Systeme (Raspberry Pi).

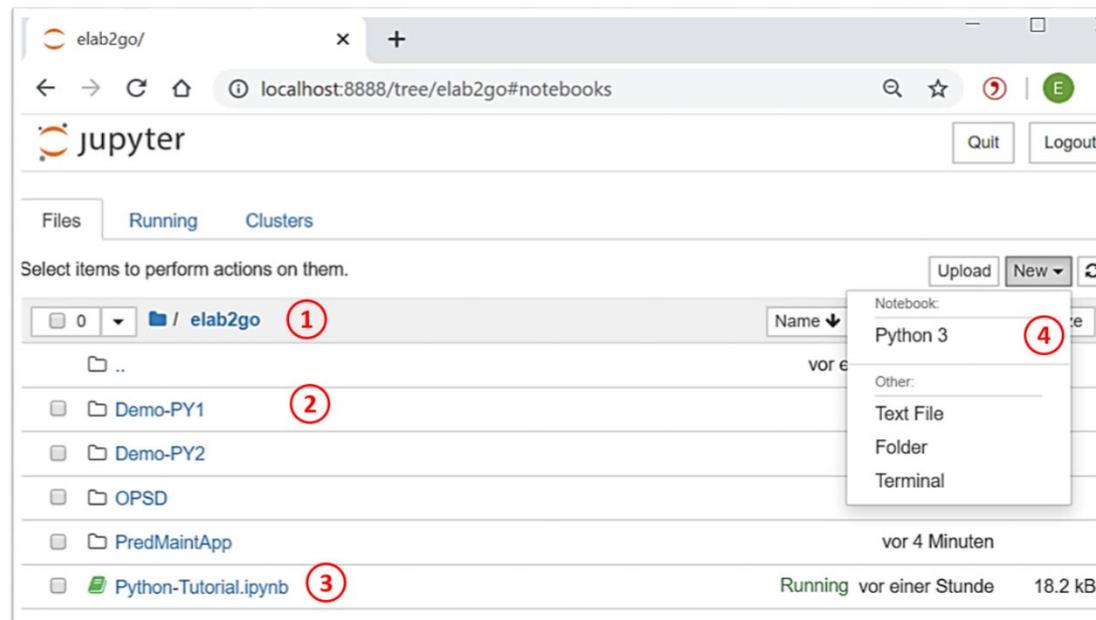
Python kommt in Kombination mit der Plattform **Anaconda** (für Anwendungs- und Paketverwaltung) und **Spyder** oder **Jupyter Notebook** als Entwicklungsumgebungen zum Einsatz.

Vorteil: Leichter Einstieg in die Datenanalyse durch Verwendung der kostenlosen Python-Programmpakete für Datenverwaltung, -Modellierung und -Analyse.

Jupyter Notebook als Lernumgebung

<https://www.elab2go.de/demo-py1/jupyter-notebooks.php>

Jupyter Notebook ist eine webbasierte Umgebung, die das Erstellen, Dokumentieren und Teilen von Demonstratoren unterstützt, und zwar insbesondere im Umfeld der Datenanalyse.

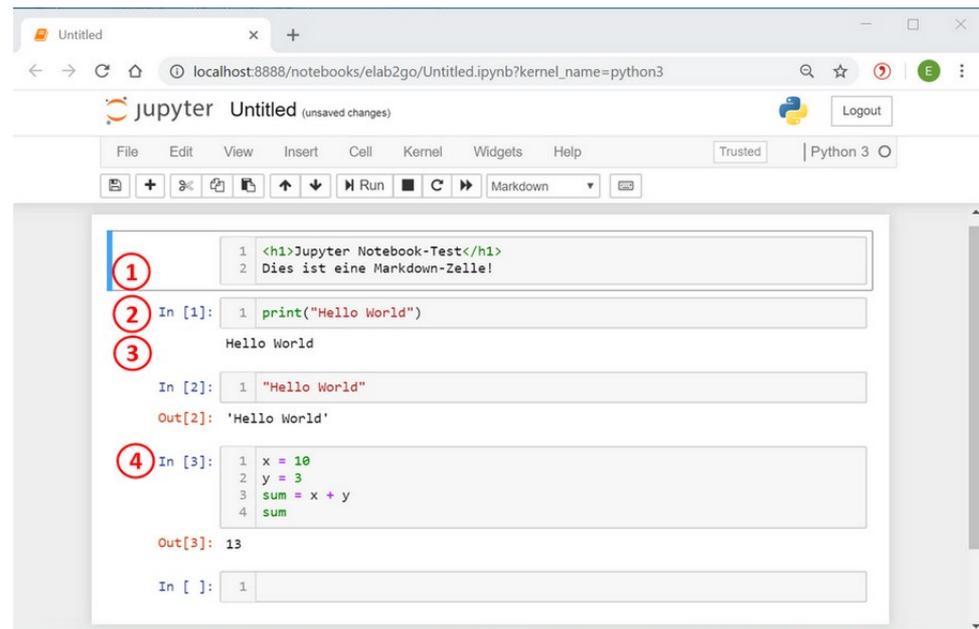


Jupyter Notebook: Funktionalität: Dokumentierten Code schreiben, ausführen und teilen

In einem Jupyter Notebook kann man Code schreiben und ausführen, Daten visualisieren, und diesen Code auch mit anderen teilen. Code und Markup (Beschreibung des Codes) werden in unabhängige Zellen geschrieben und können separat ausgeführt werden.

Notebooks können in andere Formate exportiert werden, z.B. als HTML, LaTeX, PDF, Slideshow.

Dies vereinfacht die Veröffentlichung von Notebooks im Internet (z.B. Kaggle).



The screenshot shows a Jupyter Notebook interface in a browser window. The notebook is titled "Untitled" and is running on a Python 3 kernel. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The notebook content is displayed in a scrollable area with four cells, each numbered in a red circle:

- Cell 1:** A Markdown cell containing the text `<h1>Jupyter Notebook-Test</h1>` and `Dies ist eine Markdown-Zelle!`.
- Cell 2:** A code cell with `print("Hello World")`. The output is `Hello World`.
- Cell 3:** A code cell with `"Hello World"`. The output is `'Hello World'`.
- Cell 4:** A code cell with `x = 10`, `y = 3`, `sum = x + y`, and `sum`. The output is `13`.

Zusammenfassung und Ausblick

- Maschinelles Lernen befasst sich mit der Entwicklung lernfähiger Systeme und Algorithmen.
- Je nach Art des Lernens unterscheidet man überwachtetes Lernen und unüberwachtes Lernen und verschiedene Arten von Modellen (Entscheidungsbäume, Neuronale Netze, Clusteralgorithmen, ...).
- Für die Umsetzung der Datenanalyse bzw. der Entwicklung eines lernenden Systems können verschiedene Programmiersprachen und Tools verwendet werden: Python, R, MATLAB, RapidMiner.
- Die Verwendung von Python mit Jupyter Notebook ist besonders für Einsteiger interessant, sowie im Industrie-Bereich, wo die Datenanalyse z.B. in Echtzeit auf einem eingebetteten System laufen soll.
- Auf der elab2go-Plattform bieten wir aktuell fünf Demos aus dem Themenkreis **ML & Python** an, weitere sind in Arbeit.

**VIELEN DANK FÜR IHRE
AUFMERKSAMKEIT!**