



Das QUA³CK-Prozessmodell für Data-Science-Projekte

Eine strukturierte Methodik von der Fragestellung bis zum Deployment.

Kurs: AMALEA 2025 - Woche 1

Agenda

1

Einleitung und Lernziele

Definition und Relevanz des QUA³CK-Modells.

2

Das QUA³CK-Prozessmodell im Detail

- **Q:** Question (Fragestellung formulieren)
- **U:** Understanding the Data (Datenverständnis aufbauen)
- **A³:** Die A-Schleife (Algorithmen und Analyse)
- **C:** Conclude & Compare (Schlussfolgerung und Validierung)
- **K:** Knowledge Transfer (Wissenstransfer und Bereitstellung)

3

Praxisbeispiel

Ein Iris-Klassifikator mit MLOps-Integration.

4

Anwendung im Portfolio

Übertragung auf individuelle Fallstudien.

5

Zusammenfassung und Ausblick

Wesentliche Erkenntnisse und nächste Schritte.



Lernziele der heutigen Sitzung



QUA³CK-Prozess anwenden

Das QUA³CK-Modell als strukturierten Rahmen für Machine-Learning-Projekte verstehen und dessen Schritte systematisch durchlaufen.



MLOps-Prinzipien integrieren

Moderne MLOps-Praktiken wie Experiment-Tracking in Arbeitsabläufe integrieren, um Reproduzierbarkeit und Effizienz zu steigern.



Praxisbeispiel entwickeln

Anhand des Iris-Klassifikators praktische Erfahrung mit der Anwendung des QUA³CK-Prozesses sammeln und alle Phasen durchlaufen.



Experiment-Management verstehen

Die Bedeutung strukturierter Experimente für die wissenschaftliche Validität und Qualitätssicherung von Data-Science-Projekten erkennen.



Von Analyse zu Produktion planen

Den Übergang von der experimentellen Phase zur Produktionsreife (z. B. mit Streamlit) strategisch planen und umsetzen.

Warum strukturierte Prozesse entscheidend sind

Die Zahlen sprechen für sich: Ohne strukturierte Methodik scheitern die meisten ML-Projekte. QUA³CK bietet einen bewährten Rahmen, um diese Herausforderungen zu meistern.

85-87 %

Data-Science-Projekte scheitern

13 %

Projekte erreichen
Produktionsreife

40 %

Kostenreduktion durch MLOps-
Praktiken

39 Mrd.

Prognostizierter MLOps-Markt
2034

Quellen: Gartner (2017-2019), Grand View Research (2024), VentureBeat (2019)

Eine strukturierte Methodik für Machine Learning

Das **QUA³CK**-Modell, entwickelt von Stock, S. et al. (2021) am Institut für Technik der Informationsverarbeitung (ITIV) des Karlsruher Instituts für Technologie (KIT), bietet einen praxisorientierten Rahmen zur strukturierten Durchführung von Machine-Learning-Projekten. Es wurde speziell konzipiert, um die Lücke zwischen akademischer Forschung und industrieller Praxis zu schließen und zeichnet sich durch seinen didaktischen Fokus aus, der es für Ingenieurstudierende leicht merkbar und systematisch anwendbar macht. Die zugehörige Publikation ist: „QUA³CK – A Machine Learning Development Process“ (Stock et al., 2021, KIT). Es unterteilt den komplexen Prozess in fünf systematisch zu durchlaufende Phasen.

Q – Question (Fragestellung)

Präzise Definition des Geschäftsproblems und der Projektziele. Diese Phase bildet das Fundament für alle weiteren Schritte und gewährleistet die Beantwortung relevanter Fragen.

U – Understanding the data (Datenverständnis)

Explorative Datenanalyse (EDA) zur Gewinnung von Einblicken in Datenstruktur, -qualität und -verteilung. Dabei werden erste Hypothesen generiert und die Basis für die Modellentwicklung gelegt.

A³ – Algorithm selection, Adapting features, Adjusting hyperparameters

Die iterative Phase der Modellentwicklung und -optimierung. Hierbei werden Algorithmen evaluiert, Features adaptiert und Hyperparameter optimiert, um die bestmögliche Modellleistung zu erreichen.

C – Conclude and compare (Schlussfolgerung und Vergleich)

Bewertung und Auswahl des optimalen Modells anhand definierter Metriken. Die Ergebnisse verschiedener Modelle werden systematisch verglichen, um die beste Lösung zu identifizieren.

K – Knowledge transfer (Wissenstransfer)

Dokumentation, Kommunikation der Ergebnisse und Überführung in die Anwendung. Diese Phase sichert die effektive Nutzung gewonnener Erkenntnisse und deren Integration in produktive Systeme.

Von der Analyse zur Produktion: Ein moderner Ansatz

AMALEA 2025 integriert moderne **MLOps-Praktiken** in das klassische QUA³CK-Modell, um die Kluft zwischen experimenteller Entwicklung und produktivem Einsatz zu überbrücken. Dieser umfassende Ansatz verzahnt die systematische QUA³CK-Methodik nahtlos mit den operativen Anforderungen realer Projekte. Der MLOps-Markt wächst von 1,7 Mrd. USD (2024) auf prognostizierte 39 Mrd. USD (2034) – ein jährliches Wachstum von 40,5 % (Grand View Research, 2024). Schon heute setzen 87 % der Großunternehmen KI-Lösungen ein (2025), was die Notwendigkeit robuster MLOps-Praktiken unterstreicht.

QUA ³ CK-Phase	Traditioneller Ansatz	AMALEA 2025 (MLOps-Ansatz)	Werkzeuge
Q + U	Statische Jupyter-Notebooks	Interaktive Analyse-Apps	Streamlit, Docker
A ³	Lokale, manuelle Experimente	MLFlow Experiment Tracking	MLFlow, GitHub
C	Manuelle Reports (z. B. Excel)	Automatisierter Modellvergleich	MLFlow UI, Dashboards
K	Lokale Modellbereitstellung	Cloud-Deployment & Modellportfolio	Streamlit Cloud, GitHub

Die Integration von MLOps-Praktiken in den QUA³CK-Prozess schafft eine nahtlose Verbindung zwischen experimenteller Forschung und produktiven Anwendungen. Unternehmen mit ausgereiften MLOps-Praktiken berichten von 40 % Kostenreduktion im ML-Lifecycle und 97 % Verbesserung der Modell-Performance. Dies führt zu verbesserter Reproduzierbarkeit, höherer Effizienz und einer robusteren Qualitätssicherung in Machine-Learning-Projekten.

Der AMALEA 2025-Ansatz stellt sicher, dass Studierende nicht nur theoretisches Wissen erwerben, sondern auch praxisrelevante Fähigkeiten entwickeln, die in der modernen Data-Science-Landschaft unerlässlich sind. Die Kombination aus strukturierter Methodik und fortschrittlichen Tools bereitet optimal auf die Herausforderungen realer Projekte vor.

Der entscheidende erste Schritt: Das Problem verstehen

Die Bedeutung der Phase Q

Jedes erfolgreiche Data-Science-Projekt beginnt mit einer klaren und präzisen Fragestellung. Diese Phase bildet das Fundament und ist entscheidend für den Gesamterfolg des Projekts. Eine unzureichende Problemdefinition führt selbst bei exzellenter technischer Umsetzung zu Ergebnissen, die den tatsächlichen Anforderungen nicht gerecht werden.

Statistiken belegen die Herausforderung: **85–87 % aller Data-Science-Projekte scheitern** (Gartner, 2017–2019), und nur **13 % der ML-Projekte erreichen die Produktionsreife**. Die häufigsten Gründe für dieses Scheitern sind eine **unklare Problemdefinition, fehlende Erfolgsmetriken und mangelnde Stakeholder-Abstimmung**. Dies unterstreicht die kritische Notwendigkeit, bereits in der frühen Phase eines Projekts die richtigen Weichen zu stellen.

Während in der akademischen Ausbildung oft vordefinierte Probleme gestellt werden, ist die Fähigkeit, ein Business-Problem in eine präzise Data-Science-Fragestellung zu übersetzen, im beruflichen Kontext von unschätzbarem Wert. Eine präzise Fragestellung in Phase Q reduziert das Risiko des Projektscheiterns erheblich und legt den Grundstein für nachhaltigen Erfolg.

Klare Problemstellung

Welches konkrete Problem soll gelöst werden? (z. B. „Automatische Klassifikation von Iris-Arten“)

Zielgruppe

Für wen wird die Lösung entwickelt? (z. B. „Botanik-Studenten bei der Feldarbeit“)

Erfolgsmetriken (KPIs)

Wie wird der Erfolg quantitativ gemessen? (z. B. „Genauigkeit > 95 %“)

Deployment-Ziel

Was sind die finalen Artefakte des Projekts? (z. B. „Interaktive Streamlit Web-App“)

Projektdefinition: AMALEA Iris-Projekt

Kontext und Motivation

Der Iris-Datensatz, 1936 von Ronald Fisher in seiner Arbeit „The Use of Multiple Measurements in Taxonomic Problems“ als Beispiel für Diskriminanzanalyse eingeführt, ist ein klassisches Beispiel in Statistik und maschinellem Lernen. Er umfasst Messungen von 150 Stichproben, die sich auf drei verschiedene Iris-Arten (Setosa, Versicolor, Virginica) verteilen, basierend auf vier Features (Kelch- und Blütenblattmaße). Dieser Datensatz ist einer der frühesten und meistgenutzten Benchmark-Datensätze in der ML-Geschichte und seit 1988 im UCI Machine Learning Repository verfügbar. Er eignet sich ideal als Einstiegsprojekt zur Demonstration des QUA³CK-Prozesses.

Trotz seiner relativen Einfachheit bietet der Datensatz ausreichend Komplexität, um verschiedene Algorithmen und Methoden zu vergleichen. Die Möglichkeit der visuellen Klassifikation macht das Projekt zudem für ein breites Publikum zugänglich und ermöglicht eine intuitive Vermittlung der zugrunde liegenden Konzepte. Seine klare Klassentrennung und überschaubare Komplexität machen ihn zu einem idealen Einstiegsdatensatz.



Iris Versicolor



Iris Setosa



Iris Virginica

Projektdefinition im Detail

- **Problem:** Automatische Klassifikation von Iris-Arten basierend auf Blütenmerkmalen.
- **Zielgruppe:** Botanik-Studenten zur schnellen und präzisen Identifikation von Iris-Arten im Feld.
- **Erfolgsmetriken:**
 - Accuracy: > 95 %
 - Prediction Time: < 500 ms
 - Deployment: Öffentliche Streamlit App
- **Lieferobjekte (Deliverables):**
 - Jupyter Notebook mit vollständiger ML-Pipeline.
 - Interaktive Streamlit Webanwendung.
 - Live-Deployment in der Streamlit Cloud.
 - Öffentliches GitHub Repository für das Portfolio.

Die Daten sprechen lassen: Explorative Analyse

In der Phase U (Understanding the Data) analysieren wir den Datensatz, um Muster, Anomalien und Korrelationen zu erkennen. Dies ist entscheidend, um passende Modellierungsansätze zu wählen und potenzielle Probleme frühzeitig zu identifizieren.

Vorgehen am Beispiel des Iris-Datensatzes:

- **Daten laden**

Laden des Standard-Datensatzes von scikit-learn:

```
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target
```

- **Daten inspizieren**

Überprüfung grundlegender Datensatzeigenschaften:

- **Shape:** 150 Zeilen, 5 Spalten
- **Features:** sepal length, sepal width, petal length, petal width
- **Zielklassen:** setosa, versicolor, virginica

- **Statistische Analyse**

Berechnung grundlegender statistischer Kennzahlen (Mittelwert, Standardabweichung, Minimum, Maximum) pro Feature und Klasse.

- **Visualisierung**

Erstellung von Scatter- und Box-Plots zur Visualisierung der Merkmalsverteilungen und der Klassentrennschärfe.



Die explorative Datenanalyse (EDA) ist unerlässlich, um die Struktur und Muster in den Daten zu verstehen, bevor die Modellierung beginnt. Dieser Schritt ermöglicht fundierte Entscheidungen bezüglich Merkmalsauswahl, Vorverarbeitung und Algorithmuswahl.

Am Beispiel des Iris-Datensatzes zeigen einfache Visualisierungen, dass bestimmte Merkmalskombinationen eine gute Trennung der Klassen ermöglichen, während andere stärkere Überlappungen aufweisen. Diese Erkenntnisse sind wertvoll für die Modellauswahl in der nächsten Phase.

Visuelle Analyse und zentrale Erkenntnisse

Die Visualisierung von Merkmalskombinationen liefert entscheidende Erkenntnisse für die Modellierung. Grafische Darstellungen ermöglichen intuitive Einblicke, die bei der Auswahl und Optimierung von Algorithmen unterstützen.

Scatter-Plot (Petal Length vs. Width)

Dieser Plot zeigt eine klare visuelle Trennung der drei Arten. Die Setosa-Klasse ist linear separierbar und deutlich isoliert. Versicolor und Virginica weisen eine geringe Überlappung auf, sind jedoch gut unterscheidbar.

Scatter-Plot (Sepal Length vs. Width)

Hier ist eine signifikant stärkere Überlappung der Klassen zu beobachten, insbesondere zwischen Versicolor und Virginica. Alleinige Sepal-Merkmale würden eine zuverlässige Klassifikation erschweren und zu einer höheren Fehlerrate führen.

Box-Plots

Diese bestätigen, dass die Verteilungen der Petal-Merkmale pro Klasse eine geringere Überlappung aufweisen als die der Sepal-Merkmale. Zudem zeigen die Box-Plots eine geringere Varianz innerhalb der Klassen für Petal-Merkmale.

→ **Schlussfolgerung:** Die **Petal-Merkmale sind die prädiktiv stärksten** und sollten im Modell hoch gewichtet werden. Diese Erkenntnis ist besonders wertvoll für interpretierbare Modelle wie Entscheidungsbäume, bei denen die Petal-Merkmale näher an der Wurzel des Baumes erscheinen würden.

Diese Phase der explorativen Datenanalyse verdeutlicht die Notwendigkeit, Daten vor der Modellierung umfassend zu verstehen. Die gewonnenen Erkenntnisse fließen direkt in die nächste Phase (A³) ein und beeinflussen maßgeblich die Auswahl und Konfiguration der Algorithmen.

Das Herzstück: Algorithmen auswählen und evaluieren

Im iterativen A³-Zyklus werden verschiedene Modelle trainiert und deren Leistungsfähigkeit systematisch bewertet. Diese Phase ist entscheidend für den maschinellen Lernprozess, da hier Algorithmen ausgewählt, Features angepasst und Hyperparameter optimiert werden.

Die drei A's im Detail:

1. **Algorithm Selection:** Auswahl geeigneter Algorithmen basierend auf der Problemstellung und den Dateneigenschaften.
2. **Adapting Features:** Anpassung und Transformation von Merkmalen zur Verbesserung der Modellleistung.
3. **Adjusting Hyperparameters:** Feinabstimmung der Modellparameter zur Optimierung der Performance.

Diese drei Schritte werden typischerweise mehrfach durchlaufen, wobei die Ergebnisse jeder Iteration zur kontinuierlichen Verbesserung beitragen. Im MLOps-Kontext werden diese Experimente systematisch protokolliert, um Reproduzierbarkeit und Vergleichbarkeit zu gewährleisten.

Ansatz im Notebook („AMALEA Big 3“):

Datenaufteilung

Stratifizierte Aufteilung in Trainings- (70 %) und Testdaten (30 %) zur Vermeidung von Bias.

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.3, random_state=42, stratify=y)
```

1

Modellauswahl

Auswahl von drei repräsentativen Algorithmen:

- Decision Tree (baumbasiertes Modell)
- K-Nearest Neighbors (instanzbasiertes Modell)
- K-Means (unüberwachtes Clustering-Modell)

2

MLOps-Integration

Vorbereitung des **MLFlow Experiment Trackings** für die Protokollierung und Reproduzierbarkeit der Ergebnisse.

```
import mlflow  
mlflow.set_experiment("iris_classification")  
with mlflow.start_run(run_name="decision_tree"):  
    # Modelltraining und Evaluierung  
    mlflow.log_param("max_depth", 3)  
    mlflow.log_metric("accuracy", accuracy)
```

3

Leistungsbewertung der „Big 3“

Evaluierungsmethodik

Die trainierten Modelle werden anhand des Testdatensatzes evaluiert, um ihre Leistungsfähigkeit unter realen Bedingungen zu beurteilen. Für die Klassifikationsmodelle nutzen wir die Accuracy als primäre Metrik, da alle Klassen als gleich wichtig und etwa gleich häufig vorkommend angenommen werden.

Für das unüberwachte K-Means-Modell, das ohne Labels arbeitet, verwenden wir den Adjusted Rand Index. Dieser misst die Übereinstimmung zwischen den gefundenen Clustern und den tatsächlichen Klassen.

Alle Experimente werden mit MLFlow protokolliert. Dies gewährleistet die Reproduzierbarkeit und ermöglicht einen systematischen Vergleich der Ergebnisse.



Algorithmus	Metrik	Ergebnis
Decision Tree	Accuracy	0.978
K-Nearest Neighbors	Accuracy	0.978
K-Means	Adjusted Rand Score	0.669

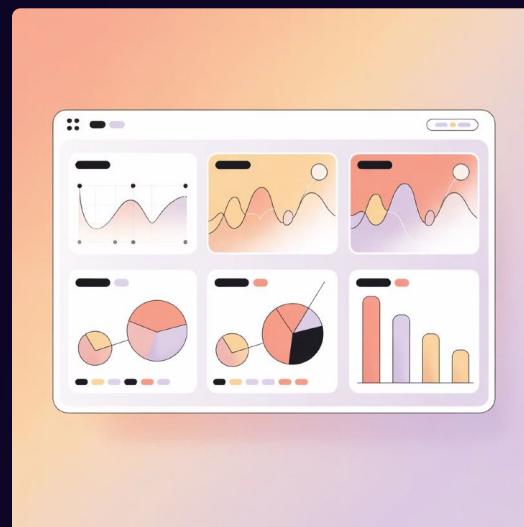
🏆 **Bestes Modell:** Der Decision Tree erzielt die gleiche Genauigkeit wie K-Nearest Neighbors, ist aber potenziell interpretierbarer und effizienter.

📈 **Performance:** Die angestrebte KPI von >95 % Genauigkeit wurde übertroffen. Mit einer Accuracy von 97,8 % erfüllen die Modelle die Anforderungen deutlich.

Das beste Modell auswählen

Phase C (Conclude and Compare) vergleicht systematisch experimentelle Ergebnisse, um fundierte Entscheidungen für das produktive Modell zu treffen. Diese Phase ist entscheidend für die Produktqualität und schlägt die Brücke zwischen experimenteller Forschung und praktischer Anwendung.

Ziel ist es, nicht nur das genaueste Modell zu identifizieren, sondern einen ausgewogenen Kompromiss zwischen quantitativen und qualitativen Kriterien zu finden. Praktisch muss ein Modell neben Genauigkeit auch effizient, interpretierbar und wartbar sein.



Systematischer Modellvergleich

1

Quantitative Metriken

- **Accuracy:** Prozentsatz der korrekt klassifizierten Instanzen.
- **Precision:** Verhältnis der True Positives zu allen positiven Vorhersagen.
- **Recall:** Verhältnis der True Positives zu allen tatsächlich positiven Instanzen.
- **F1-Score:** Harmonisches Mittel aus Precision und Recall.
- **Inferenzzeit:** Zeit für eine einzelne Vorhersage.

2

Qualitative Metriken

- **Modellkomplexität:** Anzahl der Parameter, Baumtiefe etc.
- **Interpretierbarkeit:** Verständlichkeit der Modellentscheidungen.
- **Trainingszeit:** Zeit für das Modelltraining.
- **Wartungsaufwand:** Ressourcen für Modellpflege und Aktualisierungen.

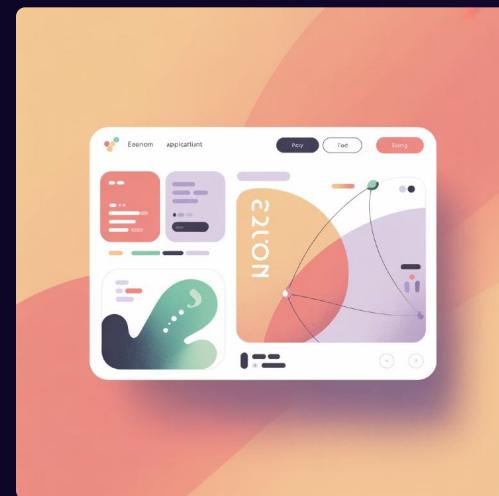
Im MLOps-Kontext ermöglichen Dashboards (z. B. im MLFlow UI) einen schnellen Überblick über alle Experimente. Sie visualisieren Metriken und erleichtern so die Entscheidungsfindung.

Vom Notebook zur Anwendung

Die letzte Phase des „QUA³CK“-Prozesses, der Wissenstransfer (K), gewährleistet die Bereitstellung von Analyseergebnissen an Stakeholder oder die Überführung in produktive Systeme. Diese Phase ist entscheidend, um den vollen Wert der durchgeföhrten Analysen zu realisieren und deren Ergebnisse nutzbar zu machen.

Im akademischen Kontext steht die Dokumentation von Methodik und Ergebnissen im Vordergrund, während in der Praxis die Überführung des Modells in ein produktives System zentral ist. Eine klare Kommunikation und solide technische Umsetzung sind dabei stets unerlässlich.

Im AMALEA-Kurs legen wir besonderen Wert auf die Erstellung eines professionellen Portfolios. Dieses demonstriert die erworbenen Fähigkeiten und dient als Referenz für zukünftige berufliche Chancen.



Dokumentation für das Portfolio

Erstellung einer prägnanten Projektzusammenfassung, die Methodik, Ergebnisse und eingesetzte Technologien darstellt. Dies ist essenziell zur Demonstration der eigenen Kompetenzen.

Vorbereitung für das Deployment

Der Code wird für die nahtlose Integration des trainierten Modells in Anwendungen wie interaktive „**Streamlit Web-Apps**“ strukturiert.

Kommunikation der Ergebnisse

Die „Key Findings“ und der „Business Impact“ werden klar formuliert und zielgruppengerecht aufbereitet, um den Mehrwert des Projekts zu verdeutlichen.



AMALEA Portfolio Summary

Aspekt	Beschreibung
Projekt	AMALEA QUA ³ CK Demo - Iris Classification
Methodik	QUA ³ CK Prozessmodell + Big 3 Algorithmen
Bester Algorithmus	Decision Tree
Performance	97,8 %
Technologien	Python, Pandas, Scikit-learn, Matplotlib
Nächste Schritte	MLFlow Integration + Streamlit Deployment

Diese Zusammenfassung bildet den Kern Ihres Portfolios und kann in verschiedenen Formaten präsentiert werden:

GitHub README

Eine strukturierte README-Datei im GitHub-Repository, die Projekt, Methodik und Ergebnisse detailliert beschreibt. Sie sollte zudem Installations- und Nutzungsanweisungen für den Code bereitstellen.

Streamlit Web-App

Eine interaktive Web-App zur Demonstration des trainierten Modells, die es Nutzern erlaubt, eigene Daten einzugeben und Vorhersagen zu generieren. Die App sollte auch wichtige Projekt- und Methodikinformationen enthalten.

Blog Post

Ein ausführlicher Blogbeitrag, der den gesamten Projektverlauf dokumentiert. Dieser dient als wertvolle Referenz für andere Datenwissenschaftler und kann auf Plattformen wie Medium, LinkedIn oder einem persönlichen Blog veröffentlicht werden.

Ihr Fahrplan zum Erfolg im Assessment

Nutzen Sie das QUA³CK-Modell als Template für Ihre Projekte. Dieses strukturierte Vorgehen ermöglicht die systematische und professionelle Umsetzung komplexer Data-Science-Projekte.

Q - Question

Definieren Sie Problem, Zielgruppe und KPIs. Wählen Sie ein Thema, das Ihre Leidenschaft und Fähigkeiten demonstriert.

K - Knowledge Transfer

Erstellen Sie eine öffentliche Streamlit Cloud App und ein aussagekräftiges GitHub-Repository als Portfolio-Elemente. Kommunizieren Sie Ihre Ergebnisse klar und professionell.



Ihr Portfolio sollte nicht nur technische Fähigkeiten, sondern auch Ihre Fähigkeit demonstrieren, komplexe Probleme strukturiert zu lösen. Das QUA³CK-Modell bietet dafür einen bewährten Rahmen.

U - Understanding

Nutzen Sie große, externe Datenquellen (Kaggle, AWS Open Data) für eine gründliche EDA. Visualisieren und dokumentieren Sie Ihre Erkenntnisse.

A³ - Algorithms

Implementieren Sie die „Big 3“ und erweitern Sie diese um Deep-Learning-Ansätze (neuronale Netze, CNNs, Transformer). Nutzen Sie MLFlow für systematisches Experiment-Tracking.

C - Conclude

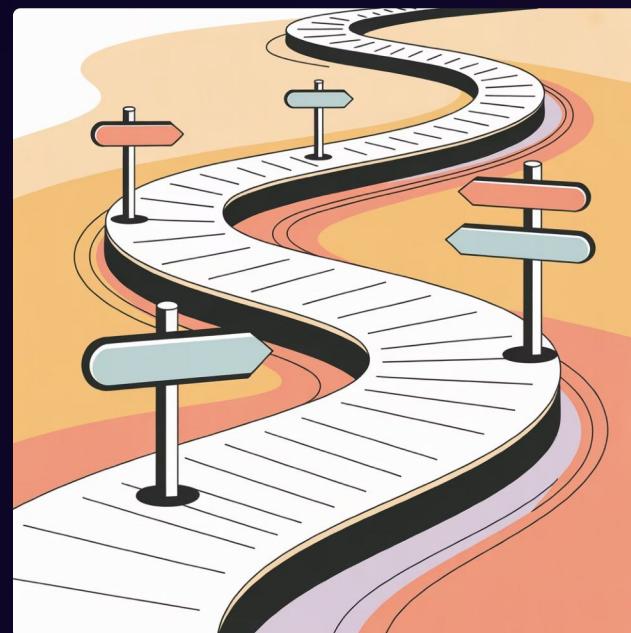
Verwenden Sie ein MLOps-Dashboard für den Modellvergleich. Bewerten Sie Ihre Modelle quantitativ und qualitativ, um fundierte Entscheidungen zu treffen.

Zusammenfassung der Erkenntnisse

In dieser Sitzung haben wir das QUA³CK-Prozessmodell als strukturierten Rahmen für Machine-Learning-Projekte eingeführt. Dieser Ansatz gliedert komplexe Projekte in überschaubare Phasen und ermöglicht eine systematische Umsetzung.

Die Integration von MLOps-Tools wie MLFlow ist zu einem professionellen Standard geworden. Diese Werkzeuge sind in der modernen Praxis unverzichtbar, da sie die Reproduzierbarkeit, Effizienz und Qualität von Data-Science-Projekten signifikant verbessern.

Der strukturierte Ansatz des QUA³CK-Modells bietet die ideale Grundlage für die Erstellung Ihrer Portfolio-Komponenten im AMALEA-Kurs. Durch die konsequente Anwendung dieser Methodik entwickeln Sie hochwertige Projekte, die Ihre Fähigkeiten als Data Scientist eindrucksvoll demonstrieren.



Nächste Schritte:

Woche 2

Entwicklung der Streamlit-App für den Iris-Classifier. Sie lernen, ein trainiertes Modell in eine interaktive Web-App zu integrieren und in der Cloud bereitzustellen.

1

Woche 7

Erstellung eines MLOps-Dashboards mit Model Registry. Sie wenden MLFlow für systematisches Experiment-Tracking und effektive Modellverwaltung an.

2

Woche 4

Vertiefung der A³-Schleife mit den „Big 3“-Algorithmen. Sie erlangen detailliertes Wissen über verschiedene Algorithmen und deren Stärken sowie Schwächen in diversen Anwendungsfällen.

3

Fragen und Antworten

Wie unterscheidet sich QUA³CK von anderen Data-Science-Prozessmodellen?

QUA³CK zeichnet sich durch einen praxisorientierten Ansatz aus, der den gesamten Lebenszyklus von ML-Projekten abdeckt. Im Vergleich zu Modellen wie CRISP-DM liegt der Fokus stärker auf iterativer Algorithmenentwicklung (A³-Schleife) und effektivem Wissenstransfer (K-Phase).

Welche Vorteile bietet MLFlow für das Experiment-Tracking?

MLFlow ermöglicht die systematische Protokollierung von Experimenten, inklusive Code, Parametern, Metriken und Artefakten. Dies steigert die Reproduzierbarkeit, vereinfacht Modellvergleiche und fördert die Teamzusammenarbeit. Im QUA³CK-Prozess gewährleistet dies nachvollziehbare und vergleichbare Ergebnisse.

Wie kann ich QUA³CK auf komplexere Projekte anwenden?

Das QUA³CK-Modell ist skalierbar und eignet sich für Projekte unterschiedlicher Komplexität. Bei komplexeren Vorhaben sind die konsequente Anwendung der A³-Schleife und die Integration von MLOps-Praktiken entscheidend. Für sehr große Projekte empfiehlt sich eine Unterteilung in agile Teilschritte.

Glossar

Hier finden Sie eine Übersicht über die wichtigsten Fachbegriffe und Konzepte, die in dieser Präsentation besprochen wurden. Sie sind nach Kategorien geordnet, um die Übersichtlichkeit zu verbessern.

QUA³CK-Prozess

Q – Question (Fragestellung)	Die erste Phase des QUA ³ CK-Modells, in der Problem, Zielgruppe und Key Performance Indicators (KPIs) definiert werden. Hier wird das Thema des Data-Science-Projekts festgelegt.
U – Understanding the data (Datenverständnis)	Phase der gründlichen Explorativen Datenanalyse (EDA) großer, externer Datenquellen, um Erkenntnisse zu gewinnen. Visualisierung und Dokumentation sind hier entscheidend.
A³ – Algorithms	Steht für Algorithm selection, Adapting features, Adjusting hyperparameters. In dieser iterativen Phase werden Machine-Learning-Algorithmen implementiert, Features angepasst und Hyperparameter optimiert, oft unter Nutzung von MLFlow.
C – Conclude and compare (Schlussfolgerung und Vergleich)	Phase des Modellvergleichs und der Bewertung. Hier werden Modelle quantitativ und qualitativ bewertet, um fundierte Entscheidungen über die beste Lösung zu treffen.
K – Knowledge transfer (Wissenstransfer)	Die letzte Phase, in der die Ergebnisse des Projekts kommuniziert und in greifbare Portfolio-Elemente wie eine Streamlit App oder ein GitHub-Repository überführt werden.

Machine Learning Begriffe

EDA (Explorative Datenanalyse)	Prozess zur Untersuchung von Datensätzen zur Zusammenfassung ihrer Hauptigenschaften, oft mit visuellen Methoden.
Feature Engineering	Der Prozess der Erstellung neuer Features aus bestehenden Rohdaten, um die Leistung von Machine-Learning-Modellen zu verbessern.
Hyperparameter	Parameter, die den Lernprozess und die Struktur des Modells steuern und vor dem Training festgelegt werden (z.B. Lernrate, Anzahl der Bäume in einem Random Forest).
Overfitting	Tritt auf, wenn ein Modell die Trainingsdaten zu genau lernt und dabei auch Rauschen oder irrelevante Muster aufnimmt, wodurch die Fähigkeit zur Generalisierung auf neue Daten leidet.
Underfitting	Tritt auf, wenn ein Modell nicht komplex genug ist, um die zugrunde liegende Struktur der Daten zu erfassen, und sowohl auf Trainings- als auch auf Testdaten schlecht abschneidet.
Train-Test-Split	Die Aufteilung eines Datensatzes in einen Trainingsatz (zum Trainieren des Modells) und einen Testsatz (zum unabhängigen Bewerten der Modellleistung).
Cross-Validation	Eine Technik zur Bewertung der Modellleistung und zur Vermeidung von Overfitting, indem der Datensatz in mehrere Teilmengen (Folds) aufgeteilt wird und das Modell mehrmals auf verschiedenen Kombinationen von Trainings- und Validierungsdaten trainiert und getestet wird.
Confusion Matrix	Eine Tabelle, die die Leistung eines Klassifikationsmodells zusammenfasst, indem sie die Anzahl der korrekt und falsch klassifizierten Instanzen pro Klasse anzeigt.
Accuracy, Precision, Recall, F1-Score	Gängige Metriken zur Bewertung der Leistung von Klassifikationsmodellen, abgeleitet aus der Confusion Matrix, die jeweils unterschiedliche Aspekte der Modellqualität hervorheben.
Decision Tree	Ein Klassifikations- oder Regressionsalgorithmus, der eine baumartige Struktur von Entscheidungen verwendet, um Vorhersagen zu treffen.
Random Forest	Ein Ensemble-Lernverfahren, das mehrere Entscheidungsbäume verwendet und deren Ergebnisse kombiniert, um die Vorhersagegenauigkeit und Robustheit zu verbessern.
Logistic Regression	Ein statistisches Modell, das zur Vorhersage der Wahrscheinlichkeit eines binären Ergebnisses verwendet wird, oft für Klassifikationsaufgaben.
Classification	Eine Art des überwachten Lernens, bei der ein Modell lernt, Datenpunkte einer von mehreren vordefinierten Klassen zuzuordnen.
Supervised Learning	Ein Machine-Learning-Paradigma, bei dem ein Algorithmus anhand von gelabelten Daten (Eingaben mit bekannten Ausgaben) lernt, Muster zu erkennen und Vorhersagen zu treffen.

MLOps Begriffe

MLOps (Machine Learning Operations)	Eine Reihe von Praktiken zur Standardisierung und Rationalisierung der Entwicklung, des Einsatzes und der Wartung von Machine-Learning-Modellen in der Produktion.
MLFlow	Eine Open-Source-Plattform zur Verwaltung des gesamten Machine-Learning-Lebenszyklus, einschließlich Experiment Tracking, Reproduzierbarkeit und Modellbereitstellung.
Experiment Tracking	Der Prozess des systematischen Protokollierens und Organisieren von Informationen zu Machine-Learning-Experimenten, wie Parameter, Metriken, Code-Versionen und Artefakte.
Model Registry	Ein zentrales Repository in MLOps-Plattformen (wie MLFlow) zur Verwaltung des Lebenszyklus von Machine-Learning-Modellen, einschließlich Versionierung, Staging und Archivierung.
Deployment	Der Prozess, ein trainiertes Machine-Learning-Modell in eine Produktionsumgebung zu überführen, wo es genutzt werden kann, um Vorhersagen zu treffen.
CI/CD (Continuous Integration/Continuous Deployment)	Praktiken aus der Softwareentwicklung, die auf MLOps angewendet werden, um den Entwicklungsprozess zu automatisieren und die schnelle und zuverlässige Bereitstellung von Modellen zu ermöglichen.
Reproducibility (Reproduzierbarkeit)	Die Fähigkeit, die Ergebnisse eines Machine-Learning-Experiments unter Verwendung der gleichen Daten, des gleichen Codes und der gleichen Konfiguration exakt zu replizieren.
Model Drift	Der Rückgang der Vorhersagegenauigkeit eines Machine-Learning-Modells im Laufe der Zeit, verursacht durch Änderungen in den zugrunde liegenden Datenverteilungen (Konzeptdrift) oder der Beziehung zwischen Eingaben und Ausgaben (Datendrift).

Tools & Technologien

Streamlit	Eine Open-Source-Python-Bibliothek, die es Data Scientists und Machine-Learning-Ingenieuren ermöglicht, interaktive Webanwendungen für Datenanalysen und Modell-Demonstrationen schnell zu erstellen und bereitzustellen.
Jupyter Notebook	Eine interaktive Entwicklungsumgebung, die es ermöglicht, Code, Text (Markdown), Gleichungen und Visualisierungen in einem einzigen Dokument zu kombinieren, ideal für Datenexploration und Prototyping.
Python	Eine weit verbreitete, interpretierte und objektorientierte Programmiersprache, die in Data Science und Machine Learning aufgrund ihrer umfangreichen Bibliotheken sehr beliebt ist.
Scikit-learn	Eine kostenlose Machine-Learning-Bibliothek für Python, die verschiedene Algorithmen für Klassifikation, Regression, Clustering und Dimensionsreduktion sowie Tools zur Modellbewertung bietet.
Pandas	Eine Open-Source-Python-Bibliothek zur Datenmanipulation und -analyse. Sie bietet Datenstrukturen wie DataFrames, die tabellarische Daten effizient verarbeiten können.

Literatur & Ressourcen

Wissenschaftliche Publikationen

- Stock, S., Becker, J., Grimm, D., Hotfilter, T., Molinar, G., Stang, M., & Stork, W. (2021). QUA³CK – A Machine Learning Development Process. Karlsruher Institut für Technologie (KIT). <https://publikationen.bibliothek.kit.edu/1000129631>
- Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7(2), 179-188.
- Gartner Research (2017-2019). Data Science Project Failure Rates and Analytics Insights.
- Grand View Research (2024). MLOps Market Size, Share & Trends Analysis Report. <https://www.grandviewresearch.com/industry-analysis/mlops-market-report>
- VentureBeat (2019). Why do 87% of data science projects never make it into production?

Große frei verfügbare Datensätze

- UCI Machine Learning Repository: <https://archive.ics.uci.edu/ml/index.php>
 - Iris Dataset: <https://archive.ics.uci.edu/ml/datasets/iris>
- Kaggle Datasets: <https://www.kaggle.com/datasets>
- Google Dataset Search: <https://datasetsearch.research.google.com>
- AWS Open Data Registry: <https://registry.opendata.aws>
- Data.gov (US Government Open Data): <https://data.gov>
- European Data Portal: <https://data.europa.eu>
- OpenML: <https://www.openml.org>

MLOps & Tools Dokumentation

- MLFlow Documentation: <https://mlflow.org/docs/latest/index.html>
- Streamlit Documentation: <https://docs.streamlit.io>
- Scikit-learn Documentation: <https://scikit-learn.org/stable/>
- Pandas Documentation: <https://pandas.pydata.org/docs/>

Weiterführende Ressourcen

- CRISP-DM Methodology: <https://www.datascience-pm.com/crisp-dm-2/>
- KDD Process: <https://www.kdnuggets.com>
- Papers with Code (ML Research): <https://paperswithcode.com>

