Testing The Untestable - Risikobasierte Qualitätssicherung für Machine-Learning Systeme

Michael Felderer
Universität Innsbruck

Zusammenfassung

Machine-Learning Systeme durchdringen zunehmend alle Lebensbereiche. Dadurch gewinnt auch die Qualität und in weiterer Folge das Testen dieser Systeme an Bedeutung. Die inhärente Unsicherheit von Machine-Learning Systemen macht risikobasierte Testansätze besonders geeignet, welche in diesem Beitrag vorgestellt werden.

1 Überblick

Unter einem Machine-Learning System (kurz ML-System) versteht man ein softwarebasiertes System, welches neben klassischen Komponenten auch Machine-Learning Komponenten (ML-Komponenten) beinhaltet. Somit kann sowohl ein Empfehlungssystem in einem Onlineshop, welches aus Daten die vorgeschlagenen Artikel lernt, als auch ein intelligent gesteuertes autonomes Fahrzeug als ML-System verstanden werden.

Beim Machine Learning (ML) lernt man in einer Trainingsphase ein stochastisches Modell aus Trainingsdaten, welches dann mit einer bestimmten Genauigkeit Vorhersagen für neue Eingabedaten machen kann. Wenn das Modell auf einem künstlichen neuronalen Netz mit verborgenen Schichten basiert, dann spricht man von Deep Learning. Moderne ML-Komponenten unterscheiden sich fundamental von traditionellen Softwarekomponenten: zum einen lernen ML-Komponenten aus Daten und sind deshalb inhärent nicht-deterministisch und stochstisch; zum anderen können sich diese autonom durch die Bereitstellung neuer Daten weiterentwickeln. Aufgrund dieser Unterschiede werden ML-Komponenten (und daraus resultierende Systeme) auch anders entwickelt und getestet: das Aufbereiten von Daten sowie die Auswahl, das Trainieren und Validieren von ML-Modellen erfolgt typischerweise durch Data Scientists oder ML-Engineers. Diese verwenden teilweise auch andere Begrifflichkeiten als im Softwaretesten: "Regression" steht im ML-Kontext etwa nicht für das Regressionstesten, sondern für statistische Regressionsmodelle [1]. Dieser Beitrag basiert auf [1], wo wir die automatisierte Qualitätssicherung für ML-Systeme diskutieren. In diesem Beitrag geben wir zunächst einen kurzen Überblick über Qualitätssicherungstechniken für ML-Systeme und behandeln anschließend entsprechende risikobasierte Testansätze.

2 Qualitätssicherungstechniken für ML-Systeme

Qualitätssicherungstechniken für ML-Systeme können entlang der drei Dimensionen Artefakttyp, Qualitätseigenschaft und Qualitätssicherungsart klassifiziert werden (siehe Abbildung 1).

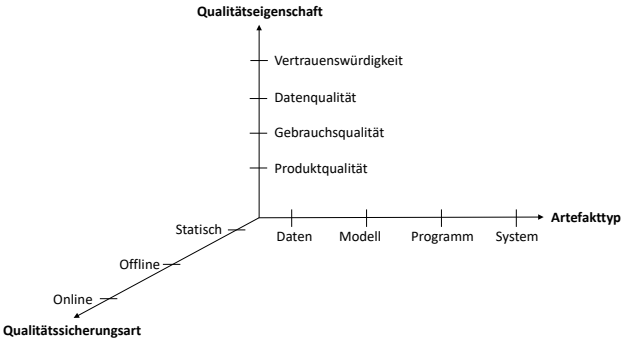

Abbildung 1: Dimensionen zur Klassifikation von Qualitätssicherungstechniken für ML-Systeme [1]

Die Artefakttypen, welche getestet werden müssen, sind die zugrundeliegenden Daten, das ML-Modell, das Programm etwa zur Einbindung von Bibliotheken sowie das System, welches klassiche und ML-Komponenten integriert.

Qualitätseigenschaften legen das Ziel und die Vorgehensweise von Qualitätssicherungsmaßnahmen fest. Neben der klassischen Produkt- und Gebrauchsqualität, spielt insbesondere die Datenqualität eine große Rolle. Darüber hinaus spielt auch die Vertrauenswürdigkeit (Trustworthiness) eine große Rolle.

Die Art der Qualitätssicherung kann statisch erfolgen ohne ein Artefakt auszuführen, etwa wenn Konsistenzprüfungen direkt auf Daten durchgeführt werden. Die Qualitätssicherung erfolgt offline, wenn ein Artefakt isoliert ausgeführt wird, etwa wenn ein ML-Modell mit historischen Daten ohne das Umgebungsverhalten zu berücksichtigen getestet wird. Die Qualitätssicherung erfolgt online, wenn Artefakte in einem Anwendungskontext und in Interaktion mit der Umgebung ausgeführt werden.

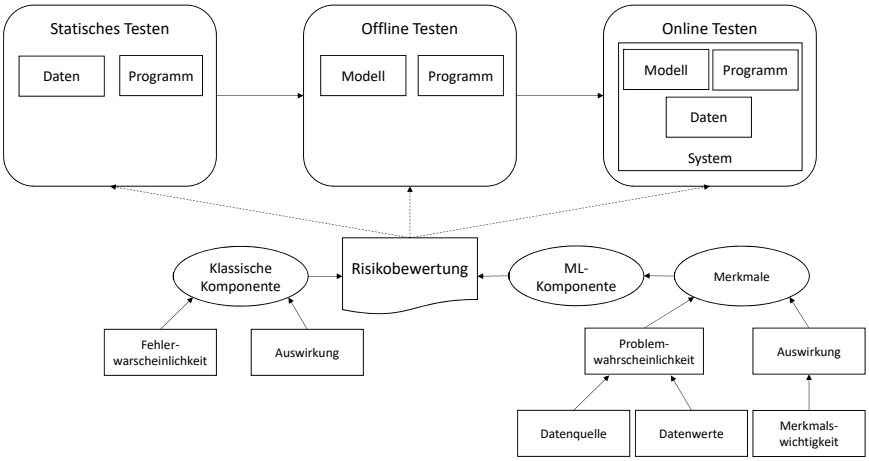

Abbildung 2: Überblick über das risikobasierte Testen von ML-Systemen [1]

3 Risikobasiertes Testen von ML-Systemen

Für das effiziente und effektive Testen klassischer Softwaresysteme hat sich das risikobasierte Testen etabliert [2]. Dabei wird eine Bewertung der Produktrisiken durchgeführt, um eine risikobasierte Teststrategie zu entwickeln. Diese erlaubt es alle Phasen des klassischen Testprozesses zu steuern. Die Risikobewertung integriert die Fehlerwahrscheinlichkeit der zu testenden Elemente mit der Auswirkung bei Eintritt eines Fehlers. Dadurch wird es ermöglicht, dass kritische Fehler früh gefunden werden bzw. die Testressourcen zielgerichteter eingesetzt werden können.

Die risikobasierte Vorgehensweise ist auch sehr hilfreich, um eine Qualitätssicherungsstrategie für ML-Systeme zu entwickeln, die etwa die Testautomatisierung für das Statische Testen sowie das Offline- und Online-Testen regelt. Wir betrachten hier nur die Risikobewertung für ML-Komponenten genauer - jene von klassischen Komponenten kann mit traditionellen risikobasierten Testansätzen adressiert werden [2]. Mit dem risikobasierten Ansatz können grundsätzlich alle in Abbildung 1 dargestellten Qualitätseigenschaften adressiert werden.

Abbildung 2 zeigt einen Überblick über die zentralen Qualitätssicherungsaktivitäten (Statisches Testen, Offline Testen, Online Testen) für die unterschiedlichen Artefakte (Daten, Modell, Programm, System) in einem ML-System und einen möglichen Risikobewertungsansatz. Dreh- und Angelpunkt der Risikobewertung für ML-Komponenten sind einzelne Merkmale oder Merkmalsgruppen. Diese Merkmale (im ML-Kontext typischerweise “Features” genannt) sind messbare Eigenschaften, die als unabhängige Eingangsvariablen in ML-Modellen zur Vorhersage abhängiger Variablen oder zur Identifikation von Mustern dienen. Bei der Bilderkennung sind solche Merkmale etwa Bildeigenschaften wie Farben oder Objekte in bestimmten Bildregionen. Bei der Wettervorhersage sind es etwa die Luftfeuchtigkeit und die Aussentemperatur. Merkmale haben konkrete Datenwerte und stammen aus Datenquellen (etwa von Sensoren oder aus Datenbanken), welche diese Werte kontinuierlich liefern oder abspeichern. Diese Informationen über Merkmale können verwendet werden, um problematische Merkmale bzw. Merkmalsbereiche zu identifizieren. Also etwa Randwerte oder Merkmale, die aus unstabilen Datenquellen stammen. Gemeinsam mit der Wichtigkeit von Merkmalen oder Merkmalsgruppen kann so ein qualitatives oder sogar quantitatives Risikomodell aufgebaut werden. Ein Risikomodell für das (statische) Testen von Daten in einem ML-System wird in [3] vorgestellt. Ein Risikomodell für das Online-Testen eines kollaborativen ML-Systems, bei dem ein Roboter kollaborativ von einem Menschen lernt und mit ihm gemeinsam Aufgaben ausführt, wird in [4] vorgestellt.

Literatur

[1] Michael Felderer. Automatisierte qualitätssicherung KI-basierter Systeme: Grundlegende Begriffe und Methoden. *German Testing Magazin*, 1/2021, 2021.

[2] Michael Felderer and Ina Schieferdecker. A taxonomy of risk-based testing. *International Journal on Software Tools for Technology Transfer*, 16(5):559–568, 2014.

[3] Harald Foidl and Michael Felderer. Risk-based data validation in machine learning-based software systems. In *3rd ACM SIGSOFT international workshop on machine learning techniques for software quality evaluation*, pages 13–18, 2019.

[4] Jubril G Adigun, Matteo Camilli, Michael Felderer, Andrea Giusti, Dominik T Matt, Anna Perini, Barbara Russo, and Angelo Susi. Collaborative artificial intelligence needs stronger assurances driven by risks. *Computer*, 55(3):52–63, 2022.