



Fakultät für Informatik

Studiengang Informatik

Erstellung einer standardisierten
Material-Kostengliederung für Projekte einer
Bausoftware mittels Natural Language Processing

Bachelor Thesis

von

Florian Weidner

Datum der Abgabe: tt.mm.jjjj

Erstprüfer: Prof. Dr. Marcel Tilly

Zweitprüfer: Prof. Dr. Johannes Jurgovsky

EIGENSTÄNDIGKEITSERKLÄRUNG / DECLARATION OF ORIGINALITY

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Rosenheim, den tt.mm.jjjj

Vor- und Zuname

Abstract

Industry Foundation Classes (IFC) ist ein öffentlicher Standard im Bauwesen zur digitalen Beschreibung von Gebäudemodellen. In diesen Modellen können auch die Materialien der einzelnen Bauteile spezifiziert werden. Ziel ist es, eine Kostengliederungsstruktur in der Bausoftware ORCA AVA aus den Materialinformationen einer IFC-Datei zu generieren. Materialien können an verschiedenen Stellen im Modell angegeben werden. Ein Algorithmus soll diese Möglichkeiten in ein Interface zusammenführen. Außerdem handelt es sich bei der Materialangabe um ein Freitextfeld. Hier soll mithilfe von Natural Language Processing (NLP) und Künstlicher Intelligenz eine Lösung entwickelt werden, um eine standardisierte Liste der Materialien zu erschaffen. **Todo: Am Ende neu schreiben**

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ausgangssituation	1
1.2	Motivation	2
1.3	Ziel der Arbeit	3
1.4	Wissenschaftliche Vorgehensweise	5
2	Grundlagen	7
2.1	Projektmanagement	7
2.1.1	Vorgehensmodell	7
2.1.2	DevOps	8
2.2	IFC	8
2.2.1	Geschichte	8
2.2.2	Format	9
2.2.3	Verwendung	9
2.2.4	Möglichkeiten für die Materialangabe eines Bauteils	10
2.3	Das Format Kostengliederung in der ORCA AVA	11
3	Problemstellung und Anforderungen	13
3.1	Problemstellung	13
3.2	Anforderungen	14
3.2.1	Funktionale Anforderungen	14
3.2.2	Weitere Anforderungen	16
4	Theoretische Konzeption	17
4.1	Grundlegende Architektur und Vorgehensweise	17
4.1.1	Implementierung als Web-Service	18
4.1.2	Konzept der Materialstrukturierung	19
4.1.3	Konzept für das Auslesen von Materialien aus einer IFC Datei	19
4.2	Algorithmen für die Textklassifizierung	19
4.2.1	Ablauf einer Textklassifizierung	19

4.2.2	Sdca Maximum Entropy	21
4.2.3	Convolutional Neural Network	21
4.3	Möglichkeiten für die Feinstrukturierung	21
4.3.1	OpenAI Schnittstelle	21
4.3.2	FastText mit Density Based Clustering	21
5	Gegenüberstellung der möglichen Konzepte	23
5.1	Vergleich der Textklassifizierung	23
5.1.1	Messkriterien	23
5.1.2	Auswahl eines Algorithmus	23
5.2	Vergleich der Feinstrukturierung	23
5.2.1	Messkriterien	23
5.2.2	Auswahl eines Algorithmus	23
6	Praktische Umsetzung	25
6.1	Implementierung des Services	25
6.2	Python-Interop	25
6.3	Implementieren des Auslesens aller Materialien aus einer IFC Datei . .	25
6.4	Strukturieren der Materialnamen	25
6.4.1	Implementierung des Preprocessings	25
6.4.2	Erstellen einer Datengrundlage	25
6.4.3	Training des Text-Klassifizierungsmodells	25
6.4.4	Training des FastText Modells	25
6.4.5	DBSCAN Clustering	25
7	Maßnahmen zur Qualitätssicherung	27
7.1	Clean Code	27
7.2	Technische Hilfsmittel	27
7.3	Tests und Abnahme	27
8	Abschluss	29
8.1	Bewertung der praktischen Umsetzung	29
8.2	Fazit	29
8.3	Integration in die ORCA AVA	29
8.4	Ausblick	29
A	Anhang	31

B	Abkürzungsverzeichnis	37
C	Definitionsverzeichnis	39
	Literaturverzeichnis	41

Abbildungsverzeichnis

1.1	Relevanz Building Information Modeling (BIM)	3
1.2	Google Trends	4
1.3	Kano Modells	5
2.1	IfcMaterial	11
4.1	Anwendungsfalldiagramm	18
4.2	Dokumentation	19
4.3	Textklassifizierung	20
A.1	IFC Manager	31
A.2	IfcMaterialLayerSet	31
A.3	IfcMaterialProfileSet	32
A.4	IfcMaterialConsituentSet	32
A.5	CostStructure	32
A.6	Verteilungsdiagramm	33
A.7	Import	34
A.8	Takeover	34
A.9	Rating	35
A.10	MaterialCategories	36

1 Einleitung

Inhalt dieser wissenschaftlichen Arbeit ist das automatische Erstellen einer Material-Kostengliederung aus einer IFC-Datei mithilfe von NLP zur Erweiterung einer Bausoftware. Diese Bausoftware ist die ORCA AVA aus dem mittelständigen Softwarehaus „ORCA Software GmbH“ aus Neubuern. In diesem Kapitel soll eine kurze Einführung über die „ORCA Software GmbH“ und das Produkt ORCA AVA gegeben werden. Außerdem wird die Motivation für die Programmerweiterung und die wissenschaftliche Vorgehensweise dieser Arbeit beschrieben.

1.1 Ausgangssituation

Die im Titel beschriebene Bausoftware ist die ORCA AVA aus dem Softwarehaus „ORCA Software GmbH“. Dieses wurde im Jahr 1990 von Dipl.-Ing. Siegfried Tille und Dipl.-Ing. Heinz Nießen gegründet. Der Hauptsitz des Unternehmens ist in Neubuern, bei Rosenheim. Das Unternehmen ist auf die Produktentwicklung von Software für die Baubranche spezialisiert. Im Vordergrund stehen die Ausschreibungssoftware ORCA AVA und die Ausschreibungstext-Plattform AUSSCHREIBEN.DE. Ziel der Entwicklung ist es die Ausschreibung, Vergabe und Abrechnung (AVA) eines Bauvorhabens für Planer, Architekten und Bauingenieure zu vereinfachen. Der Leitfaden ist, Software zu entwickeln, die jeder versteht, intuitiv bedienbar ist, einen optimalen Workflow gewährleistet und viele Import- und Exportmöglichkeiten für den Datenaustausch bietet. Diese Arbeit fokussiert sich auf eine Erweiterung der ORCA AVA. Sie ist für alle Architektur- und Ingenieurbüros, Wohnungsbaugesellschaften, Unternehmen und Behörden zur einfachen Abwicklung von Bauprojekten mit Ausschreibung, Vergabe und Abrechnung. Zusätzlich bildet sie das Kostenmanagement von solchen Projekten ab. Die Software ist außerdem BIM fähig und bietet Deutsches Institut für Normung e. V. (DIN) zertifizierte Schnittstellen für den Datenaustausch an. Neben der ORCA AVA gibt es den IFC-Manager als eigene Instanz, der IFC-Modelle anzeigen kann. Die ORCA AVA kann dann Mengen aus dem Gebäudemodell übernehmen. Es stehen drei verschiedene Editionen der Software zur Verfügung. Die ORCA AVA Starter Edition (SE), die ORCA AVA Professional

Edition (PE) und die ORCA AVA Enterprise Edition (EE). Die aktuellste Version ist die 25.0.

Technisch wird die ORCA AVA in .NET entwickelt. Ein Großteil der Anwendung besteht noch aus Visual Basic (VB) Code. Alle neuen Komponenten und Erweiterungen werden in C# implementiert. Neue Graphical User Interface (GUI)-Komponenten werden dementsprechend mit Windows Presentation Foundation (WPF) entwickelt. WPF ist ein .NET Framework für das Erstellen von Windows Applikationen mit grafischer Benutzeroberfläche von Microsoft. (Microsoft, 2022b) Die ORCA AVA und der IFC Manager (siehe Abschnitt 2.2.3) laufen in eigenen Prozessen und kommunizieren auf Prozessebene in der lokalen Umgebung.

1.2 Motivation

Ziel der „ORCA Software GmbH“ ist es, BIM noch mehr in die ORCA AVA zu integrieren. BIM bedeutet, dass die Planung von Bauprojekten vollständig auf digitaler Basis durchgeführt wird. Für jeden Projektbeteiligten besteht somit jederzeit Zugriff auf alle projektrelevanten Daten über Kosten, Mengen und Zeitabläufen. Somit können Baukosten einfacher ermittelt und der Bauprozess besser überwacht werden. In Abbildung 1.1 ist zu sehen, dass BIM ein relevanter Begriff in der Baubranche ist. Die Verwendung der Praxis ist allerdings noch nicht verbreitet. (Thomas, Baumanns and Dr. Philipp-Stephan, Freber and Dr. Kai-Stefan, Schober and Dr. Florian, Kirchner, 2016, p. 20) Mit dem „IFC First“ Ansatz ist das langfristige Ziel der ORCA AVA das Thema BIM noch mehr abzudecken. Aus den Daten eines 3D-Gebäudemodells soll automatisch ein Ausschreibungstext erstellt werden können. Bauteile aus dem Gebäudemodell sollen dann mit Kurztext, Langtext, Menge, Preis und vordefinierten Kostengliederungen in den Programmteil Bauelemente der ORCA AVA übernommen werden. Aufgrund der hohen Relevanz des Themas, spricht das Ganze viele Kunden an und stellt somit ein effektives Werbemittel für den Verkauf der Software dar. Ein Teil des langfristigen Ziels ist die Übernahme der Baumaterialien aus einer IFC-Datei. Diese bietet die erste Kostengliederung für den „IFC First“ Ansatz. Die Übernahme von weiteren Daten aus dem IFC-Modell können darauf aufgebaut werden.

Ein weiterer Trendbegriff, der mit der Programmerweiterung abgedeckt werden soll, ist „Künstliche Intelligenz“ und „Maschinelles Lernen“. Das Suchinteresse der beiden Themen ist in Abbildung 1.2 zu sehen. Die Werte geben das Google-Suchinteresse relativ zum höchsten Punkt im angegebenen Zeitraum an. Der Begriff „Maschinelles Lernen“ hat seit Anfang 2015 ein steigendes Suchinteresse. Bei „Künstliche Intelligenz“ ist das

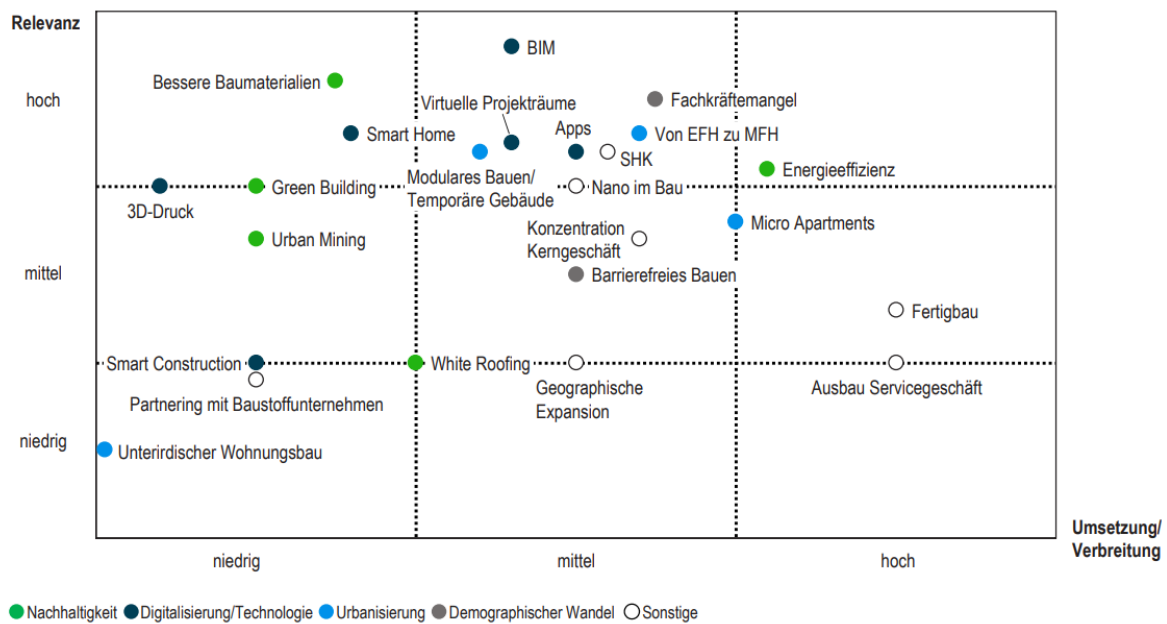


Abbildung 1.1: Megatrends Nachhaltigkeit und Digitalisierung in der Bauwirtschaft. Bilder aus (Thomas, Baumanns and Dr. Philipp-Stephan, Freber and Dr. Kai-Stefan, Schober and Dr. Florian, Kirchner, 2016, p. 20)

Suchinteresse von Anfang 2017 bis Ende 2021 konstant hoch. Der starke Anstieg ab November 2022 ist signifikant und korreliert mit der Veröffentlichung der Software ChatGPT. Man erkennt insgesamt, dass das Interesse über die letzten Jahre stetig ansteigt. Die Abdeckung dieser Begriffe ist seit geraumer Zeit ein Wunsch der Vertriebsseite für den noch besseren Verkauf der Software. Die Erweiterung wäre der erste Einsatz von maschinellem Lernen. Es ist also zusätzlich auch ein Pilotprojekt in der ORCA AVA Entwicklung, um sich mit Machine-Learning-Algorithmen vertraut zu machen und Erfahrungen in diesem Themengebiet zu bekommen.

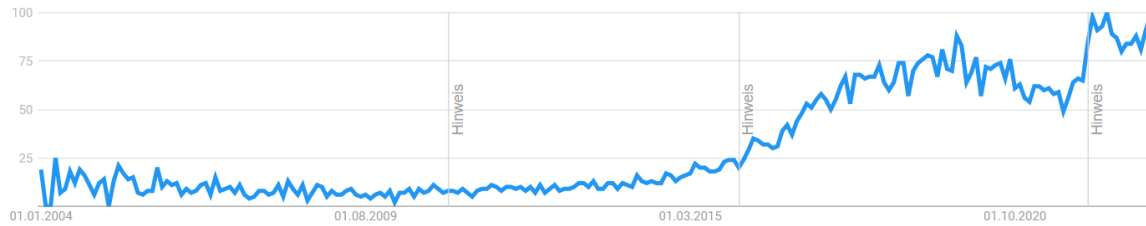
Durch die genannten Punkte ist die Programmerweiterung dem Kano-Modell nach als Begeisterungsfeature zuzuordnen. Die Kundenzufriedenheit steigt also exponentiell mit dem Erfüllungsgrad der Anforderung. (siehe Abbildung 1.3) Mit der Zeit wandelt sich es dann erst in ein Leistungsfeature und irgendwann in ein Basisfeature. (Hölzing, 2008, p. 3-4) Es besteht also die Motivation die Erweiterung möglichst exklusiv zur Verfügung zu stellen. Es eignet sich somit sehr für die ORCA AVA EE Edition.

1.3 Ziel der Arbeit

Um die Anforderungen aus Abschnitt 3.2 umsetzen zu können, muss zuvor ein Ziel formuliert werden. Die Zieldefinition für diese Arbeit wird nach der SMART Technik



(a) Suchinteresse: Künstliche Intelligenz



(b) Suchinteresse: Machine Learning

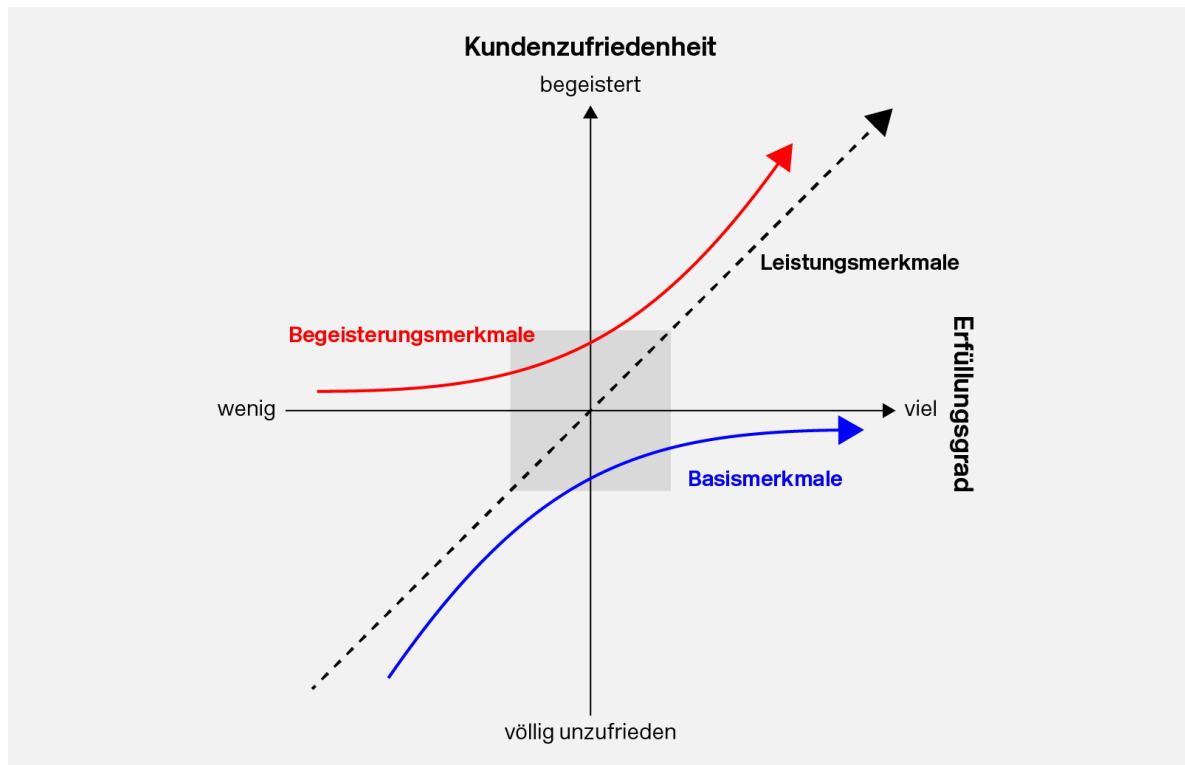
Abbildung 1.2: Google Suchinteresse der beiden Begriffe „Künstliche Intelligenz“ und „Maschinelles Lernen“ seit 2004 bis 2023

definiert. Dabei muss ein Ziel folgende Eigenschaften haben:

- Spezifisch
- Messbar
- Attraktiv
- Realisierbar
- Terminiert

Anhand dieser Kriterien wurde folgendes Ziel definiert:

Ziel. Bis zum 15.06.2023 wird eine „Draftprojekt“ mit allen vollständigen funktionalen Anforderungen für die ORCA AVA implementiert. Für die Implementierung werden passende Algorithmen nach den Kriterien der Messbarkeit (siehe Abschnitt 3.2.1) ausgewählt. Zusätzlich sollen die Maßnahmen zur Qualitätssicherung (Kapitel 7) durchgeführt werden und die Qualitätssicherungs-Abteilung alle Funktionen freigegeben haben.



Quelle: <https://dmexco-lightsails-media.s3.eu-central-1.amazonaws.com/wp-content/uploads/2021/01/04112832/Kano-Modell.png> (Accessed: 2023-2-24)

Abbildung 1.3: Kano Modell der Kundenzufriedenheit

1.4 Wissenschaftliche Vorgehensweise

Im Folgenden wird die wissenschaftliche Vorgehensweise der Arbeit aufgezeigt. In Abschnitt 1 wurde bereits die Ausgangssituation beschrieben, das Entwickeln der Programmiererweiterung motiviert und das Ziel dieser Arbeit definiert. Im nächsten Kapitel, Kapitel 2, werden Grundlagen erläutert, auf die diese Arbeit aufbaut. Im anschließenden Kapitel 3 sollen die Problemstellung und Anforderungen analysiert werden. Danach (Kapitel ??) wird eine Lösung für die Problemstellung theoretisch konzipiert und verschiedene mögliche Algorithmen analysiert. In Kapitel 5 werden die Algorithmen gegenübergestellt und nach den definierten Kriterien ausgewählt. Kapitel 7 zeigt die genutzten Maßnahmen der Qualitätssicherung auf. Am Ende wird das Ergebnis bewertet und ein Ausblick gegeben. Todo: Muss noch vervollständigt werden

2 Grundlagen

In diesem Kapitel werden die Grundlagen im Bereich der fachspezifischen Themen, Technik und Projektmanagement vermittelt. Das soll Verständnis für die folgenden Kapitel schaffen. Zuerst geht es um das Projektmanagement. Anschließend geht es um das Format IFC, dessen Geschichte und dessen Nutzungsmöglichkeiten für diese Arbeit. Außerdem wird die Struktur und der Nutzen einer Kostengliederung in der ORCA AVA veranschaulicht.

2.1 Projektmanagement

Bevor auf technische und fachliche Aspekte von IFC und Kostengliederungen eingegangen wird, folgt die Einführung in das Projektmanagements mit SCRUM und Development Operations (DevOps).

2.1.1 Vorgehensmodell

Das Projekt wurde mithilfe agiler Softwareentwicklung durchgeführt. Im ORCA AVA Entwicklungsteam wird das SCRUM Modell verwendet. Die Entwicklung der Produkterweiterung in dieser Arbeit läuft in diesem SCRUM-Prozess.

Definition 1 (Scrum). „*Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.* “ (Ken and Jeff, 2020, p. 3)

SCRUM verwendet einen iterativen, inkrementellen Ansatz. So kann Sprint für Sprint aus gewonnenen Erfahrungen der Entwicklungsablauf optimiert werden. (Ken and Jeff, 2020) Product Owner in der ORCA AVA Entwicklung ist die „ORCA Software GmbH“, in Vertretung des Produktmanagement-Teams. Die Umsetzung der Sprint-Ziele wird von einer Person implementiert.

2.1.2 DevOps

Die Bezeichnung DevOps vereint die beiden Praktiken „Development“ (Entwicklung) und „Operations“ (Vorgänge). Die traditionelle Trennung von Entwicklung und Softwarebetrieb führt oft zu Interessenskonflikten. Entwickler wollen stetig die Software verbessern und der Betrieb will Änderungen vermeiden, um die Stabilität des Systems zu gewährleisten. Durch DevOps entsteht ein Softwareentwicklungsprozess, den man durch Praktiken wie Continuous Integration, Continuous Delivery, Continuous Deployment, automatisiertes Testen, Infrastructure-as-Code und automatische Veröffentlichungen beschleunigt. DevOps steht auch für eine Entwicklungskultur mit offener Zusammenarbeit, Kommunikation, Transparenz und Eingestehen von Fehlern, um Konflikte im Team zu vermeiden. (Beetz and Harrer, 2021) Im Entwicklungsteam der ORCA AVA wird diese Praktik umgesetzt. Die technischen Hilfsmittel, die für den DevOps Prozess verwendet werden, sind in Abschnitt 7.2 beschrieben.

2.2 IFC

Die Daten für die Material-Kostengliederung werden aus einem digitalen Gebäudemodell entnommen. Der öffentliche internationale Standard (ISO 16739-1:2018) für Gebäudemodelle ist IFC. (*Industry Foundation Classes*, 2017) Dieser wird auch in der bestehenden ORCA AVA benutzt, um den Ausschreibungsprozess zu unterstützen. IFC Dateien können geöffnet, angeschaut und Informationen über das Modell in die Hauptsoftware übernommen werden. Im folgenden Kapitel wird die Geschichte, das Format von IFC und die Verwendung in der ORCA AVA erläutert.

2.2.1 Geschichte

IFC ist die Hauptleistung der buildingSMART International, Ltd. Die non-profit Organisation will mit der Spezifikation den BIM Prozess fördern und voranbringen. (Grutters, 2022) Angefangen hat die Organisation als der Verein *Industriellianz für Interoperabilität IAI e. V.* mit Sitz in Berlin. 1994 startete die Entwicklung an dem offenen Datenmodellstandard IFC. Dieser sollte die Anforderungen der Industrie an Interoperabilität gerecht werden und eine gemeinsame Basis zum Austausch von Informationen durch verschiedenen Anwendungen schaffen. Im Zusammenhang mit BIM sollten Daten lesbar, editierbar für verschiedene Systeme durch den Bauprozess und kompletten Lebenszyklus eines Gebäudes geteilt werden. (Laakso and Kiviniemi, 2012) Nach einigen Prototypen wurde 1999 IFC 2.0 veröffentlicht. Diese wurde bis 2007 mit der Version 2.3.0.1 stetig

verbessert. Die Version 2.3 wird auch in heutigen Projekten noch verwendet. 2013 wurde IFC4 veröffentlicht, welche mit der Version 4.0.2.1 die aktuellste, offizielle IFC Version ist. Das Format wird weiterhin stetig weiterentwickelt. Neue Versionen stehen schon vor einer Abstimmung der International Organization for Standardization (ISO). (van Berlo, 2022) Es gibt folgendermaßen aktuell zwei offizielle Versionen. Beide werden in der Praxis benutzt und sind, wie in Abschnitt 2.2.3 beschrieben, mit der ORCA AVA kompatibel.

2.2.2 Format

Das IFC Format kodiert folgende Daten:

- Identität, Semantik, Attribute und Relationen von Objekten
- Abstrakte Konzepte wie Performance oder Kosten
- Prozesse wie z.B. Installationen und Operationen
- Personen wie z.B. Eigentümer oder Lieferanten

Die Spezifikation kann also für das Bauen, Betreiben oder Nutzen eines Gebäudes genutzt werden. IFC ist ein Implementierungs-Unabhängiges Datenmodell, welches in verschiedenen Umgebungen und elektronischen Formaten benutzt werden kann. Es kann beispielsweise in eine relationales Datenbankschema gegossen oder auch als Dateiformat implementiert werden. Das weitverbreitetste Format ist Standard for the exchange of product model data (STEP) Physical Format (SPF) (Laakso and Kiviniemi, 2012; Grutters, 2022). SPF ist das kompakteste Format für den dateibasierten Import und Export von IFC-Dateien. Die Dateiendung des Formates ist „.ifc“. Des Weiteren kann IFC als Extensible Markup Language (XML) oder ZIP Datei verwendet werden. (Grutters, 2022)

2.2.3 Verwendung

„Today, IFC is typically used to exchange information from one party to another for a specific business transaction.“ (Grutters, 2022)

In der ORCA AVA wird IFC für das Einlesen und Übernehmen von Maßen und Mengen in die Ausschreibung verwendet. Es vereinfacht den Prozess, die in der Computer-aided design (CAD)-Software erstellten Daten einfach in die Leistungsverzeichnisse der ORCA AVA zu überführen. Die ORCA AVA kann IFC-Dateien einlesen. Im IFC Manager

wird das 3D-Modell dann angezeigt. In dem geöffneten Fenster gibt es einige fachliche Ansichten, die jegliche IFC-Daten nochmal fachlich abstrahieren. In den Ansichten können zum Beispiel bestimmte Maße oder die Anzahl verschiedener Bauteile, die aus dem IFC-Modell berechnet werden, in die ORCA AVA übernommen werden. Auch alle im IFC definierten Eigenschaften eines Bauteils sind in der Eigenschaften-Ansicht sichtbar. Hier kann man auch die Materialbezeichnung des Bauteils finden, die im Modell hinterlegt ist. In Abbildung A.1 ist die Oberfläche des IFC Managers zu sehen. Man sieht die Materialangabe rechts unten im Eigenschaftsfeld unter dem 3D-Modell.

Für das Arbeiten mit IFC-Dateien wird die open-source Bibliothek *xbim-toolkit* verwendet. Die .NET Bibliothek kann IFC-Dateien lesen, schreiben und anzeigen. Außerdem unterstützt es bei der Berechnung von komplexer Geometrie, um die Modelldaten für Analysen nutzbar zu machen. Die Entwicklung der Bibliothek startete 2007 und läuft seit 2009 in Zusammenarbeit mit der Northumbria University. Mittlerweile bildet es die Standards IFC Version 2.3 (IFC2x3) und IFC Version 4 (IFC4) zu 100% ab. Zudem bietet es an, auch IFC2x3 Modelle über das IFC4 Interface anzuprogrammieren. Somit können mit einer Codebasis beide Formate abgebildet und unterstützt werden. (Xbim Ltd., n.d.)

2.2.4 Möglichkeiten für die Materialangabe eines Bauteils

In einem IFC-Modell können Materialien auf unterschiedliche Weise einem Bauteil zugewiesen werden. Die Spezifikation bietet die Klasse *IfcMaterial*. Diese bildet fachlich ein Material ab. Es hat die Attribute Name, Description und Category. (buildingSMART International Ltd., 2017a) Eine Instanz von *IfcMaterial* kann mit einem Element oder Elementtyp über die *IfcRelAssociatesMaterial* verbunden werden. Diese Zuweisung kann über verschiedene Weisen stattfinden. In Abbildung 2.1 ist eine direkte Zuweisung über das *IfcRefAssociatesMaterial* zu sehen. Hier hängt das *IfcMaterial* direkt am Element. Bei einer Platte mit mehreren Schichten kann das *IfcMaterial* auch wie in Abbildung A.2 zu sehen an einem *IfcMaterialLayerSet* zugewiesen sein. Jede Schicht hat hier seine eigene Dicke und eigene Materialangabe. Ein Träger kann das *IfcMaterial* über ein *IfcMaterialProfileSet* abgebildet haben. (siehe Abbildung A.3) Bei zum Beispiel einer Tür können verschiedene Materialien des Rahmens und der Verglasung über das *IfcMaterialConstituentSet* abgebildet werden. (siehe Abbildung A.4) (buildingSMART International Ltd., 2017b) Zusätzlich gibt es noch ältere Möglichkeiten Materialangaben an ein Element zu binden. In IFC2x3 gibt es die Klasse *IfcMaterialList* (Thomas et al., 2007). Diese Möglichkeit ist deprecated, in der Praxis kommt das IFC2x3 Format noch

regelmäßig vor. Die *IfcMaterialList* muss also auch beachtet werden.

Neben verschiedenen Verbindungen zu *IfcMaterial* hat jedes Element bei IFC auch verschiedene Property Sets. Property Sets sind Container, die Eigenschaften je nach ihrem Namensattribut enthalten. Einige Property Sets sind in der Spezifikation des IFC Standards enthalten. Es können auch zusätzlich benutzerdefinierte Property Sets erfasst werden. Diese können auch Informationen über das Material eines Elements enthalten. (buildingSMART International Ltd., 2017c) Das PropertySet *Pset_MaterialConcrete* weist darauf hin, dass es sich z.B um Beton handelt.

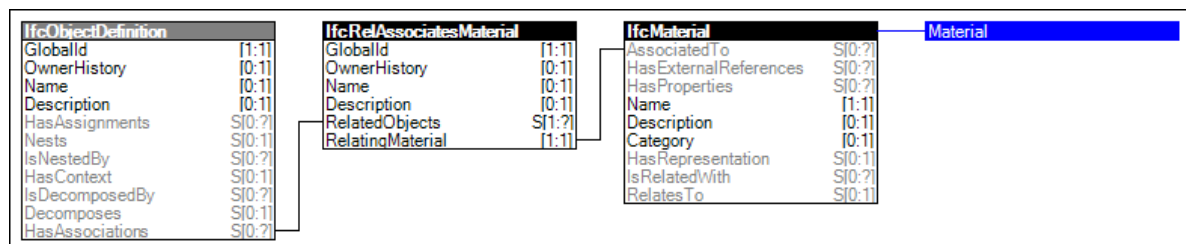


Abbildung 2.1: Material Single Association

2.3 Das Format Kostengliederung in der ORCA AVA

Die Kostengliederung bietet eine Struktur, um Gesamtkosten einer Baumaßnahme in Kostengruppen unterteilt auswerten zu können. Logisch zusammengehörende Kosten können so in eine Kostengruppe (KG) zusammengerechnet werden. Außerdem handelt es sich beim Aufbau um eine Baumstruktur, wodurch Kostengruppen hierarchisch addiert werden können. In Abbildung A.5 ist ein Beispiel einer Kostengliederung mit Materialien zu sehen. Der Gliederungspunkt *Mineralisch* wird in die Unterpunkte *Mauerwerk*, *Beton*, *Dachziegel* und *Kies* unterteilt. Die Kosten der Unterpunkte können also für den Gliederungspunkt *Mineralisch* addiert werden, um die Kosten aller Mineralien zu erfassen. Bei einem Bauprojekt kann man so Kosten über alle Projektphasen vergleichen. Von der Kostenschätzung über die Ausschreibung bis zur Rechnungsfreigabe. So können Kostenauswertungen nach den verschiedenen Kostengruppen durchgeführt werden. In einem ORCA AVA Projekt kann man verschiedene Kostengliederungen definieren. Es existieren bereits Standardkostengliederungen beim Erstellen eines neuen Projektes. Zusätzlich können neue Kostengliederungen erstellt oder importiert werden. (ORCA Software GmbH, n.d.)

Technisch ist eine Kostengliederung als Modell im C# Code definiert. Der Aufbau bildet die Baumstruktur über eine Referenz zur ParentNode und mehreren ChildrenNo-

des ab. Das Model kann über die ORCA AVA interne Middleware in der Datenbank persistiert werden. Die persistierten Kostengliederungen werden dann in entsprechenden Programmteilen angezeigt. So müssen nach der generierten Kostengliederungs-Struktur die strukturierten Materialbezeichnungen in das C# Modell abgebildet werden. So wird nach dem Import die erstellte Kostengliederung automatisch in der Oberfläche angezeigt und kann für Kostenauswertungen genutzt werden.

3 Problemstellung und Anforderungen

Im folgenden Kapitel werden die Problemstellung und Anforderungen aufgeführt, um das Projekt einzugrenzen. Diese kommen vom Auftraggeber, der „ORCA Software GmbH“ selbst.

3.1 Problemstellung

Die Problemstellung entsteht aus der Ausgangssituation (Abschnitt 1.1) und der Motivation (Abschnitt 1.2) der Bausoftware ORCA AVA. Der BIM Prozess soll noch mehr in die Software integriert werden. Deswegen wurde sich die Programmerweiterung der Materialkostengliederung gewünscht, um den ersten Schritt zu „BIM First“ (siehe Abschnitt 1.2) zu schaffen. Die Problemstellung dieser Arbeit bezieht sich der dementsprechenden Umsetzung dieser Programmerweiterung:

Problemstellung 1. *Wie lässt sich am besten eine Liste von Materialbezeichnungen in eine hierarchische Material-Kostengliederung strukturieren?*

Da es sich hier um eine offene Frage handelt, sind Spielräume für die Antwort offen. Es soll ein Konzept und eine Architektur für die Programmerweiterung entstehen. Um ein fachlich richtiges Ergebnis der Strukturierung zu erreichen, ist aber vor allem folgende Problemstellung wichtig:

Problemstellung 2. *Welcher NLP Algorithmus eignet sich am besten für das Klassifizieren und Strukturieren von Material-Daten?*

Bei dieser offenen Frage, soll am Ende ein Algorithmus aus einer Auswahl ausgewählt werden. Um verschiedene Ansätze und Algorithmen vergleichen zu können, benötigt man Kriterien der Messbarkeit. Diese werden in Abschnitt 3.2.1 definiert. So kann man neben der Berücksichtigung der Anforderungen den geeignetsten Algorithmus für das Problem auswählen.

3.2 Anforderungen

Alle definierten Anforderungen kommen vom Projektmanagement Team der ORCA Software GmbH und dem Teamleiter der ORCA AVA Entwicklung. Die Gesamtheit von Anforderungen kann in funktionale Anforderungen, Leistungsanforderungen, spezifische Qualitätsanforderungen und Randbedingungen unterteilt werden.(Glinz, 2007)

3.2.1 Funktionale Anforderungen

Definition 2 (Funktionale Anforderung). „*A functional requirement is a requirement that pertains a functional concern.*“(Glinz, 2007)

Wie in Definition 2 beschrieben, geht es bei einer funktionalen Anforderung um die Funktion einer Software. Die im Mittelpunkt stehende Funktion ist die erläuterte Erstellung der hierarchischen Materialstruktur (Anforderung 1). Weitere Funktionen sind die Anforderungen 2 und 5. Nach dem Importieren der Materialstruktur soll bei der Übernahme von Mengen die Materialkostengliederung automatisch zugewiesen werden. Zusätzlich sollen die Ergebnisse sich stetig verbessern. Dafür soll eine Möglichkeit geschaffen werden, dass Nutzer der ORCA AVA Verbesserungen eintragen und somit den Datenbestand erweitern und verbessern.

Anforderung 1. *Aus einer Liste von Materialbezeichnungen soll eine hierarchische Baumstruktur entstehen, welche in einer ORCA AVA Kostengliederung benutzt werden kann.*

Anforderung 2. *Bei der Übernahme einer Menge aus dem IFC Manager soll die Materialkostengliederung zugewiesen werden.*

Anforderung 3. *Bei schlecht generierter Struktur soll eine persistente, dauerhafte Verbesserung des Benutzers möglich sein.*

Die in Anforderung 1 generierte Materialkostengliederung soll fachlich möglichst richtig sein. Dazu werden folgende Kriterien der Messbarkeit definiert, um den genauesten Algorithmus zu finden:

Genauigkeit

Die Genauigkeit bei maschinellem Lernen misst die Wirksamkeit eines Modells. Klassifizierte Daten werden in einen Testdatensatz und einen Trainingsdatensatz geteilt. Mit dem Testdatensatz wird das trainierte Modell getestet und als Ergebnis der Genauigkeit

hat man eine prozentuale Angabe der Übereinstimmung des Testdatensatzes. (Microsoft, 2022a)

Interpretierbarkeit

Definition 3 (Interpretierbarkeit). „*Interpretability is the degree to which a human can understand the cause of a decision.*“ (Miller, 2017)

Hier geht es darum, die Ergebnisse des Algorithmus nachzuvollziehen zu können. Alleine die Genauigkeit reicht nicht dazu aus, einem Machine-Learning-Algorithmus vertrauen zu können. Ein Modell ist besser interpretierbar als ein anderes Modell, wenn seine Entscheidungen für einen Menschen leichter nachvollziehbar sind als die Entscheidungen des anderen Modells. Die Interpretierbarkeit kann man nicht direkt messen. Sie entsteht aus der Erfahrung über den Algorithmus und die Nachvollziehbarkeit der vorhergesagten Ergebnisse durch Testen. (Molnar, 2022) Das Endergebnis der Materialkostengliederungen aus verschiedenen Projekten, werden fachlich vom Produktmanagement bewertet. Diese haben das fachliche Wissen, um das Ergebnis bewerten zu können.

Robustheit

Von Deep-Learning-Algorithmen erwartet man, dass sie robust gegen kleine Störungen in der Eingabe sind. Es wurde allerdings schon festgestellt, dass kleine Störungen teilweise das Ergebnis ändern können. (Szegedy et al., 2013) Im Bezug mit auf die Materialeingabe sollen zum Beispiel verschiedene Satzzeichen keinen Einfluss auf das Ergebnis haben. Die Materialbezeichnung „Holz, Birke“ und „Holz - Birke“ sollen mit dem Algorithmus auf das gleiche Ergebnis führen.

Performance

Aspekte wie Geschwindigkeit und Ressourcenbedarf spielen eine sekundäre Rolle. Mit einem Fortschrittsbalken kann man dem Benutzer zeigen, dass es sich um einen zeitintensiven Algorithmus handelt. Wenn es den Anwendungsfluss allerdings sehr beeinträchtigt und verlangsamt, scheidet er als Option aus. Zum Punkt Performance zählt auch die Trainingsdauer des Algorithmus. Diese kann beim Vergleich von verschiedenen Algorithmen helfen.

Die funktionalen Anforderungen werden noch nicht direkt in die ORCA AVA integriert.

Ein Testprojekt soll zur Demonstration der Funktionen dienen. Es wird allerdings einen Vorschlag zur Integration in die Oberfläche der ORCA AVA geben. (siehe Kapitel 8.4)

3.2.2 Weitere Anforderungen

Neben den funktionalen Anforderungen gibt es vor allem technische Anforderungen an die Programmerweiterung. Diese wurden vom Teamleiter der ORCA AVA Entwicklung gestellt:

Anforderung 4. *Die Programmerweiterung soll in .NET (C#) oder in Python implementiert werden.*

Die technische Einschränkung der Anforderung 4 macht den Code und die Programmerweiterung für die ORCA Software GmbH wartbar. C# ist die Hauptsprache der Firma. Zusätzlich gibt es auch schon Module in einem anderen Produkt mit Python. Für beide Programmiersprachen gibt es also schon Wissen in der Entwicklung, um bei Fehlerbehebungen eine erneute Einarbeitungszeit zu vermeiden.

Anforderung 5. *Die Strukturierung soll mit der Erweiterung der Datengrundlage immer besser gemacht werden können.*

Durch das Einlesen neuer IFC-Dateien von Benutzern, werden immer neue Materialbezeichnungen gesammelt, die die Datengrundlage erweitern. Dadurch müssen immer wieder neue Begriffe klassifiziert werden können, um das Modell stetig zu verbessern. Dementsprechend müssen Künstliche Intelligenz (KI)-Modelle auch immer wieder neu trainierbar sein. Diese Funktionen sollen also einem ORCA Internen Mitarbeiter zur Verfügung stehen.

Durch die beschriebenen Anforderungen aus Abschnitt 3.2 entstand das beschriebene Ziel der Arbeit (siehe Abschnitt 1.3)

4 Theoretische Konzeption

In diesem Kapitel wird ein theoretisches Konzept für das Lösen der Problemstellung erstellt und begründet. Zuerst wird das Konzept vorgestellt. Dann werden mögliche Algorithmen erläutert und evaluiert, wie sie zu den Anforderungen passen. Im Folgekapitel werden die Algorithmen miteinander verglichen und der geeignetste ausgewählt.

4.1 Grundlegende Architektur und Vorgehensweise

In diesem Abschnitt wird das erarbeitete Konzept für die Erweiterung vorgestellt und begründet. In Abbildung 4.1 ist ein Anwendungsfalldiagramm der neuen Funktionen zu sehen. Die Anwendungsfälle des Benutzers der ORCA AVA decken sich mit den schon beschriebenen funktionalen Anforderungen (Abschnitt 3.2.1) Die zusätzlichen Anwendungsfälle ergeben sich aus Anforderung 5.

Das Ergebnis des Hauptanwendungsfalles der Material-Strukturierung ist abhängig von den Inhalten der IFC-Datei. Durch die Nutzung verschiedener CAD-Programme und eigene Vorlieben bei der Materialangabe der ORCA AVA Benutzer entstehen viele unterschiedliche Inhalte bei der Materialangabe. Um die hierarchische Strukturierung zu vereinfachen, werden Materialien zuerst nach dem „Teile-und-herrsche“ Verfahren in Oberkategorien klassifiziert. In Abbildung A.10 sind die Kategorien mit Beispielen aufgelistet. So müssen anschließend nur noch eine kleinere Liste von Materialien einer Oberkategorie weiter hierarchisch strukturiert werden.

In Abbildung A.6 ist ein Verteilungsdiagramm mit allen relevanten Objekten zu sehen. Auf dem Lokalem Computer des Benutzers laufen die beiden Prozesse der ORCA AVA und des IFC Managers. Die ORCA AVA kommuniziert mit einem Webserver über Representational State Transfer (REST). Dieser verwaltet die anhängende SQL-Datenbank und übernimmt das Strukturieren des Material-Kostengliederungs-Import.

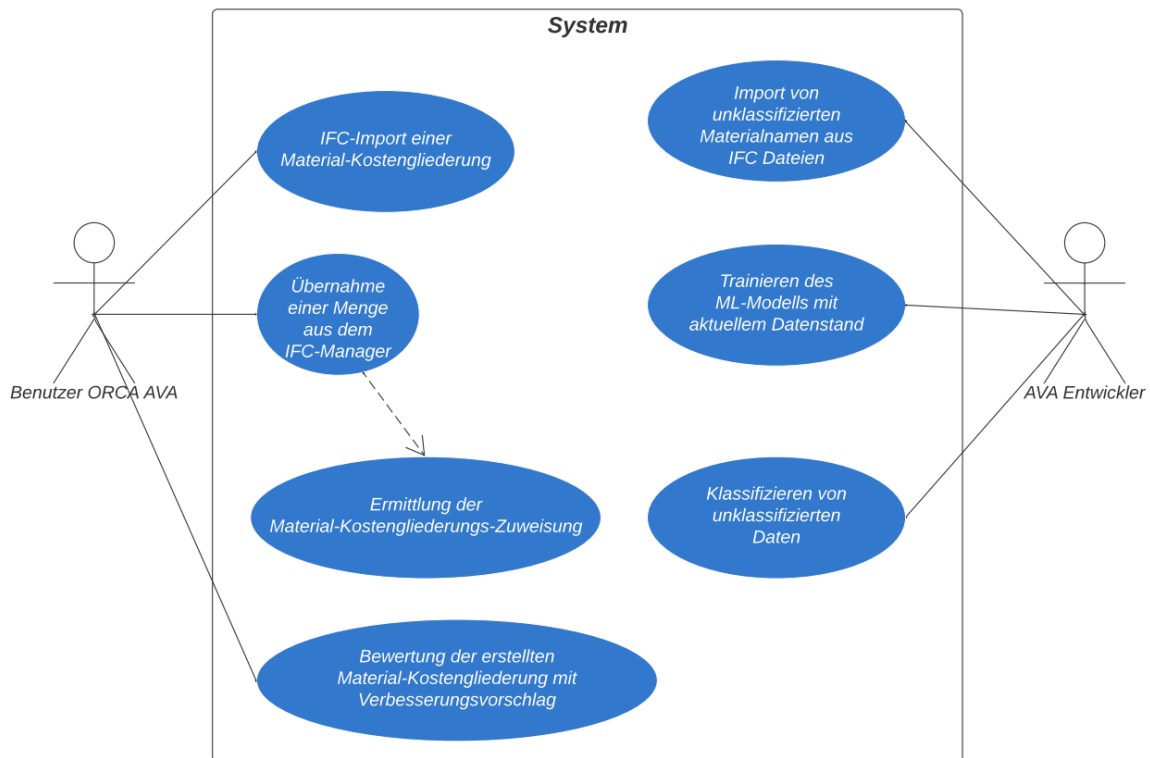


Abbildung 4.1: Anwendungsfalldiagramm

4.1.1 Implementierung als Web-Service

Für die Materialstrukturierung ist ein neuer Webservice vorgesehen, mit den die ORCA AVA kommunizieren kann. Dies bietet einige Vorteile:

- Der Import greift immer auf das aktuellste KI-Modell zu, welches zentral verwaltet werden kann
- Wegen der Verwendung von Python muss nicht bei jedem Client eine lauffähige Pythonumgebung vorhanden sein oder durchs Setup installiert werden.
- Neue Materialbezeichnungen und Klassifizierungen können direkt zentral Datenbank persistiert werden.
- Eine Änderung/Verbesserung der Datengrundlage kann durch automatisches Trainieren direkt zu einem verbessertem KI-Modell führen.

Für die ORCA AVA bestehen schon Services für die Bereitstellung von News, die Lizenzierung oder das Updaten der Software. Durch die Vorgabe der ORCA Software GmbH wird der Service mit ASP.NET implementiert. ASP.NET ist ein Open-Source

und Plattform-unabhängiges Web-Framework für die Entwicklung cloudbasierter Anwendungen wie Web-Applikationen, Internet of Things (IoT)-Apps oder mobile Backends. (.NET Foundation and Contributors, n.d.). Die Architektur dieses Services entspricht dementsprechend den schon existierenden Services.

Daten werden über den Service in einem SQL-Datenbank Server gespeichert. Hier werden auch die ORCA Software GmbH internen Standards verwendet. Daten für den Material-Kostengliederungs-Import sind in der Tabelle *Materials* persistiert. Die Tabellendokumentation ist in Abbildung 4.2 zu sehen. Zusätzlich bestehen Tabellen für die Autorisierung der Zugriffe auf den Service.

Name	Typ	Beschreibung	Besonderheit
Id	INT	Generierte Id	Primary Key
MaterialBezeichnung	NVARCHAR(100)	Materialname	
MaterialCategory	INT	Zugewiesene Material-Überkategorie (siehe enum: MaterialCategory)	
IfcFile	NVARCHAR(255)	interner Pfad zu IFC-Datei, die diese MaterialBezeichnung enthält	
BauteilGuid	NVARCHAR(100)	GUID des Bauteiles in der IfcFile angegebenen IFC-Datei	
Created	DATETIME2	Datum der Erstellung dieses Datensatzes	Default Value: getutcdate()
Updated	DATETIME2	Datum der Aktualisierung dieses Datensatzes (z.B Klassifizierung)	

Abbildung 4.2: Dokumentation der Tabelle *Materials* der Datenbank

4.1.2 Konzept der Materialstrukturierung

4.1.3 Konzept für das Auslesen von Materialien aus einer IFC Datei

4.2 Algorithmen für die Textklassifizierung

4.2.1 Ablauf einer Textklassifizierung

Das Klassifizieren von Texten teilt sich oft in die folgende Stufen. Alle diese Stufen wirken sich auf das Ergebnis der Klassifizierung aus:

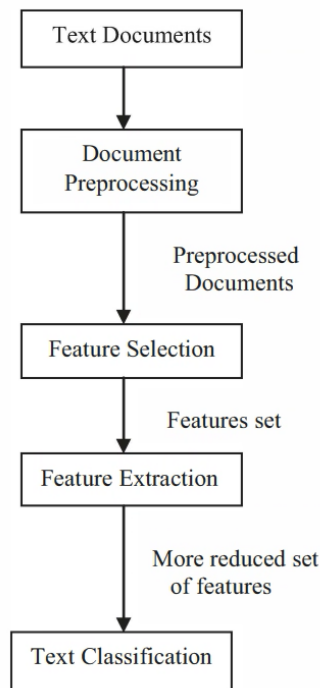


Abbildung 4.3: Stufen der Textklassifizierung (Quelle: Shah and Patel (2016))

Preprocessing

Das Preprocessing ist ein wichtiger Schritt vor der Klassifizierung. Es hat Auswirkungen auf die Genauigkeit, Interpretierbarkeit und Robustheit des Gesamtalgorithmus aus. (Gonzalez and Carlos, 2019) Materialbezeichnungen aus IFC-Dateien können verrauscht und uneinheitlich sein, da es sich um ein Freitextfeld handelt. Die Vorverarbeitung der Daten hilft, die Daten zu säubern und auf das wesentliche reduzieren. (Chandrasekar and Qian, 2016) Die Vorverarbeitungsstufe besteht normalerweise aus Aufgaben wie Tokenisierung, Entfernung von Stoppwörtern, Umwandlung in Kleinbuchstaben und Stemming. (Uysal and Gunal, 2014) Für die Materialklassifizierung werden zuerst bestimmte Zeichenketten rausgefiltert. Of befinden sich in der Bezeichnung Größen- und Farbangaben. Mithilfe von Regex werden Farben, RGB-Farbcodes (z.B *60 - 60 - 60*) und Größenangaben (z.B *400 x 300*) aus dem String entfernt. Anschließend wird der Text in Tokens und jeder Token in Kleinbuchstaben umgewandelt. Aus „*Kunststoff - grau 80-80-80*“ wird dann z.B. „*'kunststoff'*“ und aus „*Beton- C30/37 Verputzt*“ wird „*'beton', 'c30/37', 'verputzt'*“

Feature selection

Da es bei der Materialstrukturierung nur ein Feature gibt, muss keine Auswahl aus

einem Feature-Set getroffen werden. Es werden lediglich Größen- und Farbangaben aus dem Material-String im Preprocessing entfernt.

Feature extraction

Definition 4 (Feature extraction). „*Feature extraction addresses the problem of finding the most compact and informative set of features, to improve the efficiency or data storage and processing.*“ (Guyon and Elisseeff, 2006)

Zusätzlich werden in diesem Schritt beliebig formatierte Daten in eine maschinenlesbare numerische Darstellung umgewandelt. (© 2007 - 2023, scikit-learn developers, n.d.) Auch die im Preprocessing entstandenen Tokens sind noch Zeichenketten die in eine numerische Darstellung umgewandelt werden müssen.

Classification stages

Nachdem die vorherigen Schritte auf die Daten ausgeführt wurde, kann die Klassifizierung starten.

4.2.2 Sdca Maximum Entropy

4.2.3 Convolutional Neural Network

4.3 Möglichkeiten für die Feinstrukturierung

4.3.1 OpenAI Schnittstelle

OpenAI ist ein Unternehmen für KI-Forschung und -Einsatz. Sie bieten verschiedene KI-Modelle über eine Application Programming Interface (API) an. Neben Speech-ToText und einem Bildgenerierungsmodell gibt es das Sprachverarbeitungsmodell Generative Pretrained Transformer (GPT)

4.3.2 FastText mit Density Based Clustering

5 Gegenüberstellung der möglichen Konzepte

5.1 Vergleich der Textklassifizierung

5.1.1 Messkriterien

5.1.2 Auswahl eines Algorithmus

5.2 Vergleich der Feinstrukturierung

5.2.1 Messkriterien

5.2.2 Auswahl eines Algorithmus

6 Praktische Umsetzung

6.1 Implementierung des Services

6.2 Python-Interop

6.3 Implementieren des Auslesens aller Materialien aus einer IFC Datei

6.4 Strukturieren der Materialnamen

6.4.1 Implementierung des Preprocessings

6.4.2 Erstellen einer Datengrundlage

6.4.3 Training des Text-Klassifizierungsmodells

6.4.4 Training des FastText Modells

6.4.5 DBSCAN Clustering

7 Maßnahmen zur Qualitätssicherung

7.1 Clean Code

7.2 Technische Hilfsmittel

Bei der Durchsetzung von Maßnahmen der Qualitätssicherung wurden technische Hilfsmittel verwendet. Diese sind Visual Studio, Visual Studio Code, SonarQube, Azure DevOps und CI/CD. Auf diese wird in diesem Abschnitt eingegangen.

7.3 Tests und Abnahme

8 Abschluss

8.1 Bewertung der praktischen Umsetzung

8.2 Fazit

8.3 Integration in die ORCA AVA

8.4 Ausblick

Die Erweiterung soll in der Version 26 in der ORCA AVA Enterprise Edition erscheinen.

A Anhang

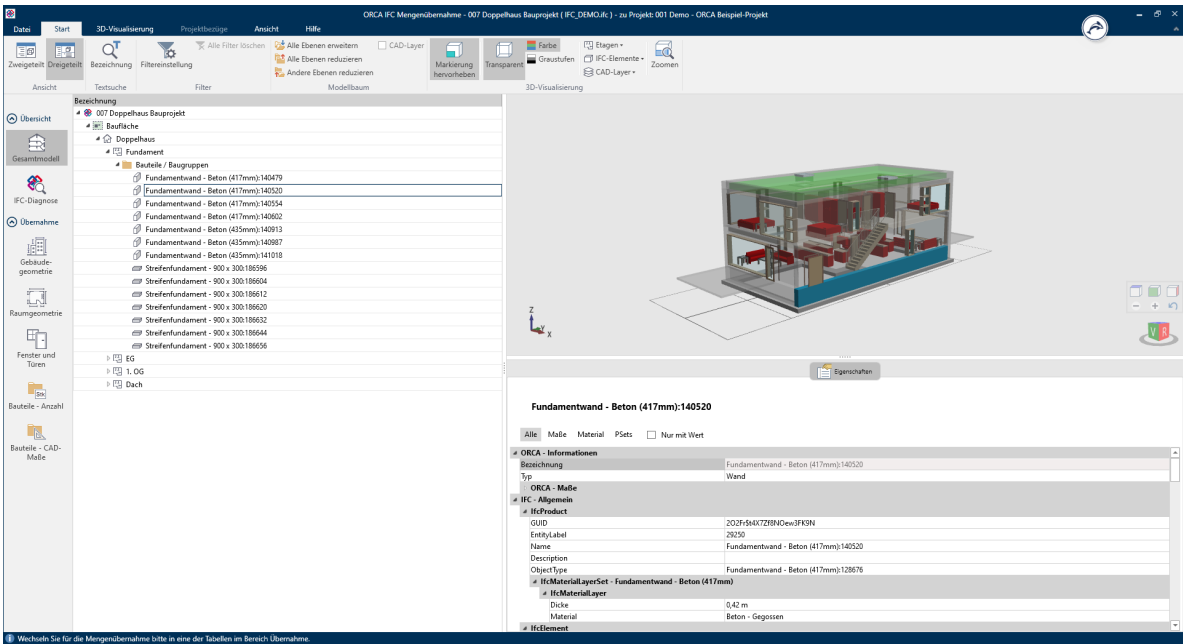


Abbildung A.1: Oberfläche des IFC Manager

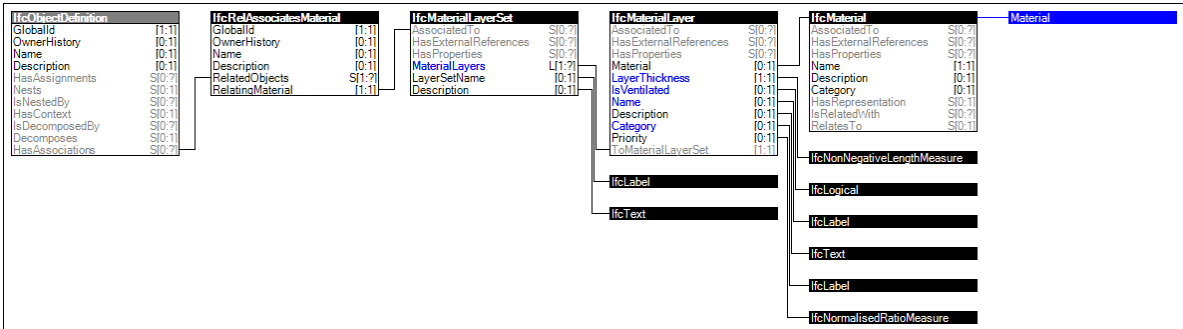


Abbildung A.2: Material Layer Set Association

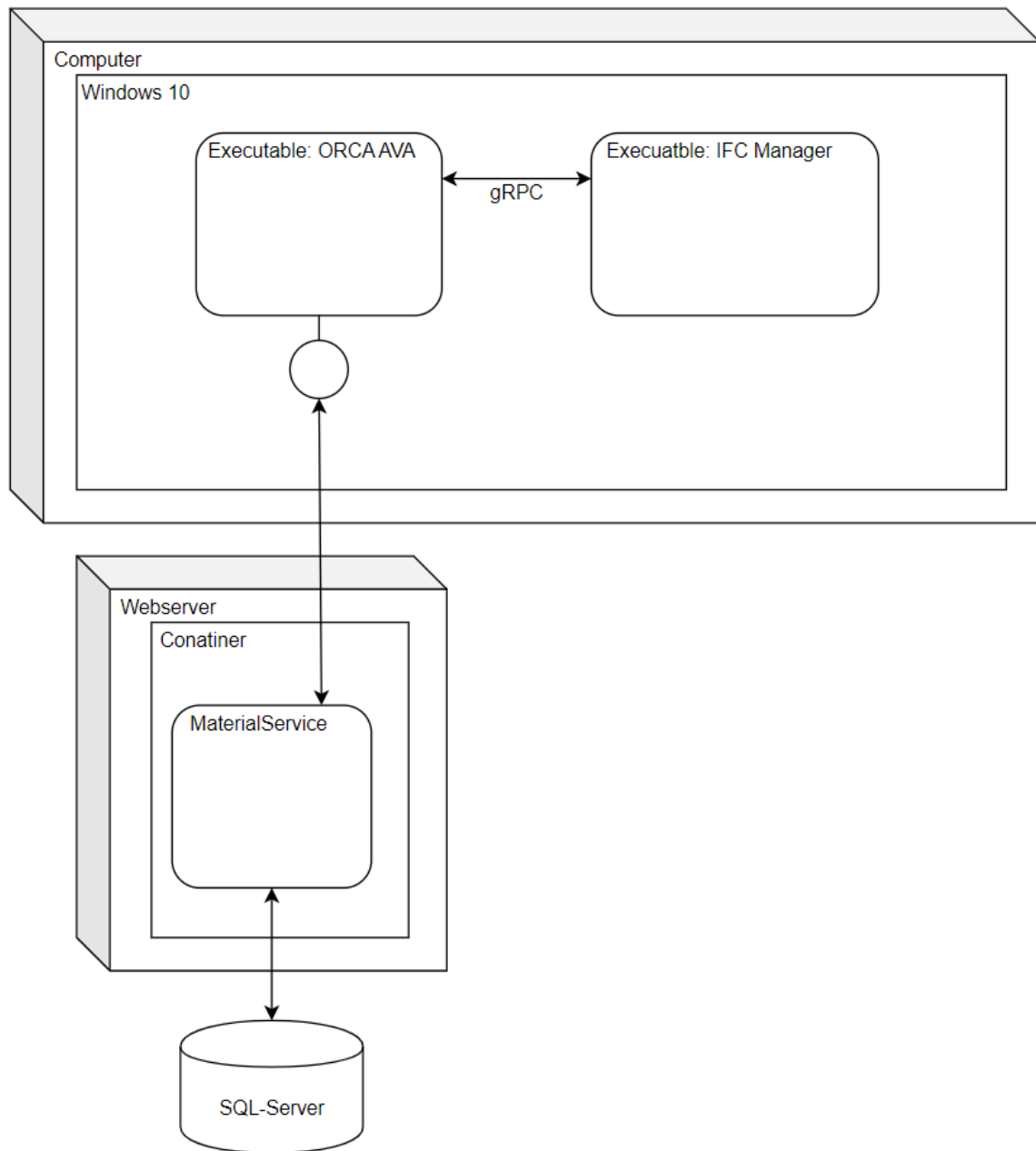


Abbildung A.6: Verteilungsdiagramm der Architektur

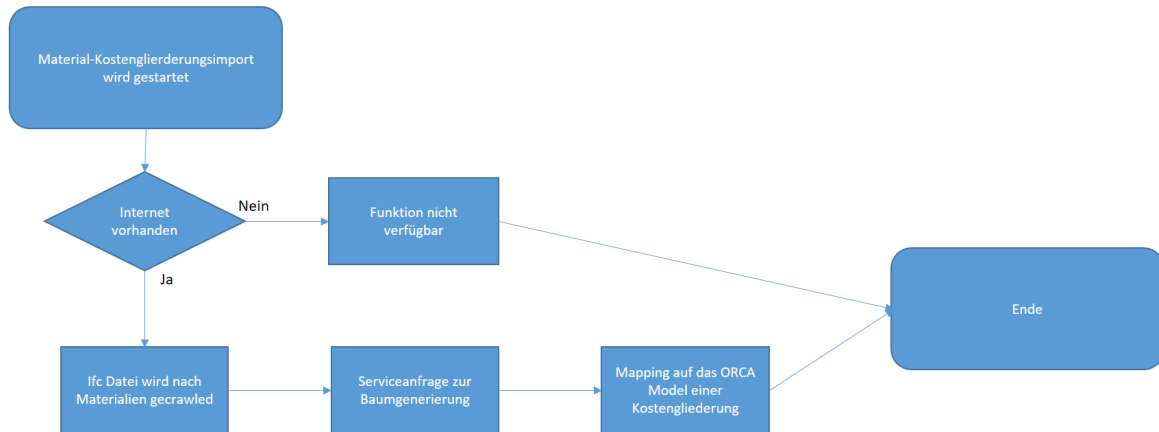


Abbildung A.7: Flussdiagramm Import der Material-Kostengliederung

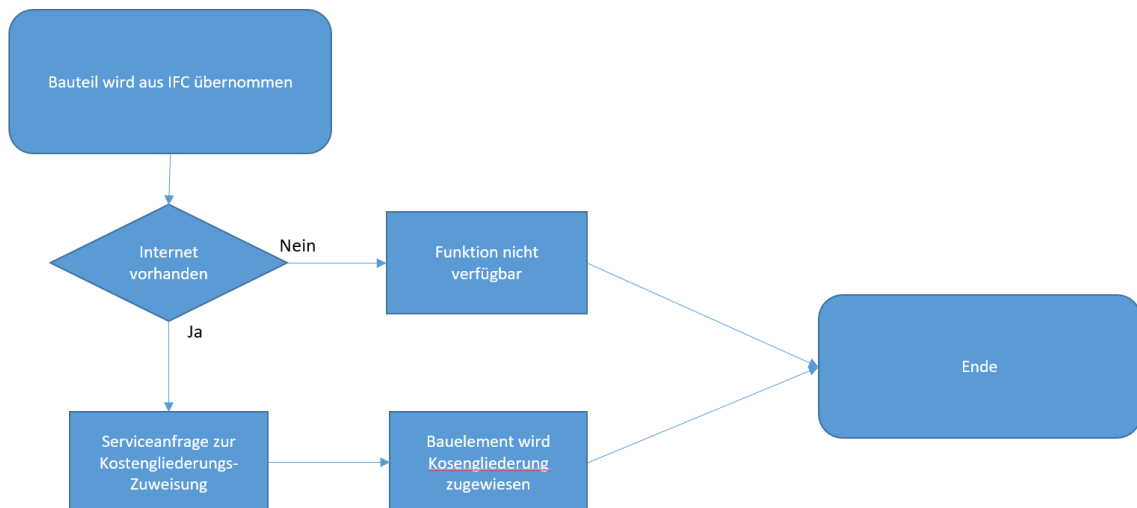


Abbildung A.8: Flussdiagramm Übernahme von Mengen aus dem IFC Manager

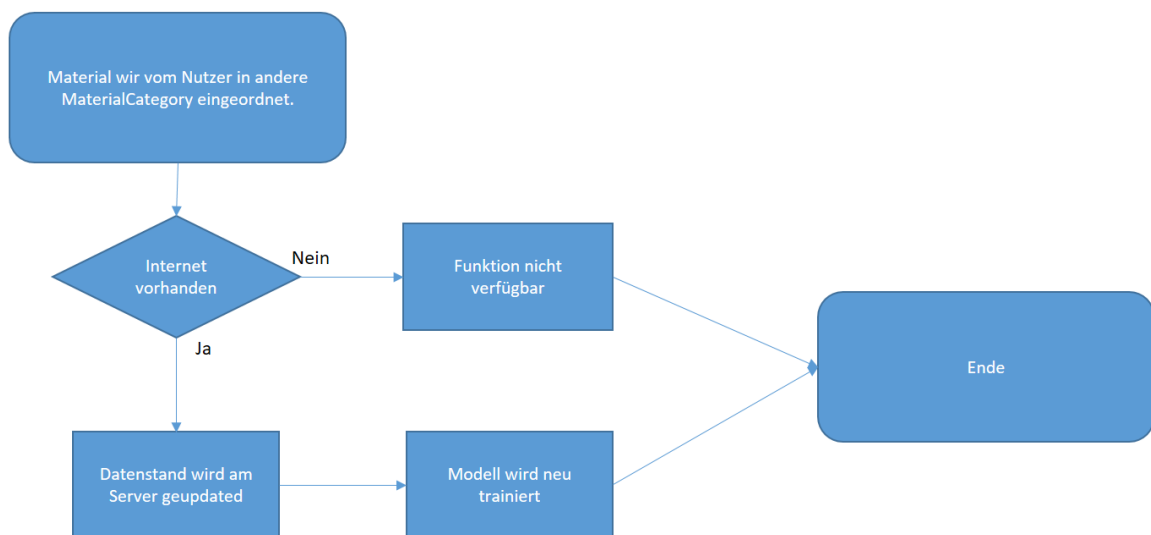


Abbildung A.9: Flussdiagramm Bewertung des Kostengliederungs-Import

Metall	<i>Eisen & Stahl</i> Eisen und Stahl Baustahl · Betonstahl/matte · Betonstahl · Spannstahl · Gusseisen · Profilstahl <i>Nichtmetalle</i> Nichtmetalle Aluminium · Magnesium · Blei · Zinn · Zink · Kupfer
Glas	<i>Glas</i> Glas Glasbaustein · Flachglas · Pressglas
Holz	<i>Holz</i> Holz Bauholz · Furnierspertholz · Leimbinder · Spertholz · Grobspanplatte · OSB-Platte · Brettschichtholz · MDF-Platte · Flachpressplatte · Multiplex-Platte · Holzwohle · Strangpressplatte
Kunststoff	<i>Thermoplaste</i> Kunststoffe (Thermoplaste) Ethylen-Tetrafluorethylen (ETFE) · Polyvinylchlorid (PVC) · Polyethylen (PE) · Polypropylen (PP) · Polyethylenmethacrylat (PMMA) · Polystyrol (PS) · Polyvinylacetat (PVAc) · Polycarbonat (PC) <i>Duroplaste</i> Kunststoffe (Duroplaste) Phenol-Formaldehyd-Harz („Bakelit“, PF) · UF-Harz · Melamin-Formaldehyd-Harz (MF) · ungesättigtes Polyester-Harz (UP) Epoxidharz (EP) <i>Elastomere</i> Kunststoffe (Elastomere) Polyurethan (PUR) · Kautschuk · Silikon · Acryl
Bitumen & Teer	<i>Dichtstoffe</i> Dichtstoffe Bitumen · Noppenbahn <i>Asphalt</i> Asphalt Walzasphalt · Gussasphalt · Naturasphalt · Halbstarrer Belag
Mineralisch	<i>Natürliche und künstliche Gesteinskörnungen</i> Natürliche und künstliche Gesteinskörnungen Sand · Kies · Bims · Hochfenschlacke · Hüttenbims · Blähton · Blähschiefer <i>Mörtel</i> Mörtel Mauermörtel · Putzmörtel · Estrichmörtel · Fliesenkleber <i>Beton</i> Beton Leichtbeton · Normalbeton · Schwerbeton · Stahlbeton · Spannbeton · Faserbeton · Stahlfaserbeton · Porenbeton <i>Naturwerkstein</i> Natürliche Bausteine (Naturwerksteine) Sandstein · Kalkstein · Granit · Schiefer · Marmor · Tuff · Grauwacke · Rhyolith · Basalt und andere <i>Kunstliche Bausteine</i> Kunstliche Bausteine Ziegel · Klinker · Tonhohlplatte · Dachziegel · Kalksandstein · Hüttenstein · Betonstein · Leichtbetonstein · Schwerbetonstein · Porenbeton · Betonwerkstein · Betondachstein · Blähziegel · Fenster, Fassaden, ...
Verbundbauteile	Fenster, Fassaden, ...
Dämmungen (siehe Arbeitshatt Dämmstoffe)	<i>Dämmstoffe</i> Dämmstoffe Flachfasern · Holzwohle-Leichtbauplate (z. B. Heraklith) · Holzweichfasenplatte · Getreideschüttungen (Ceralith) · Glasfaserdämmstoff · Hanffasern · Mineralfaserdämmstoff · Sch
Gebäudetechnik (?)	
Sonstiges	

Abbildung A.10

B Abkürzungsverzeichnis

IFC Industry Foundation Classes

IFC2x3 IFC Version 2.3

IFC4 IFC Version 4

BIM Building Information Modeling

KG Kostengruppe

DevOps Development Operations

NLP Natural Language Processing

AVA Ausschreibung, Vergabe und Abrechnung

BIM Building Information Modeling

DIN Deutsches Institut für Normung e. V.

SE Starter Edition

PE Professional Edition

EE Enterprise Edition

VB Visual Basic

GUI Graphical User Interface

WPF Windows Presentation Foundation

ISO International Organization for Standardization

STEP Standard for the exchange of product model data

SPF STEP Physical Format

XML Extensible Markup Language

CAD Computer-aided design

REST Representational State Transfer

KI Künstliche Intelligenz

IoT Internet of Things

RGB Rot, Grün, Blau

API Application Programming Interface

GPT Generative Pretrained Transformer

VB Visual Basic

C Definitionsverzeichnis

1	Definition (Scrum)	7
2	Definition (Funktionale Anforderung)	14
3	Definition (Interpretierbarkeit)	15
4	Definition (Feature extraction)	21

Literaturverzeichnis

Beetz, F. and Harrer, S. (2021), ‘Gitops: The evolution of devops?’, *IEEE Software* **39**.

buildingSMART International Ltd. (2017a), ‘Industry foundation classes - ifcmaterial’. Accessed: 2023-1-31.

URL: http://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/link/ifcmaterial.htm

buildingSMART International Ltd. (2017b), ‘Industry foundation classes 4x1 - material association’. Accessed: 2023-1-31.

URL: http://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/link/material-association.htm

buildingSMART International Ltd. (2017c), ‘Industry foundation classes 4x3- ifcpropertyset’. Accessed: 2023-1-31.

URL: https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC1/HTML/schema/ifckernel/lexical/ifcpropertyset.htm

Chandrasekar, P. and Qian, K. (2016), The impact of data preprocessing on the performance of a naive bayes classifier, *in* ‘2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)’, Vol. 2, pp. 618–619.

Glinz, M. (2007), On non-functional requirements, *in* ‘15th IEEE International Requirements Engineering Conference (RE 2007)’, pp. 21 – 26.

Gonzalez, Z. and Carlos, V. (2019), Towards explaining the effects of data preprocessing on machine learning, *in* ‘2019 IEEE 35th International Conference on Data Engineering (ICDE)’, pp. 2086–2090.

Grutters, B. (2022), ‘Industry foundation classes (ifc) – an introduction’. Accessed: 2023-2-22.

URL: <https://technical.buildingsmart.org/standards/ifc/>

Guyon, I. and Elisseeff, A. (2006), *Feature Extraction Preface*, Springer Berlin Heidelberg, Berlin, Heidelberg.

URL: https://doi.org/10.1007/978-3-540-35488-8_1

Hölzing, J. A. (2008), *Die Kano-Theorie der Kundenzufriedenheitsmessung*, Gabler Verlag Wiesbaden.

Industry Foundation Classes (2017). Accessed: 2023-1-31.

URL: http://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML

Ken, S. and Jeff, S. (2020), ‘The scrum guide’.

URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

Laakso, M. and Kiviniemi, A. O. (2012), ‘The IFC standard: A review of history, development, and standardization, information technology’, *J. Inf. Technol. Constr.* **17**(9), 134–161.

Microsoft (2022a), ‘Auswählen eines algorithmus für maschinelles lernen - azure machine learning’. Accessed: 2023-3-10.

URL: <https://learn.microsoft.com/de-de/azure/machine-learning/how-to-select-algorithms>

Microsoft (2022b), ‘Windows presentation foundation (WPF)’. Accessed: 2023-2-10.

URL: <https://github.com/dotnet/wpf>

Miller, T. (2017), ‘Explanation in artificial intelligence: Insights from the social sciences’.

URL: <https://arxiv.org/abs/1706.07269>

Molnar, C. (2022), *Interpretable Machine Learning*, 2 edn.

URL: <https://christophm.github.io/interpretable-ml-book>

.NET Foundation and Contributors (n.d.), ‘ASP.NET core’. Accessed: 2023-3-20.

URL: <https://github.com/dotnet/aspnetcore/blob/main/README.md>

ORCA Software GmbH (n.d.), ‘ORCA helpdesk - ORCA AVA 25 - kostengliederungen als basis für das kostenmanagement’. Accessed: 2023-2-1.

URL: https://helpdesk.orca-software.com/Solution/AVA25/Content/Kontexthilfe-337_8277

- Shah, F. P. and Patel, V. (2016), A review on feature selection and feature extraction for text classification, *in* ‘2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)’, pp. 2264–2268.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R. (2013), ‘Intriguing properties of neural networks’.
URL: <https://arxiv.org/abs/1312.6199>
- Thomas, Baumanns and Dr. Philipp-Stephan, Freber and Dr. Kai-Stefan, Schober and Dr. Florian, Kirchner (2016), ‘Bauwirtschaft im wandel - trends und potenziale bis 2020’.
- Thomas, L., Yoshinobu, A., James, F., Juha, H., Kari, K., Kent, R., Stefan, R. and buildingSMART International Ltd. (2007), ‘Ifcmateriallist’. Accessed: 2023-1-31.
URL: <https://standards.buildingsmart.org/IFC/RELEASE/IFC2x3/TC1/HTML/ifcmaterialresource/lexical/ifcmateriallist.htm>
- Uysal, A. K. and Gunal, S. (2014), ‘The impact of preprocessing on text classification’, *Information Processing & Management* **50**(1), 104–112.
URL: <https://www.sciencedirect.com/science/article/pii/S0306457313000964>
- van Berlo, L. (2022), ‘Ifc specifications database’. Accessed: 2023-2-22.
URL: <https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/>
- Xbim Ltd. (n.d.), ‘Xbim toolkit’. Accessed: 2023-1-31.
URL: <https://docs.xbim.net/research/history.html>
- © 2007 - 2023, scikit-learn developers (n.d.), ‘Feature extraction’. Accessed: 2023-2-10.
URL: https://scikit-learn.org/stable/modules/feature_extraction.html