Studienarbeit – Klassifikation von Bürgeranfragen

# 1. Einleitung

* [Platzhalter] Motivation und Relevanz
* [Platzhalter] Zielsetzung der Arbeit
* [Platzhalter] Aufbau des Dokuments

# 2. Theoretischer Hintergrund

## 2.1 Grundlagen von Natural Language Processing (NLP)

Natural Language Processing (NLP) ist ein Teilgebiet der künstlichen Intelligenz, dass sich mit der automatisierten Verarbeitung, Analyse und Generierung natürlicher Sprache beschäftigt. Ziel ist es, menschliche Sprache so zu modellieren, dass sie von Computern interpretiert und verarbeitet werden kann. Dabei kommen sowohl linguistische Regeln als auch statistische und maschinelle Lernverfahren zum Einsatz.

NLP findet Anwendung in zahlreichen Bereichen, darunter maschinelle Übersetzung, Textklassifikation, Chatbots, Informationsextraktion und Stimmungsanalyse. Besonders in der öffentlichen Verwaltung kann NLP genutzt werden, um eingehende Texte – etwa Bürgeranfragen – automatisiert zu analysieren und effizient zu bearbeiten.

Eine zentrale Herausforderung im NLP ist die Ambiguität natürlicher Sprache: Wörter können mehrere Bedeutungen haben, Sätze können unterschiedlich interpretiert werden. Die Vorverarbeitung von Textdaten (z. B. Tokenisierung, Normalisierung, Entfernen von Stoppwörtern) ist daher ein essenzieller Schritt, um strukturierte Eingabedaten für statistische Modelle bereitzustellen. [B3]

## 2.2 Textklassifikation im Überblick

Die Textklassifikation ist eine Kernanwendung innerhalb des NLP. Ziel ist es, einem gegebenen Text automatisch eine oder mehrere vordefinierte Klassen zuzuordnen. Typische Anwendungsfälle sind Spam-Erkennung, Kategorisierung von Support-Tickets oder Meinungsanalyse. [B4]

Klassische Textklassifikation basiert auf der Umwandlung von Texten in numerische Vektoren (Merkmalsräume), welche anschließend als Eingabe für ein maschinelles Lernverfahren dienen. Die Transformation erfolgt meist über sogenannte Bag-of-Words- oder TF-IDF-Verfahren, welche die Relevanz einzelner Begriffe innerhalb eines Korpus berechnen. [B2]

Danach kommen überwachtes Lernverfahren wie Naive Bayes, Logistic Regression oder Support Vector Machines (SVM) zum Einsatz. Diese Modelle lernen auf Basis gelabelter Trainingsdaten, wie bestimmte Textmuster mit bestimmten Klassen korrelieren. [B1]

Für die Klassifikation von Bürgeranfragen bedeutet das: Anhand eines Freitextes soll automatisiert erkannt werden, ob es sich etwa um ein Anliegen zum Thema „Abfall“, „Verkehr“ oder „Dokumente“ handelt. Damit stellt die Textklassifikation einen zentralen Bestandteil des Projekts dar.

# 3. Methodisches Vorgehen (CRISP-DM)

* [Platzhalter] 3.1 Business Understanding
* [Platzhalter] 3.2 Data Understanding
* [Platzhalter] 3.3 Data Preparation
* [Platzhalter] 3.4 Modelling
* [Platzhalter] 3.5 Evaluationskonzept

# 4. Umsetzung

* [Platzhalter] 4.1 Datenquelle und Struktur
* [Platzhalter] 4.2 Textverarbeitung (Tokenisierung, Cleaning, TF-IDF)

## 4.1 Datenquelle und Struktur

Da keine öffentlich zugängliche Sammlung von Bürgeranfragen in deutscher Sprache verfügbar war, wurde im Rahmen dieser Arbeit ein eigener Datensatz simuliert. Die Inhalte wurden mithilfe von ChatGPT generiert und anschließend manuell überprüft und überarbeitet, um einen realitätsnahen Eindruck typischer Anliegen aus dem kommunalen Kontext zu gewährleisten.

Der Datensatz umfasst 60 synthetische Einträge. Jeder Eintrag besteht aus einem kurzen freien Text (z. B. „Meine Mülltonne wurde nicht geleert.“) und einer zugewiesenen Kategorie. Die folgenden sechs Kategorien wurden verwendet:

* Abfall
* Dokumente
* Infrastruktur
* Verkehr
* Bau
* Sonstiges

Die Anfragen sind in einer CSV-Datei gespeichert und liegen in strukturierter Form im Projektordner unter data/raw/buergeranfragen\_sample.csv. Der Text liegt in der Spalte „anfrage“, die zugehörige Zielklasse in der Spalte „kategorie“. Die Daten wurden später für das Modelltraining in ein TF-IDF-Format überführt (siehe Abschnitt 4.2).

## 4.2 Textverarbeitung (Tokenisierung, Cleaning, TF-IDF)

Für die maschinelle Verarbeitung von Texten ist eine Vorverarbeitung essenziell, da Rohdaten häufig unstrukturiert, uneinheitlich und nicht direkt numerisch verarbeitbar sind. In dieser Arbeit, wurden daher mehrere Schritte zur Bereinigung und Transformation der Bürgeranfragen durchgeführt.

Zunächst wurden alle Texte in Kleinschreibung überführt, um Groß-/Kleinschreibung bei der Analyse zu ignorieren. Anschließend wurden Satzzeichen entfernt, um die Wortrepräsentationen zu vereinheitlichen und unnötige Tokens zu vermeiden.

Die anschließende Tokenisierung, also das Zerlegen des Textes in Einzelwörter („Tokens“), wurde zunächst mithilfe der Python-Bibliothek NLTK geplant. Aufgrund von technischen Schwierigkeiten beim Laden der Tokenizer-Modelle wurde stattdessen ein einfacher Fallback-Tokenizer mittels Python-Funktion split() verwendet. Für die gewählte Modellarchitektur (klassische ML-Modelle wie Naive Bayes) ist diese Vereinfachung zunächst ausreichend. Eine weiterführende Optimierung, etwa über spaCy oder Lemmatisierung, wird im weiteren Verlauf (siehe 5.3).

Die vorbereiteten Texte wurden im nächsten Schritt mithilfe des TF-IDF-Verfahrens (Term Frequency - Inverse Document Frequency) in numerische Merkmalsvektoren überführt. Dabei wird jedem Wort eine Gewichtung zugewiesen, die sowohl die Häufigkeit innerhalb eines Dokuments als auch die Seltenheit im Gesamtkorpus berücksichtigt. Dies stellt eine Weiterentwicklung gegenüber einfachen Worthäufigkeiten dar, wie sie in der Bag-of-Words-Repräsentation verwendet werden.

Die resultierende Merkmalsmatrix diente anschließend als Grundlage für das Training des Klassifikationsmodells (siehe Abschnitt 4.3).

## 4.3 Erstes Modelltraining (Naive Bayes)

Zur Kategorisierung der Bürgeranfragen wurde ein Multinomial-Naive-Bayes-Modell trainiert. Die Vektorisierung erfolgte mit TF-IDF, die Trainings- und Testdaten wurden im Verhältnis 80 % zu 20 % aufgeteilt. Bei insgesamt 60 Beispielen umfasste der Testdatensatz nur 12 Anfragen.  
  
Das Modell erreichte eine Genauigkeit von 8,3 % - das liegt deutlich unter dem Erwartungswert eines zufälligen Klassifikators. Der Klassifikationsbericht (siehe Abbildung 1) zeigt, dass nur die Kategorie „Abfall“ teilweise erkannt wurde, während andere Klassen vollständig verfehlt wurden.  
  
Die geringe Anzahl an Trainingsdaten sowie die ungleichmäßige Verteilung der Klassen im Testset sind mögliche Ursachen. In einer erweiterten Version des Projekts könnte eine gezielte Balancierung der Klassen sowie die Verwendung eines komplexeren Modells (z. B. Logistic Regression oder SVM) sinnvoll sein.

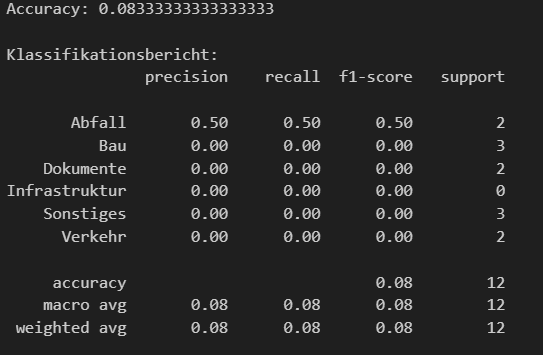
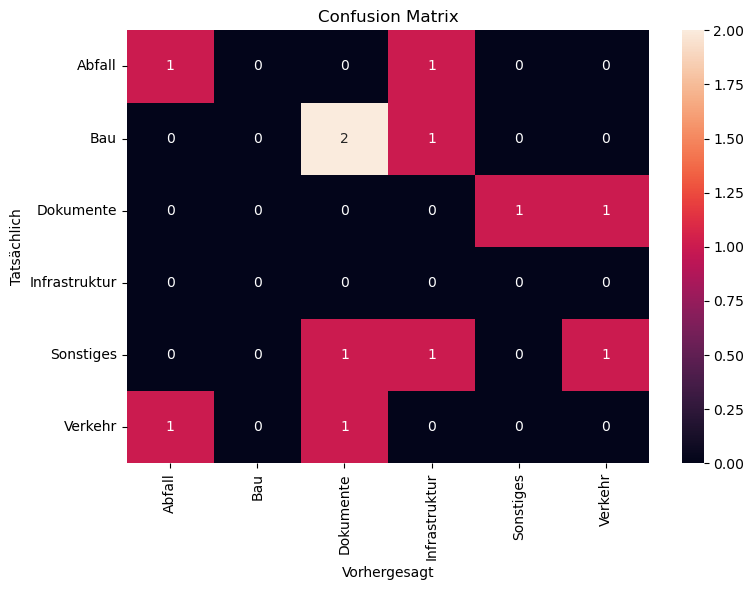


Abbildung 2: Confusion Matrix

Abbildung : Klassifikationsbericht

## 4.4 Modellverbesserung und Vergleich

### 4.4.1 Erweiterung der Textverarbeitung

zusätzliche Vorverarbeitung beschreiben (z. B. Stoppwörter, spaCy, Balancing)

### 4.4.2 Modellwechsel auf Logistic Regression

### 4.4.3 Vergleich der Ergebnisse

einen Vergleich machen (z. B. neue Confusion Matrix)

# 5. Fazit und Ausblick

* [Platzhalter] Zusammenfassung der Ergebnisse
* [Platzhalter] Bewertung & Grenzen
* [Platzhalter] Ausblick (z. B. LLM oder spaCy)

Literaturverzeichnis

[B1] IT-P GmbH, *Textklassifikation - Definition & Anwendungsbereiche*, 2023. https://www.it-p.de/lexikon/textklassifikation/ (accessed May 12, 2025).

[B2] Mwiti, D., “Python Bag of Words Model: Ein vollständiger Leitfaden,” *DataCamp*, 2024. https://www.datacamp.com/de/tutorial/python-bag-of-words-model (accessed May 12, 2025).

[B3] *Was ist NLP (Natural Language Processing)? | IBM*, 2025. https://www.ibm.com/de-de/think/topics/natural-language-processing (accessed May 12, 2025).

[B4] Wuttke, L., “Textklassifikation,” *datasolut GmbH*, 2023. https://datasolut.com/textklassifikation/ (accessed May 12, 2025).