

Fallstudie

Model Engineering (DLMDWME01)

im Master Studiengang Data Science

**Vorhersagemodell für den täglichen Bedarf an Bereitschaftsfahrenden  
beim DRK Berlin**

Verfasser: Georg Grunsky

Matrikelnummer: IU14072015

Tutor: Markus Pak

Datum: 26. Februar 2025

## I Abstract

Der Berliner Rot Kreuz Rettungsdienst (DRK Berlin) projiziert den Einsatz eines Machine Learning Systems zur effizienten Planung eines Bereitschaftsdienstplanes für Einsatzfahrer:innen. Auf Basis einer erfolgten Use Case Analyse, stellt diese Arbeit den Projektplan vor und behandelt sowohl die Projektstruktur und -teamplanung als auch die Zeitplanung, die Kostenschätzung, die Stakeholderanalyse und das Risikomanagement des Projektes.

Für den Projektmanagement-Ansatz wird eine hybride Vorgehensweise verfolgt. Agile Methoden, wie Scrum, werden mit klassischen Elementen des V-Modell XT kombiniert. Das schafft eine klare Projektstruktur mit definierten Phasenübergängen und Planungswerkzeugen und gewährleistet gleichzeitig die Vorteile der flexiblen und iterativen Entwicklung. Das V-Modell XT wurde auf das notwendigste beschränkt um ein kleines Projektteam, einen geringen „Overhead“ und eine modulare Struktur zu ermöglichen.

Mit den verwendeten Elementen des V-Modell XT wurde in manchen Bereichen zwar über die Vorgaben der Aufgabenstellung hinausgeschossen, die Dokumentation des Rahmenprozesses bietet jedoch eine hervorragende Unterstützung bei gleichzeitiger Ermutigung zur Flexibilität. Man bekommt das Gefühl, für den Projektstart „bereit“ zu sein. Der, in der Aufgabenstellung verlangte, Projektstrukturplan kommt in der Dokumentation des V-Modell XT (Angermeier et al. 2024) nicht vor. Leider ist auch in den vorgestellten Modellen des Studienmoduls „Management von IT-Projekten“ (IU Internationale Hochschule 2024), das diesem Modul üblicherweise vorangeht, kein Projektstrukturplan abgebildet. Den ersten zufriedenstellenden Anhalt dazu lieferte Wikipedia (2022). Beim Projektplan finden die Anforderungen der Aufgabenstellung sowie das V-Modell XT aber wieder zusammen. Als mögliche Ansatzpunkte wurden weiters The PM Minimalist Quick Start Guide (Greer 2011) sowie die Lektüre von How Big Things Get Done (Flyvbjerg / Gardner 2023) herangezogen, diese lieferten jedoch nicht die gehoffte Unterstützung für die Aufgabe.

Die inhaltliche Vorgabe des Projektes ist es, kosteneffizient die tägliche Anzahl des benötigten Bereitschaftspersonals vorherzusagen, gleichzeitig jedoch nie zu wenig Einsatzfahrende als Reserve vorzusehen. Das System und das Projekt bauen sich rund um diese Vorgabe auf. Die Planung sieht vor mit einem Entwicklungsteam in mehreren, teils parallelen Sprints, diese Minimalanforderung umzusetzen, sinnvoll um weitere Funktionen zu erweitern und das System, mit entsprechender Hardware und benutzerfreundlichen Frontends, in eine stabile Umgebung einzubetten, die sich gut in den IT-Betrieb des DRK Berlin integrieren lässt. Die wahrscheinlich langen Lieferzeiten der Hardwarekomponenten geben der Entwicklung einen zeitlichen Rahmen für die Umsetzung der Sprints. Inhaltlich wird dieses Projekt nicht als „zeitkritisch“ eingestuft. Gegebenenfalls wirkt sich daher eine Verzögerung zwar auf die, der Kostenstelle des Projektes zugeschriebenen, Personalkosten jedoch nicht auf sonstige Aspekte und Abhängigkeiten der Umsetzung aus. Der Fokus der Entwicklung liegt, auch in der Priorisierung der Arbeitspakete, auf den, für das Machine Learning Modell relevanten Aufgaben. Das Vorhersagemodell selbst bildet das Herzstück der Aufgabe. In der Zeit- und Kostenschätzung wurde

daher der Einsatz eines/r weiteren Data Scientist zwar erwogen, jedoch auch wieder verworfen.

Anhand der exemplarisch dargestellten Risiken, ist auch ersichtlich, dass nicht nur ein potentieller Reputationsschaden für das DRK (aufgrund einer zu niedrig prognostizierten Anzahl nötiger Bereitschaftsfahrender und damit mangelnder Einsatzbereitschaft) das Projekt gefährden kann, sondern auch eine mögliche Ablehnung durch die Nutzer, die durch die Umsetzung mehr Verantwortung übernehmen müssen als in den derzeitigen Arbeitsprozessen. Dies sind u.a. zwei wesentliche Punkte, denen im Rahmen der Projektumsetzung, bei der Gestaltung des Modells und der Regelung der zukünftigen Arbeitsprozesse, unbedingt Rechnung getragen werden muss. Die zukünftigen Nutzer werden daher bereits ab Projektbeginn sowohl für das Requirements Engineering als auch für die Qualitätssicherung hinzugezogen.

## I.1 Making of

Ausgangspunkt dieser Arbeit war ein MachineLearning Szenario, das dem Modul „Model Engineering“ (Pak 2024) dieses Studienganges entnommen wurde. Dank der Zustimmung der zuständigen Tutoren wurde es möglich, dasselbe Thema in drei unabhängigen aber inhaltlich dennoch aneinander anschließenden Modulen zu bearbeiten. Dadurch ergab sich eine durchgängige Bearbeitung des Problemstellung durch mehrere Entstehungsphasen eines Machine Learning Systems hindurch.

1. **Modul „Data Science UseCase“** : Analyse des Anwendungsfalls (ML-Canvas) bis zur Präsentation für die Freigabe durch das Präsidium.
2. **Modul „Technische Projektplanung“** : die gegenständliche Arbeit
3. **Modul „Model Engineering“** : Die Umsetzung des Vorhersagemodells mit bereitgestellten Trainingsdaten

Die ganzheitliche Betrachtungsweise der Thematik ließ bereits jetzt einen besseren Einblick in die Prozesse und Herausforderungen der Softwareprojektentwicklung mit Machine Learning Aspekten entstehen.

Für die technische Bearbeitung der Aufgabenstellung wurden mehrere Werkzeuge verwendet. In vorangegangenen Modulen hat sich bereits die Kombination eines Git-Repositories für die LATEX-Dateien der Bearbeitung bewährt. Zusätzlich wurde probeweise das Repository auch einem Git-Projekt hinzugefügt um die Zeitschätzung des Projektes in Form von Iterations, Meilensteinen und Issues über das Git-Projekt zu gestalten. Dies erwies sich zwar als machbar, aber nicht als „sauber“, da im Repository, die tatsächlichen Issues der Portfoliophasen mit den fiktiven Issues des Projektes vermischt wurden. Eine Testversion der Software „objectiF RPM“ (microTool GmbH 2024) unterstützte zwar der Entwicklung eines Verständnisses für die Herausforderungen in der Projektplanung, eine sinnvolle Herangehensweise und auch für Details der Umsetzung (wie zB Reviewmöglichkeiten für Anforderungen und „Slicing“ von Use Cases), war jedoch für die tatsächliche Verwendung im Rahmen der Portfolioaufgabe zu umfangreich und hätte den Rahmen gesprengt. Das GANTT-Diagramm wurde anschließend mit dem relativ intuitiv gehaltenem Tool „YouTrack“ von JetBrains erstellt. Gerade, die Möglichkeit die Arbeitspakete hierarchisch zu strukturieren und zeitlich einfach anzupassen, machten das GANTT-Diagramm zu einem wertvollen Werkzeug, dass gute Einblicke in die Erfordernisse der Projektentwicklung bot.

# Inhaltsverzeichnis

<b>I Abstract</b>	<b>II</b>
I.1 Making of . . . . .	III
<b>II Abkürzungsverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Das Vorhersagemodell</b>	<b>2</b>
2.1 Organisiere . . . . .	2
2.1.1 Business Understanding . . . . .	2
2.1.2 Data Acquisition and Understanding . . . . .	2
2.1.3 Modeling . . . . .	2
2.1.4 Deployment . . . . .	2
2.1.5 Customer Acceptance . . . . .	2
2.1.6 Ordnerstruktur . . . . .	3
2.2 Beurteile die Qualität . . . . .	3
2.3 erstes Basismodell . . . . .	3
2.4 Modell . . . . .	3
2.4.1 Variablen . . . . .	4
2.4.2 Modellresultate . . . . .	4
2.4.3 Fehleranalyse . . . . .	4
2.5 Vorschlag . . . . .	4
<b>3 Zusammenfassung</b>	<b>6</b>
<b>4 Literaturverzeichnis</b>	<b>7</b>
<b>5 Abbildungsverzeichnis</b>	<b>8</b>
<b>6 Tabellenverzeichnis</b>	<b>9</b>

## II Abkürzungsverzeichnis

<b>AI</b>	Artificial Intelligence
<b>KI</b>	Künstliche Intelligenz
<b>GPT</b>	Generative Pre-trained Transformer Model
<b>PPO</b>	Proximal Policy Optimization
<b>NLP</b>	Natural Language Processing
<b>RLHF</b>	Reinforcement Learning from Human Feedback
<b>FAQ</b>	Frequently Asked Questions

# 1 Einleitung

Conversational interfaces are popular, but they are far from intelligent. Amazon' Alexa and other voice-response interfaces don't understand language. They simply launch computerized sequences in response to sonic sequences, which humans call verbal commands.

Die Abkürzung „GPT“ steht hierbei für Generative Pre-trained Transformer Model (GPT). In einem Interview mit Melanie Subbiah, einer Autorin des ersten GPT-3 Konzepts und Artificial Intelligence (AI)-Ingenieur bei OpenAI, wird die Bedeutung der Abkürzung anschaulich erklärt:

Im gewählten Szenario geht es um die Herausforderung, die Planungslogik des Bereitschaftsdienstplans für den Berliner Rotkreuz-Rettungsdienst zu verbessern. Das, weiter unten beschriebene, Thema ist dem Modul „Model Engineering“ des Masterstudienganges Data Science entnommen (Pak 2024), und wurde für zwei weitere Module adaptiert:

1. **Data Science UseCase:** Analyse des Anwendungsfalls bis zur Präsentation für die Freigabe durch das Management.
2. **Technische Projektplanung:** Aufbauend auf dem PitchDeck vom Projektstrukturplan über eine Kostenschätzung und das Risikomanagement bis zur Stakeholder-Analyse.
3. **Model Engineering:** Die Umsetzung des UseCase als Projekt in einem Git-Repository mit bereitgestellten Daten bis zu einem fertiggestellten Vorhersagemodel.

Die Motivation, dieses Thema auch im Modul „Data Science UseCase“ zu bearbeiten, besteht darin, die Problemstellung aus mehreren Blickwinkeln zu behandeln und so ein durchgängiges Verständnis für die Abwicklung realer Szenarien zu erlangen. Die ganzheitliche Betrachtungsweise der Thematik lässt einen tieferen Einblick in die Prozesse der Data Science entstehen, erhöht die Identifikation mit dem UseCase und macht somit schlussendlich auch mehr Spaß.

## **2 Das Vorhersagemodell**

### **2.1 Organisiere**

das Projekt mithilfe der CRISP-DM oder der MS Team Data Science Methode. Mache einen Vorschlag, wie die Ordnerstruktur eines Git-Repositories für das Projekt aufgebaut werden soll. Beachte, dass Du den finalen Code des Projekts nicht nach dieser Ordnerstruktur aufbauen musst.

Warum für TDSP entschieden moderner - optimiert für agile Entwicklung - gegenständliches Projekt hat ebenfalls Anfang und Ende - flexiblere Gestaltung des Arbeitsprozesses da alle großen „Überschriften“ von einander abhängen (oder so...)

Der Link zu TDSP aus dem Studienscript, aber auch von den referenzierenden GitHub-Seiten aus ist auf Microsoft nicht mehr verfügbar. Das stattdessen angezeigte Dokument AI-Implementierung enthält zwar einen Verweis auf TDSP mit einem Link, dieser führt jedoch ebenfalls wieder zum selben AI-Implementierungsdokument. => Anm: schnell verändernde Branche

Schritte TDSP (?)

#### **2.1.1 Business Understanding**

(Referenz auf vorangegangene Arbeiten)

#### **2.1.2 Data Acquisition and Understanding**

Referenz UseCase Analyse Bestehende Daten und Hinweise auf mögliche Erweiterung Explorative Datenanalyse

#### **2.1.3 Modeling**

Referenz UseCase Analyse Referenz Monitoring (UseCase Analyse)

#### **2.1.4 Deployment**

Referenz Monitoring (UseCase Analyse) Referenz Projektplanung

#### **2.1.5 Customer Acceptance**

Referenz Risiko in der Projektplanung

## 2.1.6 Ordnerstruktur

Git Link Das Programm wurde vom Autor mit dem Namen „*functionfinder*“ betitelt und steht auf GitHub unter dem Link

<https://github.com/GGProjects/DLMDWMP01>

zum Download zur Verfügung.

Die weitere Ordnerstruktur orientiert sich an einem Blog-Post von ?, der, nach Meinung des Autors, ein leicht verständliches Framework für Pythonprojekte bietet. Bereitgestellt über (?)

Fusioniert mit (?) ... Umsetzung nicht in einer Azure Umgebung, daher diesbezgl modifiziert.

In der nachfolgenden Abbildung werden die wesentlichen Ordner und Dateien der bereitgestellten Programmstruktur dargestellt. Hervorzuheben ist hierbei das Package *functionfinder*, das sämtliche Pythonmodule beinhaltet, die für die Programmfunktionalität verantwortlich sind, sowie das Modul *ffrunner*, über das die eigentliche Ausführung des Programmes gestartet wird. Die übrigen Verzeichnisse des Projekts stehen für die bereitgestellten Daten (*data*), die Dokumentation (*docs*), die vom Programm erzeugten Ausgaben (*output*), sowie für die vorgesehenen UnitTests (*tests*) zur Verfügung.

Ordnerstruktur ist für den Bedarf dieses Projektes zu umfangreich, Idee ist aber ein template für zukünftige Projekte zu schaffen, wo Modelle in eine Gesamtsoftware eingebettet sind. Die einzelnen Modell- und Packageordner könnten auch als eigene Git-Repos angelegt werden. Die Nummerierung sorgt für gleichbleibende Anordnung und leicht verwaltbare Strukturen

Ordnerstruktur ist sicher noch ausbaufähig, Änderungen werden sich noch ergeben.

## 2.2 Beurteile die Qualität

des zur Verfügung gestellten Datensatzes. Bereite Deine Erkenntnisse so auf und visualisiere sie so, dass Businesspartner in einer klaren und einfachen Weise die wichtigen Zusammenhänge verstehen können.

Stelle ein

## 2.3 erstes Basismodell

(ein sogenanntes Baseline-Modell) auf, sowie ein präzises Vorhersagemodell, das den Businessanforderungen genügt, nämlich die Erfolgsrate der Kreditkartenzahlungen zu erhöhen und gleichzeitig die Transaktionskosten gering zu halten.

Damit die Businesspartner Vertrauen in Dein neues

## 2.4 Modell

entwickeln, solltest du die Wichtigkeit der einzelnen erklärenden



### **2.4.1 Variablen**

diskutieren und die

### **2.4.2 Modellresultate**

so interpretierbar wie möglich gestalten. Außerdem ist eine detaillierte

### **2.4.3 Fehleranalyse**

sehr wichtig, damit die Businesspartner auch die Schwachstellen Deiner Herangehensweise verstehen.

Im letzten Schritt des Projekts soll ein

## **2.5 Vorschlag**

unterbreitet werden, wie Dein Modell in die tägliche Arbeit des Fachbereichs eingebunden werden kann, beispielsweise wie eine graphische Benutzeroberfläche (GUI) aussehen könnte.

Abb. 1: Struktur des Ordneraufbaus

```

DMLDWME01
|
+---00_docs                                # Dokumente
|   +---01_general_basics                  # unspezifische Grundlagen
|   +---02_project_basics                  # projektspezifische Grundlagen
|   \---03_artifacts                       / Projektprodukte
|       +---01_paper                       # Latex Files dieser Arbeit
|       +---02_data_reports                # Reports der Datenanalyse
|       |   \---00_dictionaries             # Datenschema Beschreibungen
|       |
|       \---03_model_reports                # Reports der Modellerstellung
|       \---[...models]                    # Subfolder einzelner Modelle
|
+---01_data                                # Datenbereitstellung
|   +---01_lake                            # Data Lake – Rohdatenspeicher
|   |   \---01_use_case_2                  # Rohdaten dieser Aufgabenstellung
|   |
|   \---02_warehouse                       # Data Warehouse – Aufbereitete Daten
|
+---02_config                             # Konfigurationsdateien
|
+---03_main                               # Hauptmodul der Anwendung
|
+---03[a-y]_...                           # einzelne Packages der Anwendung
|
+---02z_code_snippets                     # Hilfscode – wird bei Abschluss entfernt
|   +---00_own_snippets                    # Eigene Hilfscodedateien
|   \---[...other snippets]                # Hilfscode anderer Quellen
|
+---04a_model_standby                     / 04[a-z] Ordner der ML-Modelle
|   +---00_sample_data                     # Beispieldaten zur Dokumentation
|   |   +---01_raw                         # Rohdatenauszug
|   |   +---02_processed                   # Auszug aufbereiteter Daten
|   |   \---03_for_modelling               # Auszug der Trainingsdaten
|   |
|   +---01_dataanalysis                    # Code explorativer Datenanalyse
|   +---02_modeling                        # Code der Modellerstellung
|   \---03_deployment                      # Code des Modeldeployments
|       \---model_version_final.R          # Namenskonvention aktueller Modell-Version
|
+---03_tests                              # Unittests
|
\---04_output                             # Generierte Ergebnisse
    +---01_documentation                   # Softwaredokumentation
    +---02_data                             # Datenausgabe
    +---03_figures                          # Grafiken
    +---04_logs                             # Logausgaben
    \---05_apps                             # User Interfaces

```

Quelle: Eigene Darstellung.

### **3 Zusammenfassung**

## 4 Literaturverzeichnis

**Angermeier, D./Bartelt, C./Bauer, O./Beneken, G./ Wittmann, M. (2024):** *V-Modell-XT Das deutsche Referenzmodell für Systementwicklungsprojekte Version 2.4.*

**Flyvbjerg, B./ Gardner, D. (2023):** *How Big Things Get Done: The Surprising Factors That Determine the Fate of Every Project, from Home Renovations to Space Exploration and Everything in Between.* Currency, New York.

**Greer, M. (2011):** *The PM Minimalist Quick Start Guide: The Absolute Least You Can Do!*

**IU Internationale Hochschule (2024):** *Management von IT-Projekten.*

**microTool GmbH (2024):** *objectiF RPM Benutzerhandbuch.*

**Pak, M. (2024):** *Aufgabenstellung\_DLMDWME01.*

**Wikipedia (2022):** *Projektstrukturplan.*

**5 Abbildungsverzeichnis**

1     Struktur des Ordneraufbaus . . . . . 5

## 6 Tabellenverzeichnis