



DIPLOMARBEIT

**QUANTIFIZIERUNG VON EINFLUSSPARAMETERN
FÜR DAS FEATURE ENGINEERING UND
NUTZUNG DES MACHINE LEARNING FÜR DEN
VORENTWURF VON BIM-MODELLEN SOZIALER
WOHNUNGSBAUTEN**

Bearbeiter:

cand. ing. Johannes Frank Schüler

Matrikelnummer: 4039308

zur Erlangung des akademischen Grades:

Diplom-Ingenieur (Dipl.-Ing.)

Betreuer:

Prof. Dr.-Ing. habil. Karsten Menzel (TU Dresden)

Christoph Großmann, M.Eng. (IPROconsult GmbH)

Betreuender Hochschullehrer:

Prof. Dr.-Ing. habil. Karsten Menzel

Eingereicht am: 06.09.2019



Aufgabenstellung für die Diplomarbeit

Name: Johannes Frank Schüler

Vertiefung: Konstruktiver Ingenieurbau

Thema: Quantifizierung von Einflussparametern für das Feature Engineering und Nutzung des Machine Learning für den Vorentwurf von BIM-Modellen sozialer Wohnungsbauten

(Quantification of influencing parameters for feature engineering and use of machine learning for the preliminary design of BIM models of social housing)

Zielsetzung:

Die Mieten und Wohnungspreise steigen seit einigen Jahren immer mehr an. Einer der Gründe dafür ist die Abwanderung der Landbevölkerung in die Städte. Damit wächst die Einwohnerzahl bei gleichbleibendem Wohnungsangebot. Gleichzeitig findet eine Privatisierung des Wohnungsbaus statt, bei dem sich das Ziel der Renditenmaximierung über die Gemeinnützigkeit stellt. Bevölkerungsgruppen mit geringen Einkommen sind dadurch kaum noch in der Lage, sich Wohnraum in den Städten leisten zu können.

Um diesem Problem Abhilfe zu schaffen sind kostengünstige Lösungen zur Schaffung von Wohnraum gefragt. Ein Entwurf von BIM-Modellen mittels Machine Learning ermöglicht die optimale Anpassung an die gegebenen Zielparameter. So kann beispielsweise automatisiert die Ausnutzung der Grundfläche maximiert werden, ohne per Hand unzählige Variationen durchzugehen. Solche Optimierungen lassen sich auf beliebige Zielgrößen anwenden und könnten damit nicht nur im sozialen Wohnungsbau eingesetzt werden.

Um den Entwurfsprozess von Maschinen ausführen zu lassen, ist eine Quantifizierung der Eingabeparameter von Nöten. Beim Entwurf von Wohnungen muss sich der Architekt an den Zielgrößen des Auftraggebers orientieren. Die Anforderungen sind nicht immer für die digitale Verarbeitung geeignet und müssen erst transformiert werden, damit ein Expertensystem damit arbeiten kann. Dieses wiederum muss das nötige Expertenwissen vorhalten, um die quantifizierten Daten qualitativ bewerten zu können.

Arbeitsumfang:

Während der Ausarbeitung sollen folgende Punkte bearbeitet werden:

1. Überblick über die Funktionsweisen des Machine Learning im Zusammenhang mit Methoden des Feature Engineerings
2. Ermittlung des notwendigen Expertenwissens im Bereich des Wohnungsentwurfes
3. Definition der notwendigen Zielparameter im sozialen Wohnungsbau
4. Quantifizierung ausgewählter Einflussparameter im Hinblick auf die spätere Nutzbarkeit im Machine Learning
5. Entwicklung eines Software-Prototypen zur Optimierung der gewählten Einflussparameter mittels Machine Learning

Wissenschaftlicher Betreuer:

Prof. Dr.-Ing. habil. Karsten Menzel
Institut für Bauinformatik
Technische Universität Dresden

Externer Betreuer:

Christoph Großmann, M.Eng.
BIM-Manager, IT
IPROconsult GmbH

Ausgehändigt am: 07.05.2019

Einzureichen am: 07.09.2019



Prof. Dr.-Ing. habil. Karsten Menzel
Verantwortlicher Hochschullehrer

INHALTSVERZEICHNIS

Abbildungsverzeichnis.....	V
Abkürzungsverzeichnis.....	VIII
1. Einleitung	9
1.1. Problemstellung	9
1.2. Ziele und Abgrenzung.....	10
1.3. Aufbau der Arbeit.....	11
2. Maschinelles Lernen	13
2.1. Allgemeines	13
2.1.1. Lernarten	13
2.1.2. Fluch der Dimensionalität	15
2.1.3. Fehlerquellen.....	16
2.1.4. Evaluierung	16
2.1.5. Overfitting	17
2.2. Feature Engineering.....	18
2.3. k-Nearest-Neighbor	20
2.4. Support Vector Machines	21
2.5. Clustering.....	23
2.6. Evolutionäre Strategien	23
2.7. Reinforcement Learning.....	24
2.8. Decision-Trees	25
2.9. Neuronale Netze	27
2.9.1. Grundlagen	27
2.9.2. Backpropagation	28
3. Grundlagen des Entwerfens.....	30
3.1. Parameter	30
3.1.1. Funktionen.....	32

3.1.2.	Bewohner.....	33
3.1.3.	Gebäudeanordnung.....	34
3.1.4.	Wohnungsgrundriss.....	37
3.1.5.	Raumentwurf.....	42
3.2.	Entwurfsprozess.....	44
4.	Sozialer Wohnungsbau	46
4.1.	Strategien für erschwinglichen Wohnraum	47
4.2.	Förderungskriterien	49
5.	Prototypische Umsetzung.....	54
5.1.	Konzeption.....	54
5.1.1.	Problemstellung	54
5.1.2.	Grundrissgenerierung	56
5.1.3.	Wahl der Machine Learning Methode	59
5.1.4.	Bewertungskriterien	61
5.2.	Implementierung	63
5.3.	Evaluierung.....	76
6.	Schlussbetrachtung	78
6.1.	Zusammenfassung und Ergebnisse	78
6.2.	Ausblick.....	79
	Literaturverzeichnis.....	81
	Selbstständigkeitserklärung.....	84

ABBILDUNGSVERZEICHNIS

Abbildung 1: Angebotsmieten 2004 bis 2016 (BBSR, 2017).....	9
Abbildung 2: Bearbeitungsschritte zur Aufgabenerfüllung (eigene Darstellung)....	11
Abbildung 3: Machine-Learning-Ansatz (Géron, et al., 2018)	13
Abbildung 4: Lernverfahren und ihre Modelle (Frauenhofer-Gesellschaft, 2018 S. 10)	14
Abbildung 5: Knowledge Discovery in Databases (übersetzt: Wissensfindung in Datenbanken) (Frochte, 2018 S. 16)	18
Abbildung 6: Klassifizierung des roten Datenpunktes mit Hilfe der $k = 5$ nächsten Nachbarn (eigene Darstellung).....	20
Abbildung 7: Klassifizierung der Datenpunkte durch eine Trennebene aus Unterstützungsvektoren (eigene Darstellung)	21
Abbildung 8: SVM mit verschiedenen Kernelarten (Frochte, 2018 S. 300)	22
Abbildung 9: Verfahrensweise beim Clustering (Frochte, 2018 S. 304).....	23
Abbildung 10: Beispielhafte Kreuzung des genetischen Codes zweier Individuen (eigene Darstellung).....	24
Abbildung 11: Einsatzaufgabe für Reinforcement Learning (eigene Darstellung)...	25
Abbildung 12: Beispiel für einen Entscheidungsbaum (Frochte, 2018 S. 121).....	26
Abbildung 13: künstliches Neuron (Weinberger, et al., 1994 S. 102).....	27
Abbildung 14: Neuronales Netz mit 4 Schichten (Ognjanovski, 2014).....	28
Abbildung 15: Altersdifferenzierte Bevölkerungsentwicklung in Sachsen (SMI, 2019)	34
Abbildung 16: Beispielgebäude mit Abstandsflächen (Schröder, 2016).....	35
Abbildung 17: Erschließung mit offenem Ende in einem Gemeinschaftsraum (Heckmann, et al., 2018 S. 48).....	38
Abbildung 18: Sonneneinfall nach Tageszeit und Himmelsrichtung (Krebs, et al., 2014 S. 16)	39
Abbildung 19: Gegensätzliche Kriterien zur Zonierung einer Wohnung (Krebs, et al., 2014)	40

Abbildung 20: Musterschlafzimmer (Krebs, et al., 2014 S. 37)	42
Abbildung 21: Flächenoptimierte Badezimmer bieten oft wenig Aufenthaltsqualität (Krebs, et al., 2014 S. 50)	43
Abbildung 22: Funktionsdiagramm (Heckmann, et al., 2018 S. 56)	45
Abbildung 23: Studentenwohnheim Wien 1020 (STUWO, 2012)	46
Abbildung 24: Co-Living-Spaces im Vergleich zu Einzelabnehmern (ollie.co, 2019)	48
Abbildung 25: Angemessenheitsrichtwerte Bruttokaltmiete (Landeshauptstadt Dresden, 2019).....	52
Abbildung 26: Maximale Haushaltsgrößen und deren Bedarfsanteil (Landeshauptstadt Dresden, 2018).....	53
Abbildung 27: Layout mittels dichtester Packung (eigene Darstellung).....	57
Abbildung 28: Layout mittels k-dimensionaler Bäume (eigene Darstellung).....	58
Abbildung 29: Voronoi-Diagramm (eigene Darstellung).....	58
Abbildung 30: Traditionelles Vorgehen (blau), Erzeugung eigener Daten (rot) (eigene Darstellung).....	60
Abbildung 31: Evolutionäre Strategie (eigene Darstellung).....	60
Abbildung 32: Konzeptionierung des Softwareprototypen (eigene Darstellung)....	63
Abbildung 33: Input-Tabelle für den Benutzer (eigene Darstellung)	64
Abbildung 34: Datenimport aus einer Excel-Arbeitsmappe (eigene Darstellung) ...	65
Abbildung 35: Generierung zufälliger Starträume (eigene Darstellung)	65
Abbildung 36: Visualisierung der Startpunkte auf der Wohnungsfläche (eigene Darstellung).....	66
Abbildung 37: Kollisionsalgorithmus in Python (eigene Darstellung).....	66
Abbildung 38: Visualisierung der zu überprüfenden Kollisionsbereiche (links) und die erste Ausdehnung (rechts) (eigene Darstellung).....	67
Abbildung 39: Algorithmus zum Berechnen des Abstandes zu den nächsten Nachbarn und der Ausdehnung zu diesen (eigene Darstellung).....	68
Abbildung 40: Visualisierung des verdichteten Kerns (links) und des Endergebnisses (rechts) (eigene Darstellung).....	68

Abbildung 41: Algorithmus zur Ausdehnung der Kernzone zu den Außenwänden (eigene Darstellung).....	69
Abbildung 42: Überprüfungsfunktion auf valide Grundrisse (eigene Darstellung) .	70
Abbildung 43: Bewertungsfunktion der Layoutentwürfe (eigene Darstellung)	71
Abbildung 44: Kreuzungsprozess und Übernahme der vier fittesten Individuen (eigene Darstellung).....	72
Abbildung 45: Funktion zur Transformation und zum Export der Koordinaten (eigene Darstellung)	73
Abbildung 46: Dynamo-Script zum Erstellen des BIM-Modells in Revit (eigene Darstellung)	74
Abbildung 47: 3D-Ansicht (links) und Grundriss (rechts) des erzeugten Wohnungslayouts (eigene Darstellung)	74
Abbildung 48: Bewertungsfunktion der Fensterorientierung (eigene Darstellung)	75
Abbildung 49: Anwendung der Ungarischen Methode in Python (eigene Darstellung)	76
Abbildung 50: Verbesserung der Bewertung der Grundrisse durch reine Kreuzung (eigene Abbildung).....	77

ABKÜRZUNGSVERZEICHNIS

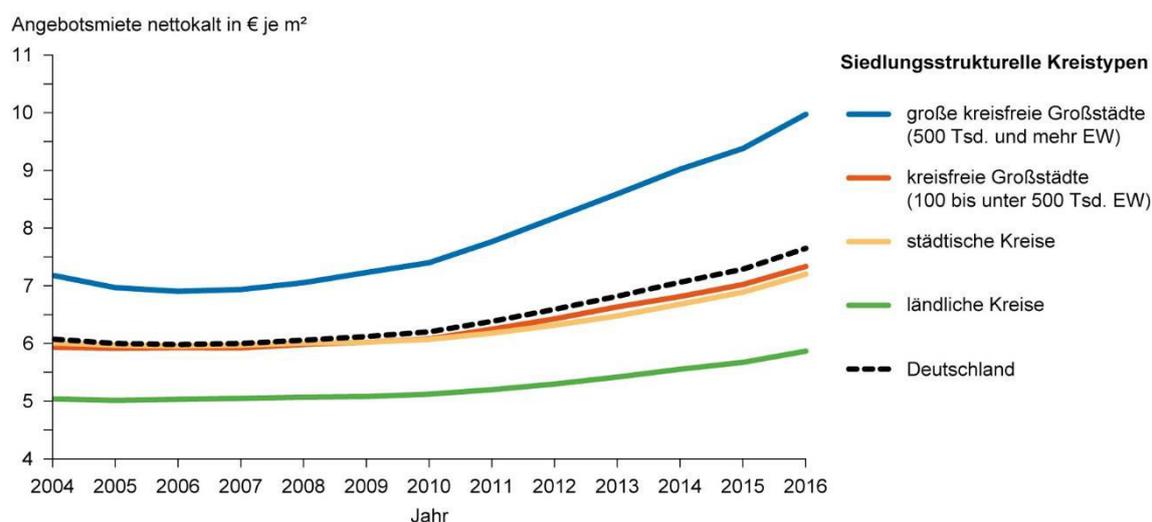
BIM	Building Information Modeling
DB	Datenbank
EW	Einwohner
MBO	Musterbauordnung
ML	Machine Learning
MRFIw	Muster-Richtlinien über Flächen für die Feuerwehr
RBF	radial basis function
RL	Richtlinie

1. EINLEITUNG

1.1. PROBLEMSTELLUNG

Wohnraum in Großstädten wird immer kostspieliger, teilweise so teuer, dass sich einkommensschwache Bevölkerungsgruppen die Mieten in Zentrumsnähe nicht mehr leisten können. Diese Misslage ist das Ergebnis marktwirtschaftlicher Prozesse, da dem begrenzten Angebot an Wohnfläche im Stadtinneren eine stetig steigende Nachfrage gegenübersteht. Das Wohnungsangebot wird durch Vermietungsplattformen und einen steigenden Pro-Kopf-Flächenverbrauch zusätzlich verknappt. Zudem erhöht sich die Zahl der Einwohner in den Städten. Die Gründe dafür liegen in der Landflucht, bedingt durch eine wirtschaftliche Zentralisierung in den Metropolen, dem stetigen Wachstum der Bevölkerung und der Zuwanderung. Auch der demografische Wandel nimmt Einfluss auf die Verknappung des Wohnraumangebots. Denn die im Bundesdurchschnitt älter werdende Bevölkerung behält meist nach dem Auszug der Kinder die nun freie Wohnfläche aus Gründen der Gewohnheit und der Verbundenheit zum Wohnumfeld. Falls doch ein Umzug in Erwägung gezogen wird, wird meistens auf Barrierefreiheit geachtet, welche einen höheren Platzbedarf mit sich bringt und so die gehobenen Ansprüche älterer Mitmenschen an den Wohnraum widerspiegelt.

Angebotsmieten nach Kreistypen 2004 bis 2016



Anmerkung: Mieten für Erst- und Wiedervermietung
Datenbasis: BBSR-Wohnungsmarktbeobachtung, IDN ImmoDaten GmbH

© BBSR Bonn 2017

Abbildung 1: Angebotsmieten 2004 bis 2016 (BBSR, 2017)

Die momentan gebauten Wohnungen reduzieren zwar den Wohnungsmangel, sind aber preislich nur besser verdienenden Bevölkerungsschichten zugänglich. Die

Neubauten erhalten aufgrund des modernen Zustandes ihrer Ausstattung und Bauweise meist zusätzliche Preisaufschläge. Diese Investition im oberen Preissegment steigert den Preisauftrieb anstatt ihn zu bremsen, da so der Mietspiegel angehoben wird.

1988 wurden die Steuervorteile für gemeinnützige Wohnungsbauten abgeschafft. Hinzu kam die Abgabe der Verantwortung für die Schaffung sozialen Wohnraums vom Bund an die Länder im Rahmen der Föderalismusreform von 2006. Die Länder wiederum verlagerten die Aufgabe an die Städte und Kommunen, welche auch aktuell nicht die benötigten finanziellen Mittel besitzen und vorhandene Wohnungsgesellschaften gewinnorientiert ausrichten. Dies verursachte einen Rückgang des sozialen Wohnungsbaus in Deutschland. Wohnungssuchende sind seither auf staatliche Subjektförderung in Form von Wohngeld, Wohnkostenzuschüssen des Arbeitslosengeldes II oder Wohnungsbeihilfe angewiesen. (Dömer, et al., 2016 S. 22-23)

1.2. ZIELE UND ABGRENZUNG

Statt mit Fördergeldern die Mieteinnahmen der privaten Wohnungsanbieter zu steigern, könnte damit der soziale Wohnungsbau unterstützt werden. Denn langfristig gesehen, würde sich die Investition staatlicher Gelder in den sozialen Wohnungsbau für den Bund rechnen, da sich die selbst errichteten Gebäude in der Zukunft durch die eingesparte Subjektförderung amortisieren werden. Um die Schaffung von mietpreis- und belegungsgebundenem Mietwohnraum zu vereinfachen, soll diese Arbeit dazu dienen, die Planung, genauer gesagt den Entwurf der einzelnen Wohnungen zu vereinfachen.

Dabei ist die Planung von vielen verschiedenen Parametern abhängig. Diese werden nicht nur durch die Gesetzgebung bestimmt, sondern auch durch die Bereiche der Ästhetik und Soziologie beeinflusst. Grundsätzlich ist jedes Bauprojekt ein Individualvorhaben, da schon allein aufgrund des Standorts ständig wechselnde Umgebungsbedingungen beachtet werden müssen. Dafür ist eine hohe Anzahl an Spezialisten der einzelnen Fachbereiche nötig, um die Parameter qualitativ berücksichtigen zu können. Die Planung eines Bauwerkes beginnt immer mit dem Vorentwurf durch einen Architekten. Dieser Anfangsschritt soll ausführlich auf Verbesserungspotential untersucht werden.

Effizienz wird heutzutage oft durch Automatisierung erreicht. Dazu eignen sich vor allem neue digitale Technologien und die Nutzung steigender Rechenkapazitäten.

Vor diesem Hintergrund wird nach einem Ansatz gesucht, den bisher händischen Entwurfsprozess der Architekten mit Hilfe von digitaler Datenverarbeitung zu unterstützen und damit zu optimieren. Aus dem großen Gebiet der Künstlichen Intelligenz wird das Teilgebiet des Machine Learning genauer beleuchtet und eine mögliche Kombination von Technik und Architektur ergründet.

1.3. AUFBAU DER ARBEIT

Für die Bearbeitung der zu dieser Arbeit zugehörigen Aufgabenstellung ist eine umfassende Einarbeitung in die Themen Machine Learning und Grundlagen des Entwerfens essentiell, um später geeignete Punkte zur Verknüpfung zu entdecken. Der Aufbau der Arbeit orientiert sich an der Bearbeitungsreihenfolge wie in Abbildung 2 dargestellt.

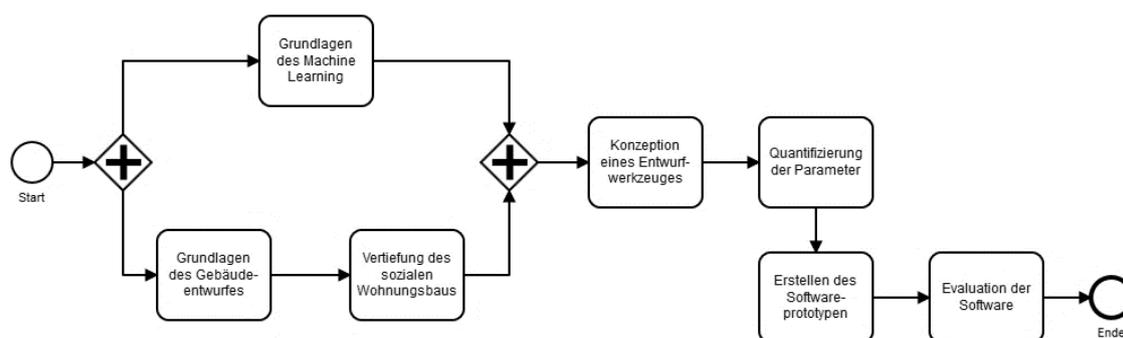


Abbildung 2: Bearbeitungsschritte zur Aufgabenerfüllung (eigene Darstellung)

Zu Beginn der Arbeit steht eine Einführung in das Maschinelle Lernen. Dabei wird auf die Grundlagen verschiedener Lernmethoden eingegangen. Auch allgemeine Probleme, wie Overfitting und der Fluch der Dimensionalität, werden aufgegriffen. Im Anschluss daran werden die einzelnen Verfahren des Machine Learning in ihren Funktionsweisen genauer untersucht.

Im Bereich des Gebäudeentwurfs folgt nach einem kurzen Überblick der verschiedenen Randbedingungen die Auswertung gestalterischer und sozialer Faktoren, an welchen sich Architekten bei der Ausarbeitung von Gebäudeentwürfen orientieren. Dabei folgt der Blick vom Großen ins Kleine, das heißt zuerst werden ganze Viertel, dann Gebäude, dann Wohnungen mit ihren einzelnen Räumen und zum Schluss der Bewohner betrachtet.

Im Anschluss daran wird explizit der soziale Wohnungsbau untersucht. Zugleich sollen Konzepte für erschwinglichen Wohnraum vorgestellt werden. Hieraus ergeben sich weitere Parameter, die in die Umsetzung einfließen könnten. Sehr

wichtig sind die Förderungskriterien für soziale Wohnungsbauten durch die Kommunen und Städte.

Letztendlich können nun die erarbeiteten Informationen zu einem Konzept für eine entwurfsunterstützende Software für Architekten zusammengeführt werden. Nach einer ausführlichen Beschreibung der Implementierung folgt die Evaluation des Software-Prototypen.

2. MASCHINELLES LERNEN

2.1. ALLGEMEINES

Maschinelles Lernen ist ein iterativer Prozess, bei dem aus bestehenden oder generierten Daten gelernt werden soll. Je nach Art der Aufgabe und Vorkommen der Daten können verschiedene Algorithmen zum Einsatz kommen. Das eigentliche Training findet über eine wiederkehrende Anpassung des Fehlers statt, um diesen zu minimieren.

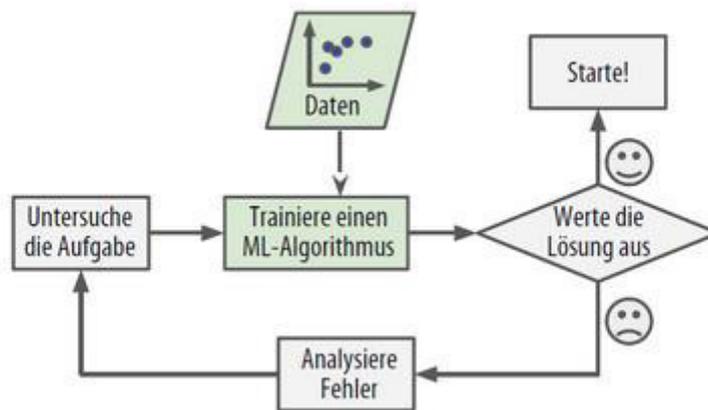


Abbildung 3: Machine-Learning-Ansatz (Géron, et al., 2018)

2.1.1. LERNARTEN

Ein Ansatz des Machine Learning ist das Eager-Learning (übersetzt: eifriges Lernen), bei dem für die Erzeugung des Modells der meiste Aufwand während des Trainings betrieben wird. Im Gegensatz dazu gibt es auch das Lazy-Learning (übersetzt: träges Lernen), bei welchem erst zum Abfragezeitpunkt die Hauptarbeit stattfindet. Diese Modelle sind zwar langsamer in der Abfrage, bilden jedoch ein genaueres, lokales Modell, das besser auf die jeweiligen Anforderungen zugeschnitten ist, da es einen aktuelleren Wissensstand besitzt. (Frochte, 2018 S. 110)

Obwohl alle Lernarten das Abbilden einer Funktion als Ziel verfolgen, werden die Lernarten des Machine Learning am häufigsten dadurch unterschieden, ob beim Lernen gelabelte Daten genutzt werden oder nicht. Ein Label ist der konkrete Funktionswert eines Datenpunktes auf der abzubildenden Funktion und muss häufig durch einen Menschen erstellt werden. Durch die enorme Menge an Daten, die zum Trainieren einer Maschine erforderlich sind und die Kosten einer

menschlichen Arbeitskraft, ist die Erstellung der Label im Vergleich zur Beschaffung ungelabelter Daten verhältnismäßig teuer. Deshalb werden gelabelte Daten auch als das Gold unter den Daten bezeichnet (Frochte, 2018 S. 20).

Lernstil	Lernaufgabe	Lernverfahren	Modell
Überwacht	Regression	Lineare Regression	Regressionsgerade
		Klassifikations- und Regressionsbaumverfahren (CART)	Regressionsbaum
	Klassifikation	Logistische Regression	Trennlinie
		Iterative Dichotomizer (ID3)	Entscheidungsbaum
		Stützvektormaschine (SVM)	Hyperebene
		Bayessche Inferenz	Bayessche Modelle
Unüberwacht	Clustering	K-Means	Clustermittelpunkte
	Dimensionsreduktion	Kernel Principal Component Analysis (PCA)	Zusammengesetzte Merkmale
Bestärkend	Sequentielles Entscheiden	Q-Lernen	Strategien
Verschiedene	Verschiedene	Rückwärtspropagierung	Künstliche Neuronale Netze

Abbildung 4: Lernverfahren und ihre Modelle (Frauenhofer-Gesellschaft, 2018 S. 10)

Das Supervised Learning (übersetzt: überwachtes Lernen) benötigt genau diese Label während des Trainings, um aus den Beispielen zu lernen. Dabei induziert es Schlussfolgerungen aus den gesammelten Informationen und vergleicht diese mit den zuvor erstellten Labeln. Es gibt zwei fundamentale Prinzipien der Zielfindung beim Supervised Learning. Die Klassifikation ordnet einen neuen Datenpunkt einer bestimmten Gruppe mit ähnlichen Eigenschaften zu und klassifiziert so das Objekt. Es erhält damit lediglich einen festgelegten Klassennamen. Dagegen wird bei der Regression versucht, eine skalare Zielgröße vorherzusagen, indem die Veränderungen anderer Objekteigenschaften betrachtet werden. (Géron, et al., 2018 S. 8-9)

Im Gegensatz dazu wird beim Unsupervised Learning ohne Label gearbeitet. Darunter versteht man beispielsweise das Clustering, bei dem ungelabelte Objekte mit ähnlichen Eigenschaften zu Clustern zusammengefasst werden. Dies eignet sich gleichzeitig perfekt für eine Visualisierung der Daten, womit das Erkennen von Anomalien einhergeht. Kommen gelabelte und ungelabelte Daten vermischt zum Einsatz, spricht man von Semi-Supervised Learning (übersetzt: halbüberwachtes Lernen). Dabei werden beispielsweise zuerst alle Daten geclustert und später die

Label der Mehrzahl der Objekte eines Clusters den ungelabelten Datenpunkten desselben Clusters zugeordnet. (Géron, et al., 2018 S. 10-12)

Ein weiterer Lerntypus ist das Reinforcement Learning (übersetzt: bestärkendes Lernen). Dabei wird nach dem Trail-and-Error-Prinzip vorgegangen, indem die hier als Agent bezeichnete Maschine Belohnungen oder Bestrafungen für eine bestimmte Handlung erhält. (Görz, et al., 2014 S. 461-462)

Zur weiteren Unterscheidung kann die Art der Aktualisierung des Lernmodelles herangezogen werden. Beim Batch bzw. Offline Learning wird das Training auf allen verfügbaren Daten und in einem Schritt durchgeführt, wodurch die Rechenkapazität des genutzten Systems stark ausgelastet und je nach Größe des Datensatzes für längere Zeit belegt wird. Erst danach wird das Modell aktiv geschaltet und kann mit unbekanntem Daten verwendet werden. Kommt ein neues Trainingsbeispiel hinzu, muss der gesamte Datensatz von Beginn an erneut erlernt und später das vorherige Modell ersetzt werden. Bei häufigem Zuwachs der Trainingsbeispiele ist ein inkrementelles Lernverfahren wie das Online Learning praktikabler. Dabei werden Mini-Batches in kleinen Schritten gelernt und anschließend gelöscht. So wird die Kapazität des Systems geschont und vor allem Speicherplatz eingespart. Über die Lernrate kann der Einfluss der neuen Daten auf das Modell reguliert werden. Zum Beispiel hat eine niedrige Lernrate einen trägen Effekt auf die Einarbeitung neuer Erkenntnisse. (Géron, et al., 2018 S. 14-16)

Die beim Machine Learning getroffenen Schlussfolgerungen sind entweder induktive oder deduktive Inferenzen. Bei der Induktion wird von einzelnen Instanzen über ein Ähnlichkeitsmaß auf das neue Objekt geschlossen, weshalb man diese Art der Wissensbildung auch als instanzbasiertes Lernen bezeichnet. Das modellbasierte Lernen hingegen nutzt Deduktion, bei welcher aus dem Modell vorhandene Parameter angepasst und auf neue Instanzen übertragen werden. (Keller, 2000 S. 20-21)

2.1.2. FLUCH DER DIMENSIONALITÄT

Die Wahrscheinlichkeit, dass ein beliebiger Punkt eines Einheitsquadrates ein Randpunkt (also ein Extrempunkt) ist, liegt bei 0,4 % und der durchschnittliche Abstand zweier zufälliger Punkte auf dem gesamten Quadrat beträgt 0,52 Längeneinheiten. Vergleichsweise dazu steigt die Extrempunkt Wahrscheinlichkeit eines Hyperwürfels im 1.000.000-dimensionalen

Raum auf 99,9 % und der Abstand erhöht sich auf 408,25 Längeneinheiten. (Géron, et al., 2018 S. 206)

Dieser Umstand verdeutlicht ein Problem im mehrdimensionalen Raum. Wenn wenige Datensätze (Punkte) mit einer hohen Anzahl an Attributen (Dimensionen) in maschinellen Lernen zum Einsatz kommen, spricht man vom Fluch der Dimensionalität (Frochte, 2018 S. 109). Dieser besagt, dass bei der Beschreibung eines Raums mit einer konstanten Anzahl von Merkmalen, in höheren Dimensionen eine schlechtere Beschreibung durch diese Merkmale stattfindet, da die Informationsdichte rapide abnimmt.

2.1.3. FEHLERQUELLEN

Der Erfolg des gewählten Lernmodells hängt von zwei Faktoren ab:

- Gewählte Lernmethode
- Qualität und Quantität der Daten

Die Lernmethode spielt bei gleichzeitigem Vorhandensein hoher Quantität der Daten eine untergeordnete Rolle, da selbst primitive Lernmethoden bei einer ausreichend hohen Beispielmenge akzeptable Ergebnisse liefern. Allerdings sind in der Praxis vorwiegend kleine bis mittlere Datensätze verfügbar und das Hinzufügen weiterer Trainingsdaten ist kostspielig, weshalb auf die Optimierung der Lernstrategie geachtet werden sollte. Weiterhin könnten nichtrepräsentative Beispiele oder fehlerhafte Erhebungsmethoden das Ergebnis verfälschen. Das bedeutet, dass die zu generalisierenden Fälle richtig aufgenommen und vollständig in die Trainingsdaten eingebaut werden müssen. Qualitativ minderwertige Daten durch Ausreißer oder Lücken in den Merkmalen sind zu vermeiden oder auszubessern. Methoden dazu sind in Kapitel 2.2 zum Feature Engineering beschrieben. (Géron, et al., 2018 S. 23-26)

2.1.4. EVALUIERUNG

Um das entstandene Modell testen zu können, werden verschiedene Datengruppen aus den Trainingsdaten gebildet. 20 % aller Beispiele sind Testdaten für eine abschließende Überprüfung des gelernten Modells. Die restlichen 80 % werden zum Trainieren verwendet. Die Differenz der Funktionswerte der Testdaten zu den Ergebnissen des Trainingsmodells ist der Verallgemeinerungsfehler. Ist dieser im

Vergleich zu dem beim Training aufgetretenen Fehler groß, liegt Overfitting vor. (Wartala, 2018 S. 68)

2.1.5. OVERFITTING

Overfitting tritt durch zu starkes Trainieren der Modelle auf. Das heißt, wenn der Lernalgorithmus sich zu stark auf die Trainingsbeispiele beschränkt und nicht mehr ausreichend generalisiert. Dieser Effekt wird verstärkt, wenn die Daten stark verrauscht sind. Dabei ist es sogar möglich, dass der Lernalgorithmus ein Muster im Rauschen erkennt und dieses fälschlicherweise im Modell mit einprägt. Um Overfitting zu vermeiden, ist es möglich, das Modell zu vereinfachen, mehr Trainingsbeispiele zu lernen oder mittels Feature Engineering die Qualität der Trainingsdaten zu verbessern. Das genaue Gegenteil des Overfittings, nämlich eine zu starke Verallgemeinerung, passiert beim Underfitting. Wenn das Modell zu einfach ist, kann man die Parameteranzahl erhöhen oder bessere Merkmale verwenden. Restriktionen für das gewählte Modell können über einen Hyperparameter im Lernalgorithmus angewendet werden. Die Schwierigkeit dabei ist die optimale Anpassung, um weder Over-, noch Underfitting zu erhalten. Bei linearen Modellen ist die Gefahr des Overfittings geringer. (Géron, et al., 2018 S. 27-29)

Kreuzvalidierung ist eine Methode, ein Modell auf Overfitting zu testen. Dabei wird die gesamte Trainingsmenge in n gleich große Mengen unterteilt und dann der Reihe nach eine Teilmenge zum Testen und die restlichen $n-1$ Teilmengen zum Trainieren verwendet. Der Gesamtfehler wird aus dem Durchschnitt der einzelnen Trainingsläufe gebildet. Im Extremfall, der sogenannten Leave-One-Out-Kreuzvalidierung, wird nur ein einzelner Datensatz anstelle einer Teilmenge zum Testen verwendet. (Görz, et al., 2014 S. 387-388)

2.2. FEATURE ENGINEERING

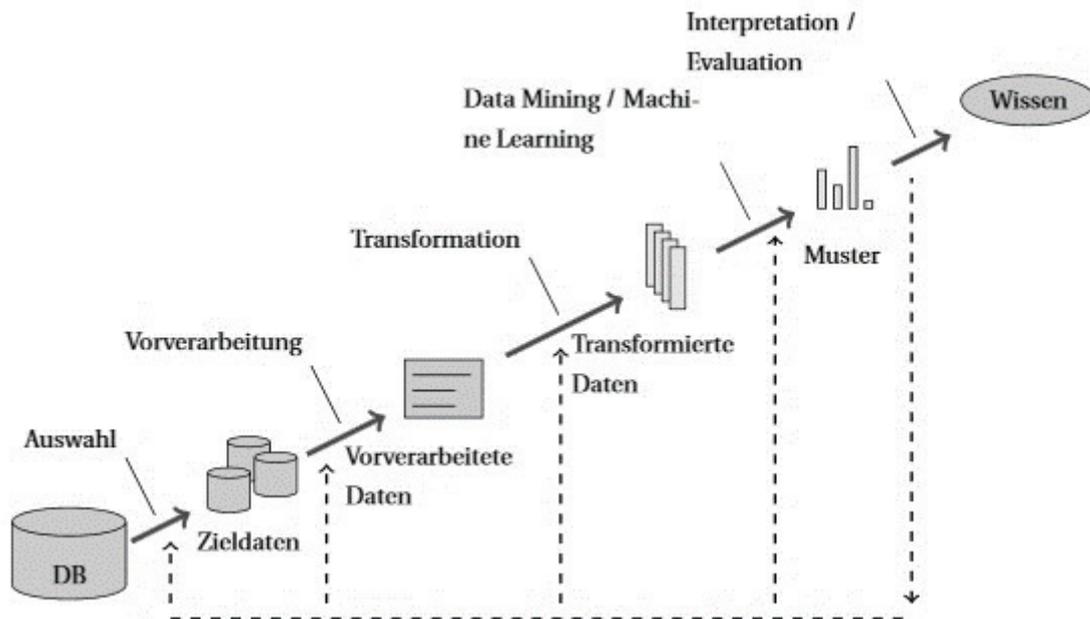


Abbildung 5: Knowledge Discovery in Databases (übersetzt: Wissensfindung in Datenbanken) (Frochte, 2018 S. 16)

Im Feature Engineering geht es darum, die bedeutsamsten Eigenschaften von Objekten für die Verwendung im Machine Learning zu ermitteln. Es sollen unbrauchbare, doppelte oder aussageschwache Eigenschaften außen vor gelassen werden. Die damit einhergehende Reduzierung der Datenbankgröße ermöglicht simultan die Einsparungen der Rechenleistung sowie des Speicherplatzes. Gleichzeitig findet durch die Nutzung von weniger Features eine bessere Generalisierung des Lernmodells statt. Dadurch verringert sich das Risiko des Overfittings. Außerdem ist das entstandene Modell leichter zu interpretieren. All diese Vorteile sind durch Kombination, Vereinfachung oder Transformation einzelner Merkmale möglich. Das Ziel ist es, konzentriertes Wissen in ein Machine Learning Modell einzubinden, in dem die benötigten Features optimal an das maschinelle Lernverfahren angepasst werden, um so eine akkurate und schnelle Voraussage treffen zu können. (Helic, et al., 2018)

Features sind ausgewählte Attribute bestimmter Eigenschaften von Datenobjekten. Diese treten numerisch oder kategorial auf. Numerische Merkmale unterscheiden sich in diskret (beispielsweise Anzahl) und kontinuierlich (beispielsweise Höhe, Zeit). Nominale Eigenschaften sind Kategorien (beispielsweise Geschlecht, Farbe). Wenn diese Kategorien zusätzlich eine bestimmte Rangordnung untereinander besitzen, werden sie ordinale Merkmale genannt. (Kern, et al., 2017)

Beim Feature Processing können drei Methoden zur Anwendung kommen. Durch Binarisierung verändern sich numerische zu booleschen Werten, welche für die Weiterverarbeitung durch spezielle Algorithmen benötigt werden. Mittels Diskretisierung werden kontinuierliche Informationen zu diskreten Werten umgewandelt. Falls eine Skalierung der Daten notwendig ist, kommen Transformation oder Normierung zum Einsatz. (Helic, et al., 2018)

Nach der Vorbereitung aller Features wird die eigentliche Selektion ausgeführt, deren Ziel es ist, redundante und irrelevante Merkmale zu entfernen. Während der Feature Selection wird eine kleine Auswahl an Eigenschaften ausgewählt, um die Verarbeitbarkeit und Qualität der Datenbank zu erhöhen. Um adäquate Feature-Sets zu finden, werden die Methoden der Feature Analysis und Feature Evaluation angewendet, die die Nützlichkeit der einzelnen Eigenschaften bewerten. Dabei kommen suche- oder korrelationsbasierende Algorithmen zum Einsatz. Bei der search-based Feature Selection werden aus der Gesamtheit der Eigenschaften sogenannte Subsets (übersetzt: Untermengen) gebildet und diese auf verschiedene Arten evaluiert. Das zuletzt als am geeignetsten bewertete Subset wird mit einem Stopkriterium verglichen und bei dessen Erfüllung übernommen. (Dong, et al., 2018 S. 193-194)

Es gibt drei verschiedene Arten zur Erzeugung von Untermengen. Mit dem Filteralgorithmus werden einzelne Attribute nur über einen statistischen Wert evaluiert und die besten ausgewählt. Das geht schnell, einfach und ist skalierbar. Leider wird der Einfluss des Lernalgorithmus vollkommen vernachlässigt. Außerdem verfälschen ähnliche Attribute durch ihre äquivalente Bewertung die Subsets. Im Gegensatz dazu wird bei der Wrappermethode der Einfluss des Lernalgorithmus und der einzelnen Parameter untereinander beachtet, wodurch allerdings die Gefahr des Overfittings und hohe Laufzeiten auftreten können. Eine Fusion beider Algorithmen wird als Embedded- oder Hybridansatz bezeichnet. (Dong, et al., 2018 S. 195)

Unter correlation-based Feature Selection versteht man die Bewertung der Eigenschaften in Bezug auf Relevanz und Redundanz, woraus sich vier Gruppen ergeben:

- stark relevant,
- schwach relevant und nicht redundant,
- schwach relevant und redundant
- und irrelevant.

Das optimale Subset kann dann aus den stark relevanten und schwach relevanten, nicht redundanten Attributen zusammengesetzt werden. (Dong, et al., 2018 S. 194-196)

2.3. K-NEAREST-NEIGHBOR

Der k-Nearest-Neighbor-Algorithmus ist ein Lazy-Learner. Dabei wird ein Abfrageobjekt in Bezug zu seinen k-nächsten Nachbarn klassifiziert. Die k-nächsten Punkte werden für einen Mehrheitsentscheid herangezogen. Die am häufigsten auftretende Klasse wird dem gesuchten Abfragepunkt zugeordnet. Damit ist dieser Algorithmus auch sehr anfällig für den Fluch der Dimensionalität, weshalb im Preprocessing unbedingt eine Dimensionsreduktion der Datensätze erfolgen sollte. Das Verfahren ist bei großen Abfragemengen auf einem großen Datenbestand relativ langsam, da bei jeder Klassifizierung der Abstand zu allen anderen Datenpunkten berechnet werden muss und diese dann sortiert werden müssen. Durch ein wenig mehr Aufwand in der Trainingsphase, nämlich durch die Erstellung eines k-dimensionalen, unbalancierten Suchbaumes, erfolgt eine Speicherung einzelner Abstände. Durch diese Art der Organisation der Trainingsdaten ist nun eine schnellere Abfrage möglich. (Frochte, 2018 S. 111-113)

Bei der Regression mit k-nächsten Nachbarn werden die Abstände gewichtet. Dabei werden weiter entfernte Punkte weniger stark gewichtet als nähere. Bei weit verteilten Daten besteht allerdings die Gefahr, durch Rauschen eine falsche Vorhersage zu treffen, da ein naher, verrauschter Punkt die entfernten richtigen Punkte überschatten kann. (Frochte, 2018 S. 114-115)

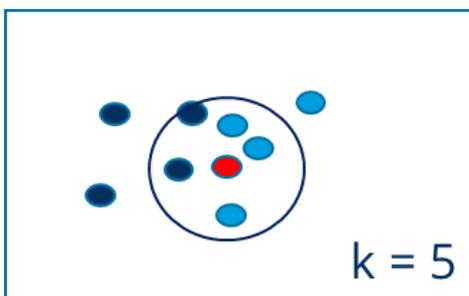


Abbildung 6: Klassifizierung des roten Datenpunktes mit Hilfe der $k = 5$ nächsten Nachbarn (eigene Darstellung)

Die Abstandsmessung kann auf verschiedene Weisen stattfinden. Am bekanntesten ist wohl die euklidische Norm. Dabei wird der Abstand über die Bildung eines rechtwinkligen Dreiecks und den darauf anwendbaren Satz des Pythagoras ermittelt. Ein Beispiel für eine weitere Methode zur Bestimmung der Distanz ist die

Manhattan-Norm, bei der der Abstand nur über horizontale und vertikale Schritte ermittelt wird. Eine Berechnung kann durch die Aufsummierung der einzelnen Vektoreinträge erfolgen. (Ertel, 2016 S. 212-213)

Eine Alternative zur k-Nearest-Neighbor-Klassifikation ist die least-squares Method (übersetzt: Methode der kleinsten Quadrate). Diese ist darauf bedacht, den kleinsten quadratischen Abstand und damit den kleinsten quadratischen Fehler zwischen einer Hyperebene und den zugehörigen Datenpunkten zu ermitteln. Der Abstand wird quadriert, da auf diese Weise große Fehler über 1 stärker einfließen als jene kleiner als 1, welche durch die Multiplikation mit sich selbst sogar noch kleiner werden. Diese Methode funktioniert auch auf nichtlinearen Modellen, solange die Eingabeparameter linear in das Modell eingehen. (Frochte, 2018 S. 105-108)

2.4. SUPPORT VECTOR MACHINES

Support Vector Machines gehören zu den Kernel-Methoden, welche sich der Transformation von Vektorräumen bedienen. Sie sind sowohl linear als auch nichtlinear ausführbar. Die Klassifizierung ist ihre Spitzendisziplin, welche anschließend sehr gut veranschaulicht werden kann. Aber auch Regression und die Ausreißerererkennung können durch ihre Flexibilität durchgeführt werden. Sie sind nicht gut im Skalieren und deshalb eher für weniger umfangreiche Datensets zu empfehlen. (Géron, et al., 2018 S. 145-148)

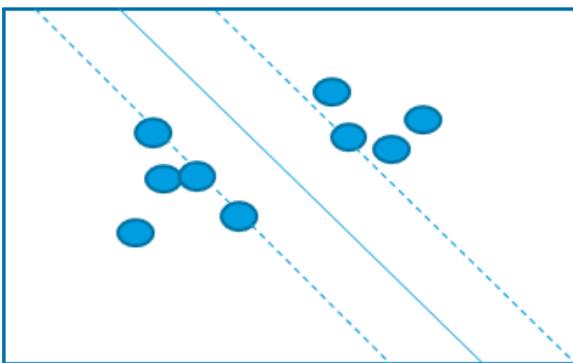


Abbildung 7: Klassifizierung der Datenpunkte durch eine Trennebene aus Unterstützungsvektoren (eigene Darstellung)

Im Wesentlichen geht man bei den Support Vector Machines in zwei Schritten vor. Zuerst findet eine Transformation des Vektorraums in eine höhere Dimension statt, bei welcher die Daten linear trennbar werden. Dies ist immer möglich, solange die einzelnen Datenpunkte nicht widersprüchlich sind. Das heißt, dass sie nicht zu zwei

Klassen gleichzeitig gehören. Danach wird eine Hyperebene gebildet, die so verläuft, dass sie die Klassen mit einem möglichst großen Mindestabstand trennt. Dabei befindet sie sich mittig zwischen den Support Vectors (übersetzt: Stützvektoren), also den äußersten Elementen der jeweiligen Klassen. Durch die meist geringe Anzahl der stützenden Datenpunkte ist keine große Rechenleistung für das Verfahren nötig. (Ertel, 2016 S. 298-299)

Die ausschließliche Verwendung von Randpunkten einer Klasse nennt man Hard-Margin-Ansatz. Sollten die Stützvektoren nicht nur aus den Randdaten bestehen, sondern zusätzliche Datenpunkte im Inneren benutzen, spricht man vom Soft-Margin-Ansatz. Dieser kann auch für nicht exakt trennbare Datensätze genutzt werden. (Géron, et al., 2018 S. 146)

In Abhängigkeit der Kernelart entstehen verschiedene Klassifikationen. Dies ist in der folgenden Abbildung zu erkennen. Die Maschine klassifiziert anders als der Mensch, da beispielsweise die Endpunkte eines Halbkreises wesentlich weiter auseinanderliegen als die Punkte der hineinragenden anderen Klasse. Die RBF (radial basis function) als Kernel kommt in diesem Fall der menschlichen Intuition am nächsten. (Frochte, 2018 S. 300-301)

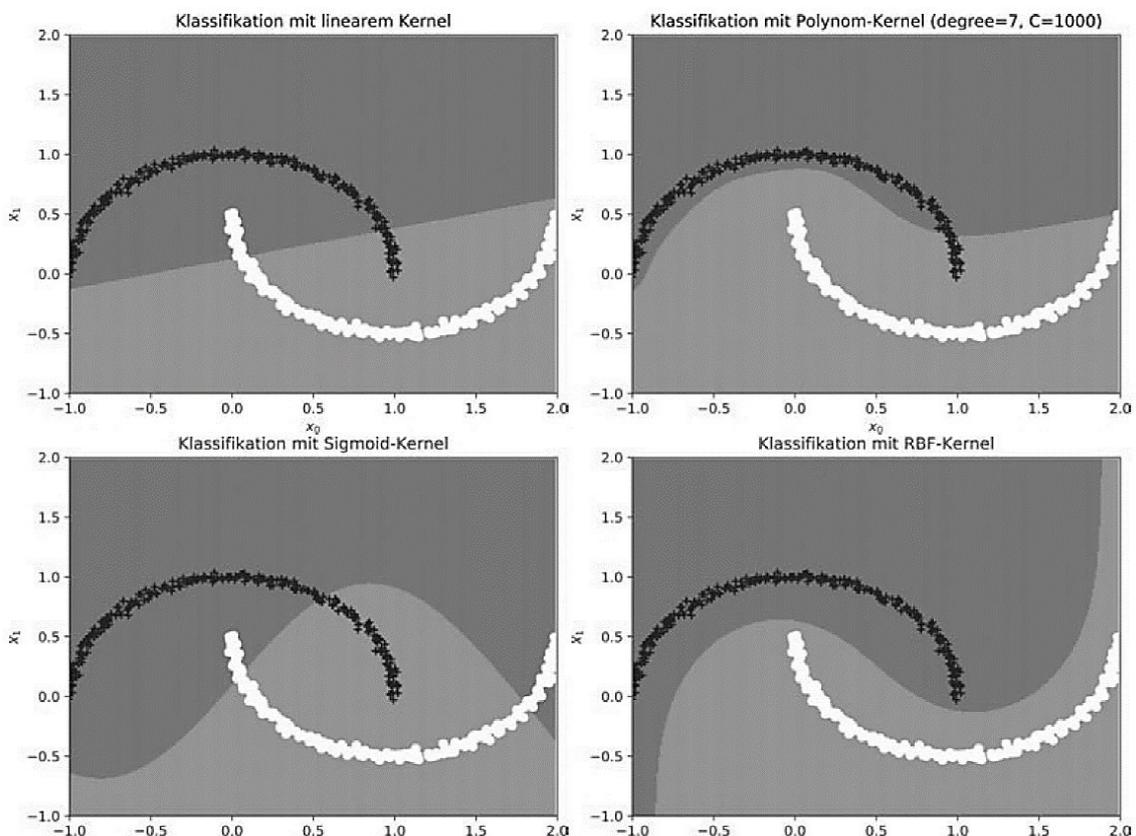


Abbildung 8: SVM mit verschiedenen Kernelarten (Frochte, 2018 S. 300)

2.5. CLUSTERING

Beim Clustering liegen keine Zielwerte (Klassennamen) der Gruppierungen vor, weshalb es zu den unüberwachten Lernverfahren gehört und sich damit von einer Klassifikation unterscheidet. Es wird versucht, Objekte auf eine Weise zu gruppieren, dass deren Ähnlichkeit zueinander und die Unterschiede zu anderen Gruppen maximiert werden. Des Weiteren können die verschiedenen Datengruppen hierarchisch geordnet werden, indem in mehreren Schritten untersucht wird, welche zwei Gruppen sich am meisten ähneln. Diese fasst man dann zusammen, bis am Ende nur noch ein großes Cluster übrigbleibt. Dieses Verfahren nennt man auch agglomeratives Clustering, welches sehr gut in einem Dendrogramm visualisiert werden kann. (Görz, et al., 2014 S. 460)

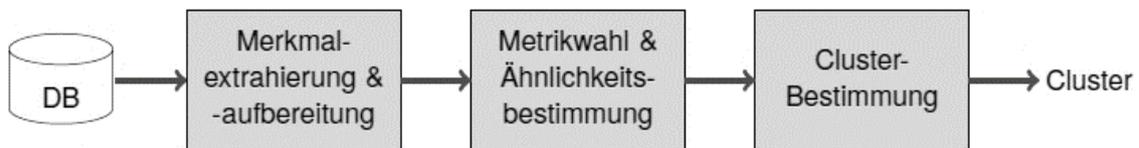


Abbildung 9: Verfahrensweise beim Clustering (Frochte, 2018 S. 304)

Am häufigsten wird dazu das schnell konvergierende k-means-Verfahren verwendet. Hierbei werden anfangs zufällige Cluster gebildet und die dazugehörigen Zentren ermittelt. Danach werden auf Basis des Abstands zwischen den einzelnen Elementen und dem zugehörigen Zentrum die Cluster angepasst und somit auch das Zentrum neu bestimmt. Je nach Wahl der Ausgangscluster können verschiedene Ergebnisse entstehen. Dies kann durch Kreuzvalidierung überprüft werden. Anzumerken ist, dass der Algorithmus auch Ausreißer als eigene Cluster erkennt, weshalb im Voraus unbedingt eine Datenbereinigung stattfinden sollte. (Ertel, 2016 S. 246-247)

2.6. EVOLUTIONÄRE STRATEGIEN

Diese Methode orientiert sich an der evolutionären Entwicklung von Individuen. In jeder Generation pflanzen sich die „fittesten“ Phänotypen einer Population fort. Dabei werden durch Rekombination oder Mutation ihrer Gene Nachkommen mit leicht veränderten Eigenschaften erzeugt. Überleben diese die Selektion, bestehend aus dem Kampf um das Überleben, können sie ihren genetischen Code an die nächste Generation weitergeben und der evolutionäre Kreislauf startet von vorn. Dabei haben Individuen mit qualitativ fitteren Genen besser Chancen sich fortzupflanzen.

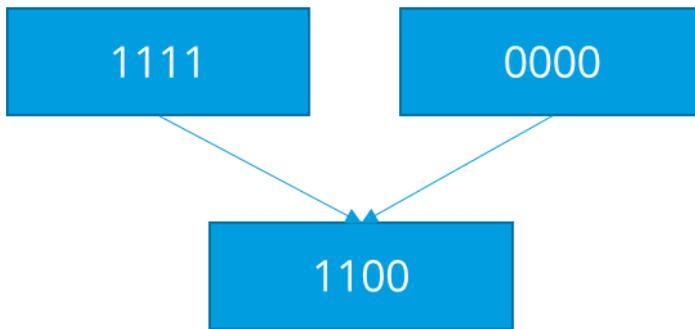


Abbildung 10: Beispielhafte Kreuzung des genetischen Codes zweier Individuen (eigene Darstellung)

Dieses Vorgehen kann auch als Suchalgorithmus verstanden werden, in dem über eine heuristische Methode das globale Extremum des Suchraums angestrebt wird. Die Entdeckung des globalen Optimums ist allerdings nicht zwingend gegeben. Vielmehr ist hierbei die enorme Zeiteinsparung bei simultanem Finden einer annehmbaren Lösung relevant. Phänotypen werden über eine Fitnessfunktion bewertet, wovon abhängt, ob sie zur Erzeugung einer neuen Generation verwendet werden oder nicht. Über Kreuzung und Mutation der genetischen Codes eines einzelnen oder mehrerer Individuen miteinander entstehen neue Genkombinationen. Da bei reiner Kreuzung die Suche in einem lokalen Optimum stagnieren kann, wird Mutation als stochastischer Einflussfaktor verwendet, um die Varianz der Lösungen zu erweitern. Der Gencode muss für den Kreuzungsprozess als Dezimalzahl oder Binärzahl transformiert werden. Aus der neuen Generation werden nun die am schlechtesten bewerteten Individuen selektiert, sodass nur noch die besten Kombinationen im Genpool enthalten bleiben. (Donath, et al., 2012 S. 46)

2.7. REINFORCEMENT LEARNING

Dieses Lernverfahren wird auch bestärkendes Lernen genannt und kommt zum Einsatz, wenn keine hochwertigen Daten zur Verfügung stehen, aus denen das gewünschte Lernziel ablesbar ist. Dabei wird nach dem Trail-and-Error-Prinzip vorgegangen. Es wird versucht, Aktionen mit positiven Ergebnissen zu wiederholen und solche mit einem negativen Ausgang zu vermeiden. Ob eine Wiederholung stattfindet ist von der Höhe des sogenannten Reward (übersetzt: Belohnung) abhängig. Für das Erreichen eines Zieles gibt es eine hohe Belohnung, beispielsweise in Form von Punkten. Beim Scheitern des Systems werden Punkte entzogen. Aber auch für Aktionen, die notwendig sind, um ans Ziel zu gelangen, geht eine geringe Menge an Punkten verloren, um kontinuierliches, sinnloses Handeln zu unterbinden. Das System soll also versuchen, mit möglichst wenigen Aktionen seine

Aufgabe zu erfüllen, damit seine Belohnung möglichst schnell hoch ausfällt. Die diskontierte kumulative Belohnung in Abhängigkeit der Verhaltensregel (engl.: Policy) π wird wie folgt ausgedrückt:

$$V_{dc}^{\pi}(s_t) = \sum_i \gamma^i * r_{t+i}$$

Dabei ist γ der Diskontierungsfaktor, durch den zeitlich entferntere Belohnungen geringer gewichtet werden als nähere. Die Belohnung hängt vom aktuellen Zustand s_t und der Verhaltensregel π ab. Letztere wird optimal, wenn der Reward für alle Zustände maximiert wird. Der wichtigste Ansatz hierfür ist die Markov-Kette. Diese berechnet die Wahrscheinlichkeiten für das Eintreten nachfolgender Zustände. (Görz, et al., 2014 S. 461-462)

Im folgenden Beispiel ist der rote Punkt der Agent, welcher einen optimalen Weg mit dem höchsten Reward finden soll. Um ein sinnloses Hin- und Herbewegen zu vermeiden wurden alle Felder initial mit -1 belegt. Es ist nun herauszufinden, wie man am schnellsten das Feld mit dem Wert 100 erreicht.

-1	100	-1
-1	-1	-1
	-1	-50

Abbildung 11: Einsatzaufgabe für Reinforcement Learning (eigene Darstellung)

2.8. DECISION-TREES

Ein Datenbaum besteht aus einer Wurzel, mehreren Knoten bzw. Verästelungen und Blättern, welche sich am Ende der Äste befinden. An der Wurzel und den Knoten werden Entscheidungen über die Abzweigungen getroffen, welche irgendwann zu einem Blatt führen. Dabei wird der Weg der Entscheidung nachvollziehbar dargestellt. Beim Maschinellen Lernen wird aus vorhandenen Daten ein Baum erstellt, durch den das System zu Klassifikation oder Regression in der Lage ist. Dafür sollten die Bäume annähernd fehlerfrei und klein sein. Für die Größe gibt es verschiedene Messverfahren, wie die Anzahl der Blätter, die weiteste Pfadlänge, die Pfadlängensumme für jedes Blatt oder die gewichtete Pfadlängensumme.

Der CART (Classification and Regression Trees) Algorithmus zur Erstellung von Decision-Trees beschränkt sich nur auf binäre Baumstrukturen und erhält damit mehr Tiefe. Das zur Erstellung des Baumes genutzte Maß ist die Impurity (übersetzt: Unreinheit), welche ein Maß für die Unterscheidbarkeit zweier Klassen ist. Damit kann der Algorithmus die gelernten Datensätze bestmöglich an jedem Knoten verzweigen.

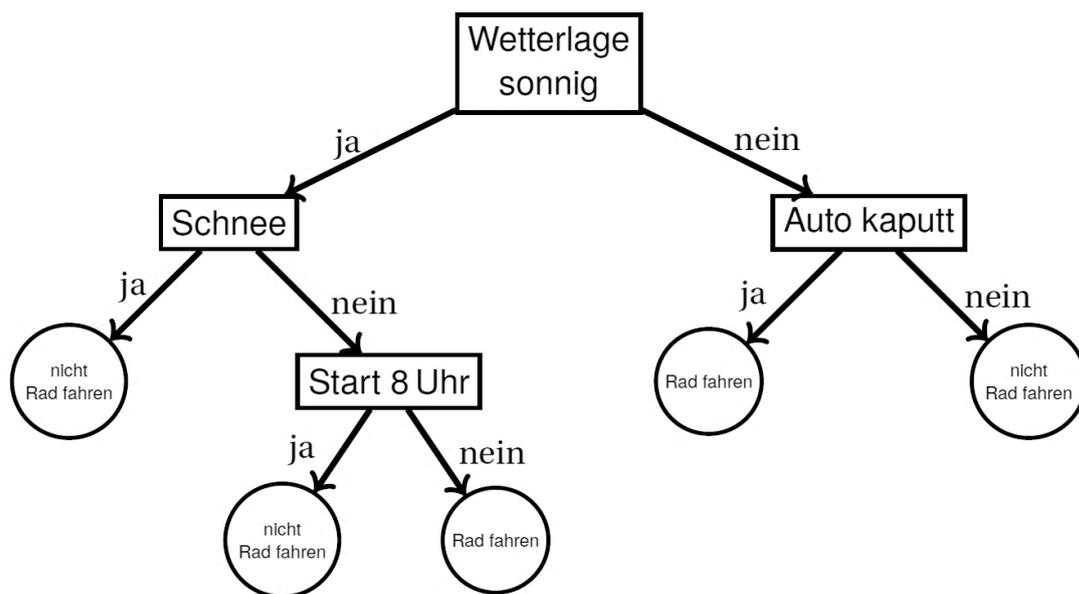


Abbildung 12: Beispiel für einen Entscheidungsbaum (Frochte, 2018 S. 121)

Trainingsdaten mit hohem Rauschen führen zu Overfitting. Es bilden sich unzählige kleine Äste, um alle gelernten Fälle abdecken zu können. Um dem entgegen zu wirken, wird der Baum durch das sogenannte Pruning (übersetzt: Beschneidung) verschnitten. Auf diese Weise werden unbrauchbare Verästelungen reduziert. Beim Pre-Pruning wird schon während der Erstellung des Baumes darauf geachtet, ungenutzte Äste mittels maximaler Tiefen, Mindestverbesserungsraten und der Begrenzung der Anzahl der Blätter wegzulassen. Hingegen werden beim Post-Pruning bereits erstellte Blätter wieder entfernt. Hierzu wird ein Knoten vorläufig durch ein Blatt ersetzt und gemessen, wie sich der Fehler verändert. Sollte eine Verbesserung auftreten, wird die Änderung übernommen. Das entstandene Blatt wird durch Mittelwertbildung (Regression) oder Mehrheitsentscheidung (Klassifikation) ermittelt. (Frochte, 2018 S. 150-152)

2.9. NEURONALE NETZE

2.9.1. GRUNDLAGEN

Ein Neuron besitzt mehrere Eingänge (Dendriten) und nur einen Ausgang (Axon). Dazwischen befindet sich die verarbeitende Einheit, die Synapse. Die einzelnen Eingangssignale sind gewichtet und werden von der Synapse summiert. Überschreitet diese Summe einen bestimmten Schwellenwert, gibt das Neuron ein Signal aus. Je nach Berechnung der Summationsfunktion (z. B. gewichtete Summe über das Produkt der Eingaben, euklidisches Skalarprodukt, Hamming-Distanz etc.) und der Übertragungsfunktion verhält sich das Neuron unterschiedlich. Die Übertragungsfunktion ist meistens sigmoid oder treppenförmig und damit nichtlinear. (Weinberger, et al., 1994 S. 102)

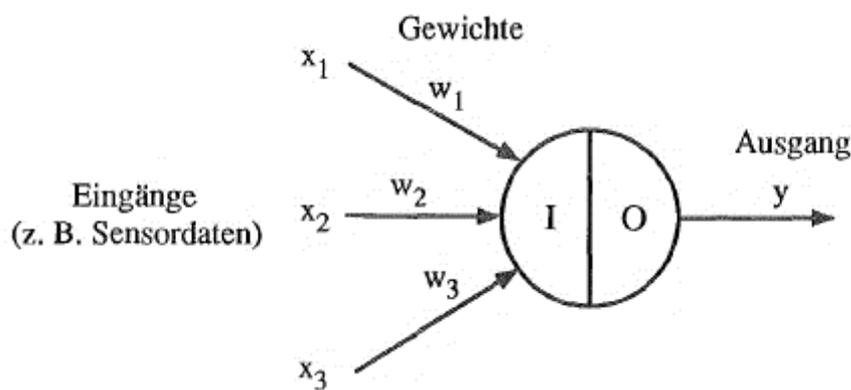


Abbildung 13: künstliches Neuron (Weinberger, et al., 1994 S. 102)

Einzelne Neuronen sind sehr einfach strukturiert und nicht besonders leistungsfähig. Erst der Einsatz einer Vielzahl von ihnen und die Vernetzung untereinander bringen einen größeren Nutzen. Neuronen mit gleichen Aufgaben werden in sogenannten Layern zusammengefasst. Es gibt immer einen Input- und einen Output-Layer. Alle dazwischen befindlichen Schichten heißen Hidden-Layer. Die einzelnen Layer sind in geeigneter Weise miteinander verbunden. Bei Feed-Forward-Netzen bestehen keine Verbindungen zwischen einzelnen Neuronen einer Schicht. Sie sind nur von dem vorgelagerten Layer abhängig und leiten die ankommenden Signale in Richtung Output. Feed-Back-Netze (rekurrente Netze) hingegen erlauben Verbindungen zwischen jedem beliebigen Neuron und lassen damit auch Querverbindungen und Rückkopplungen zu. (Weinberger, et al., 1994 S. 104-105)

In der Lernphase eines neuronalen Netzes werden die Gewichte der einzelnen Verbindungen anhand von Trainingsbeispielen gelernt. Dabei werden die Gewichte Schritt für Schritt angepasst, um das gewünschte Lernziel zu erreichen. Dies kann zur Stärkung einzelner Verbindungen, aber auch zum Abbruch vorhandener Verbindungen führen. Während der Anwendungs- und Funktionsphase kann das Netz für die gelernte Klassifikation oder Regression genutzt werden. Gewichtungen kleiner 0 werden inhibitorisch, größer 0 excitatorisch genannt. Die Wichte der Verbindung vom Neuron i zum Neuron j wird mit w_{ij} bezeichnet. Multipliziert mit dem Ausgabewert x_i der vorherigen Neuronen i und über alle Neuronen im Netz summiert, ergibt sich die Netzaktivität net_j . Durch die Wichtung mit 0 fallen ganze Verbindungen weg. Dies wird bei der Optimal Brain Damage Methode genutzt, da zu große Netze dazu neigen, nur noch auswendig zu lernen. (Boersch, et al., 2007 S. 283)

$$net_j = \sum_i w_{ij} * x_i$$

2.9.2. BACKPROPAGATION

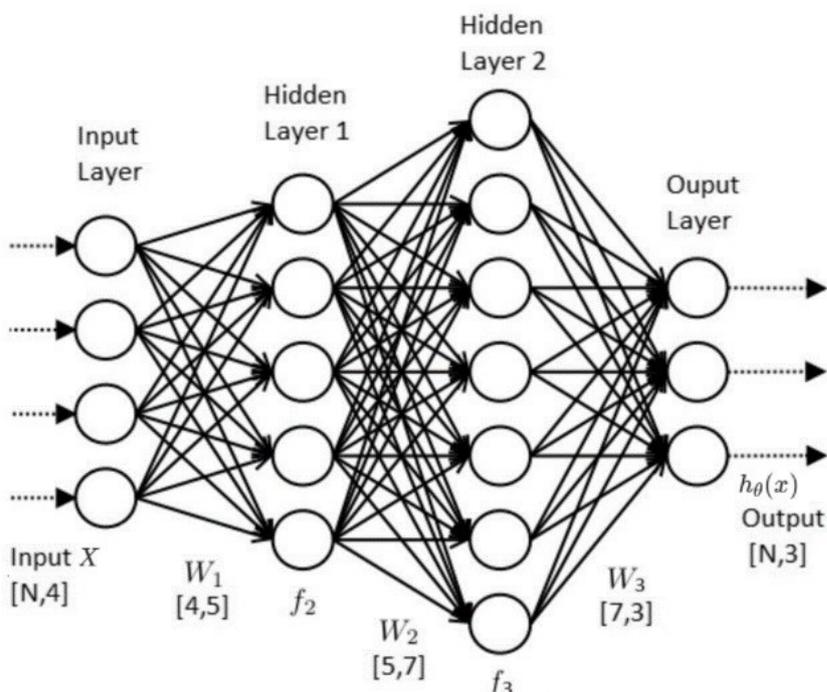


Abbildung 14: Neuronales Netz mit 4 Schichten (Ognjanovski, 2014)

Der Lernalgorithmus der Backpropagation (übersetzt: Fehlerrückführung) kommt bei Feed-Forward-Netzen mit beliebig vielen Schichten zum Einsatz, welche eine

stetige Funktion approximieren. Das Grundprinzip besteht darin, den Fehler über das Gradientenabstiegsverfahren zu minimieren. Dabei wird iterativ der Anstieg der multidimensionalen Fehlerfunktion errechnet, um einen Schritt in die abfallenste Richtung machen zu können. Hierbei spielt die Größe des Schrittes eine wichtige Rolle, da davon die Laufzeit und die Ergebnisfindung abhängen. Bei zu kleinen Schritten dauert die Suche nach dem Minimum sehr lange und man kann in lokalen Minima stecken bleiben. Bei zu großen Schritten wiederum kann das globale Optimum übersprungen werden. Über die sogenannte Lernrate wird die Weite des Schrittes festgelegt. Das Verfahren verläuft wie folgt:

1. zufällige Initialisierung der Gewichte
2. Einlesen des ersten Datensatzes
3. feed-forward-Berechnung der einzelnen Layer
4. Bestimmung des Fehlers zwischen Output des Netzes und Label des Datensatzes
5. schichtweise Rückführung des Fehlers (feed-back)
6. daraus resultierende Anpassung der Gewichte

Diese Prozedur wird mit jedem Trainingsdatensatz wiederholt. (Keller, 2000 S. 176-180)

3. GRUNDLAGEN DES ENTWERFENS

„Die Planung einer Wohneinheit sollte nicht nur die notwendigen Funktionen und daraus resultierende Baukosten berücksichtigen, sondern den grundsätzlichen Anspruch haben, hohe Wohnqualität zu schaffen.“

(Krebs, et al., 2014 S. 9)

Die räumliche Konfiguration dient zur Umsetzung der essentiellen Anforderungen, wie funktionale Zusammenhänge, Erlebnisqualitäten oder Verhältnisse zwischen den Individuen einer Gemeinschaft (Dömer, et al., 2016 S. 48). Wohnen ist ein Grundbedürfnis. Die Abschirmung vor Witterung und der Schutz vor Bedrohungen durch Dritte erfüllen das menschliche Verlangen nach Sicherheit. Es ermöglicht, das Leben in einem sozialen Raum und dessen Individualisierung. In diesem Kapitel sollen die dazu genutzten Leitsätze und normativen Vorschriften dargestellt werden, welche beim Vorentwurf zu beachten sind.

3.1. PARAMETER

Aufgrund der vielfältigen Randbedingungen und Anforderungen an den Wohnraum gibt es eine scheinbar unendliche Vielfalt an Erscheinungsformen. Jeder Betrachter nimmt die Erscheinung eines Entwurfes anders wahr und setzt daher auf unterschiedliche Schwerpunkte bei der Wertung der einzelnen Entwurfparameter. Trotz dessen lassen sich einige Grundregeln für den Entwurf von Wohnungen und Wohngebäuden erkennen.

Beispielsweise erstellte Christopher Alexander eine Sammlung von 253 Mustern für den Entwurf und die Konstruktion von Städten, Vierteln, Außenanlagen, Wohngebäuden, Wohnungen und deren Details in seinem Werk „Eine Muster-Sprache – Städte, Gebäude, Konstruktionen“ (2011). Aus diesem umfangreichen Leitfaden soll der Leser für jedes Teilproblem seines Entwurfes eine Lösung finden, indem er sich mit den zugehörigen Mustern eine eigene angepasste Muster-Sprache erstellt. Dazu wird zuerst ein großes Anfangsmuster ausgewählt, welches dem Gesamtumfang des Projektes am ehesten entspricht. Dieses verweist auf weitere, feinere Muster, welche wiederum auf Detailmuster verweisen. Diese Muster geben ästhetische und architektonische Eigenschaften wieder, die ein zeitloses Wohlbefinden erzeugen. Im Folgenden sollen einige interessante Beispielmuster aufgezeigt werden.

Muster 107: Gebäudeflügel mit Tageslicht

Um das Tageslicht optimal zu nutzen, soll das Gebäude in Flügel mit einer maximalen Breite von 7,5 m aufgeteilt, anstatt als kompakte Masse gebaut zu werden.

Muster 112: Zone vor dem Eingang

Zur Erschaffung von mehr Ruhe im Eingangsbereich und zur Förderung des Bewusstseins für den Wechsel von der öffentlichen Straße zum privaten Haus wird empfohlen eine Änderung des Lichts, der Geräusche, der Bodenoberfläche, der Höhenlage oder der Umschließung vorzunehmen.

Muster 129: Gemeinschaftsbereiche in der Mitte

Zur Belebung der Gemeinschaft soll der Gemeinschaftsbereich im Schwerpunkt der Raumanordnung positioniert werden. Gleichzeitig sollen die Hauptwege tangential daran vorbeiführen, um allen Bewohnern ein soziales Zentrum zu bieten.

Muster 167: Zwei Meter Balkon

Da schmale Freisitze unter 1,8 m Auskragungstiefe nur sehr selten genutzt werden, empfiehlt Alexander, dass Balkone mindestens 2,0 m tief sein und wenn möglich teilweise in das Gebäude zurückgesetzt werden sollen, um eine strikte Trennung zwischen Wohnung und Außenbereich zu vermeiden.

Muster 192: Fenster mit Blick auf die Außenwelt

Da Räume ohne Ausblick wie Gefängnisse auf Menschen wirken, sollen ausreichend Fenster (je nach Region) angeordnet werden, welche einen bestmöglichen Ausblick bieten.

Muster 194: Fenster im Inneren

Auch zwischen zwei Räumen können Fenster eingebaut werden, wenn einer der beiden Räume aufgrund mangelnder Aktivität ungenutzt wirkt oder nicht ausreichend belichtet wird.

Muster 196: Türen in den Ecken

Türen die mittig in einer Wand angelegt werden, sind nur für sehr große Räume oder Eingangszimmer geeignet. Ansonsten sollen

Türen ausschließlich nahe den Ecken eingebaut werden, um keine Bewegungsmuster zu kreieren, die den Raum in verschiedene Teile aufspaltet. Bei zwei Türen sollen diese gegenüber angeordnet werden, um auch in diesem Fall keine Trennung durch den entstehenden Durchgang zu schaffen.

Muster 198: Schränke zwischen Räumen

Schränke sollten immer an Innenwänden positioniert werden. Erstens werden diese so hervorragend für den Schallschutz genutzt. Und zweitens ist diese Lage vorteilhafter für die Belichtung, da die Raumtiefe verkürzt wird und der Schrank neben dem Fenster nur die Ausbreitung des natürlichen Lichtes behindern würde.

Im Gegensatz dazu gibt es einige normative Regelungen, welche insbesondere der Einhaltung von Barrierefreiheit, Brandschutz und Gebrauchstauglichkeit (z.B. Schallschutz, Energieeinsparung) dienen. Sogar im Zusammenhang mit der Erhebung von Steuern gelten bestimmte Anforderungen. Beispielsweise ist nach § 181 Abs. 9 Bewertungsgesetz eine Ansammlung von Räumen erst ab einer Gesamtwohnfläche von 23 m² eine Wohnung. Neben der Einhaltung der jeweiligen Bauordnung des Landes müssen weitere Normen und Richtlinien beachtet werden.

- DIN 18022: Küchen, Bäder und WCs im Wohnungsbau
- DIN 18025-1: Wohnungen für Rollstuhlbenutzer; Planungsgrundlagen
- DIN 18025-2: Barrierefreie Wohnungen; Planungsgrundlagen
- DIN 18030: Norm-Entwurf; Barrierefreies Bauen; Planungsgrundlagen
- ISO/TR 9527: Hochbau; Bedürfnisse von Behinderten in Gebäuden – Leitfaden für den Entwurf

Auf der Grundlage der gezeigten Muster und Normen werden in den nachstehenden Unterkapiteln umfangreiche Entwurfsgrundlagen aufgezeigt, welche bei der Erstellung eines architektonischen Vorentwurfes unbedingt zu beachten sind.

3.1.1. FUNKTIONEN

Nach Krebs (2014 S. 16-17) kann die grobe Einteilung einer Wohnung nach den Nutzungsbereichen erfolgen. Man unterscheidet dabei drei Bereiche. Der Wohnbereich dient zum längeren Verweilen und erfordert daher spezielle Aufenthaltsqualitäten. Gleichzeitig ist er individuell eingerichtet und stark vom

Benutzer geprägt. Die Verkehrsbereiche übernehmen die Verbindungsfunktion der einzelnen Räume und Wohnungen. Sie üben Einfluss auf das Wohngefühl durch Trennung und Hierarchisierung der Teilbereiche aus. Funktionsbereiche zeichnen sich durch eine signifikante Infrastruktur aus und sind nur durch erhebliche bauliche Veränderungen austauschbar. Damit sind sie zwar im Gebrauch festgelegt, gestatten aber erst eine gewisse Unabhängigkeit von außen.

Als Funktionen werden die sechs essentiellen Aktivitäten in einer Wohnung bezeichnet (Heckmann, et al., 2018 S. 56):

- Versammeln
- Arbeiten
- Essen
- Kochen
- Schlafen
- Waschen & Erleichtern

Aber die Wohnung ist nicht nur ein praktischer Ort, an dem der Bewohner isst, schläft und arbeitet. Der repräsentative Aspekt hat gleichermaßen einen entscheidenden Einfluss. Er zeigt, wie ein Mensch lebt, was er als wichtig empfindet und in welcher Umgebung er sich wohlfühlt. Die Sicherheit der eigenen vier Wände gewährt die Selbstentfaltung der Bewohner. Andererseits sind die Bewohner aufgrund ihrer festen Adresse lokalisierbar und damit weniger sicher und versteckt, als ihnen die Wohnung vermittelt. In diesem Sinne sind beispielsweise Obdachlose von mehreren ideellen Verlusten betroffen, da sie neben dem Schutz auch die Selbstrepräsentation nicht umsetzen können. (Eichinger, 2004)

3.1.2. BEWOHNER

Zukünftiger Wohnraum soll sich an den einzelnen Generationen orientieren. Während früher die traditionelle Kernfamilie mit Mutter, Vater und ein bis drei Kindern zeitlich konstant auftrat, gibt es heute verschiedene, schnell wechselnde Familienmodelle, wie Patchwork-Familien oder alleinerziehende Elternteile. Die Grundrisse sollten demzufolge wesentlich flexibler, sogar kombinierbar werden. Die aktuelle Generation der 18- bis 30-jährigen ist geprägt durch längere Ausbildungszeiten, das Voranstellen der Karriere vor die Gründung einer Familie und die höheren Erwartungen an Flexibilität. Dies führt dazu, dass jene Generation mehr auf Wohngemeinschaften und Ein-Raum-Wohnungen fixiert ist. Für Einzelpersonen eignen sich auch sogenannte Mikroapartments, in denen

tageszeitlich wechselnde Nutzungsszenarien durch flexible Möbel möglich werden. Sogar die Auslagerung von Wohnfunktionen wie Kochen oder Waschen in Restaurants oder Waschalons wird angewendet, sodass am Ende eine Symbiose mit dem städtischen Kontext entsteht. So wird unbezahlbarer Wohnraum sukzessive durch Verringerung privater Fläche minimiert. Weiterhin muss in den kommenden Jahren die ins Rentenalter eintretende Generation der Babyboomer mit barrierefreien oder assistierenden Wohnkonzepten bedacht werden. (Heckmann, et al., 2018 S. 37-39)

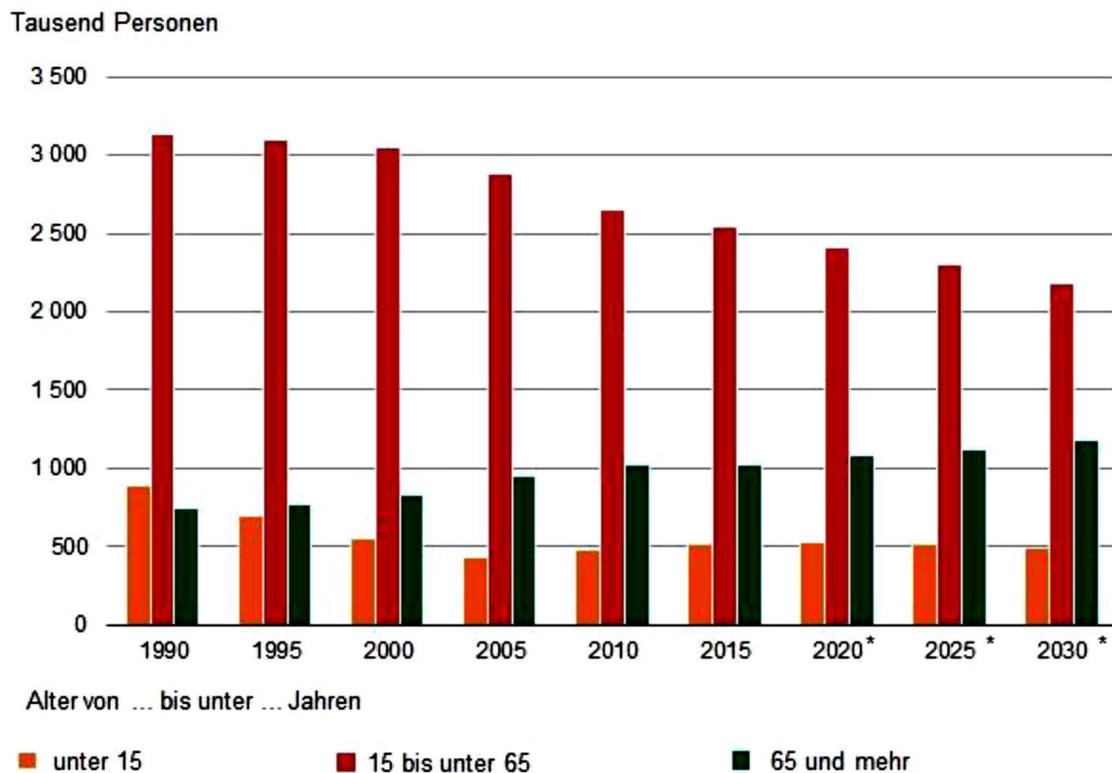


Abbildung 15: Altersdifferenzierte Bevölkerungsentwicklung in Sachsen (SMI, 2019)

3.1.3. GEBÄUDEANORDNUNG

Die Lage ist der größte Kostenfaktor. In ländlichen Gegenden ist ausreichend Bauland verfügbar. Die urbane Zentralisierung der Städte sorgt jedoch für einen Zuzug vieler Menschen. Auf diese Weise entsteht in den Metropolen eine erhöhte Nachfrage für ein Gut, welches nicht ausreichend angeboten werden kann, da es physisch begrenzt ist. Dem gegenüber stehen ländliche Gebiete mit sinkenden Bevölkerungszahlen. Diese Regionen bieten aufgrund einer sehr geringen Nachfrage große Bauflächen zu günstigen Preisen.

In der Nachkriegszeit betrug die Wohnfläche pro Person 16 m². Heute ist sie auf 48 m² gestiegen. Gleichzeitig erhöhte sich der Anteil der Singlehaushalte in den Großstädten auf über 50 %, da Mehrraumwohnungen vermehrt von Einzelpersonen bezogen wurden. (Dömer, et al., 2016 S. 6)

In der Stadtplanung gibt es die sogenannte offene Bauweise mit freistehenden Baukörpern. Orientieren sich diese am Straßennetz und treten mit einer gewissen Regelmäßigkeit auf, wirken sie raumbildend. Bei der Anordnung unabhängig von der vorhandenen Umgebung, zum Beispiel durch einen Versatz der einzelnen Gebäude, entsteht der Eindruck eines fließenden Raums. (Bott, et al., 1996 S. 27)

Zur rückseitigen Erreichbarkeit eines Gebäudes ist nach § 5 Musterbauordnung (MBO) eine Feuerwehrezufahrt notwendig. Die genauen baulichen Anforderungen werden in der Muster-Richtlinie über Flächen für die Feuerwehr (MRFIFw) festgelegt. Beispielsweise muss die Zuwegung eine lichte Breite von 3,0 m und eine lichte Höhe von 3,5 m aufweisen. Hinzu kommt die Tragfähigkeit einer Achslast von 10 Tonnen und einem maximalen Gesamtgewicht von 16 Tonnen. Weiterhin muss nach § 6 MBO die Abstandsfläche eines freistehenden Gebäudes mindestens 3 m betragen. Sie ist abhängig von der Dachform und der Gebäudeklasse und wird in der Regel durch die Formel $0,4 * H$ ermittelt, wobei H für die Gebäudehöhe steht. Überschneidungen von zwei Abstandsflächen sind nur unter besonderen Umständen erlaubt.

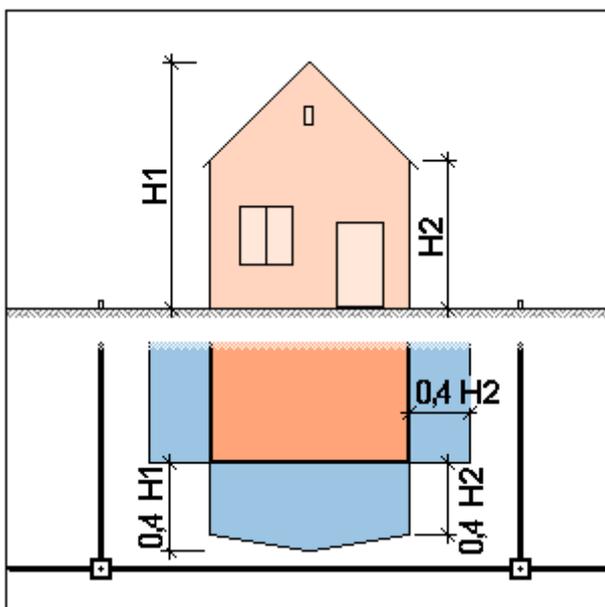


Abbildung 16: Beispielgebäude mit Abstandsflächen (Schröder, 2016)

Ein Quartier benötigt nahegelegene öffentliche Flächen, ein Gebäude gemeinschaftliche Flächen und eine Wohnung private Freiflächen. Öffentliche Flächen sind Straßen, Plätze oder Gassen. Sie machen eine Ansammlung von Häusern erst zu einem Ort zum Wohnen, in dem Begegnungen und Nutzungsüberlagerungen sozialen Kontakt schaffen und identitätsbildend sind. Eine private Freifläche hingegen ist ein Attraktivitätsfaktor des Eigenheims. Sie soll in Form von Gärten, Terrassen, Balkonen, Wintergärten oder Loggien im verdichteten Wohnungsbau Anwendung finden, um Wohnqualität zu sichern. Eine Mischung aus den beiden Typen bildet halböffentliche Flächen. Gemeinschaftliche Höfe, zentrale Erschließungsflächen oder Gemeinschaftsräume sind weder der gesamten Öffentlichkeit zugänglich, noch werden sie ausschließlich privat von einem Mieter genutzt. (Bott, et al., 1996 S. 39-41)

Die Anordnung von Parkplätzen erhöht den kostspieligen Platzverbrauch und ist abhängig von den städtebaulichen Randbedingungen, der Anbindung an den öffentlichen Nahverkehr und von der Alters- und Sozialstruktur. Daraus ergibt sich ein Stellplatzschlüssel, welcher zwischen einem Parkplatz für zwei Wohnungen in der Stadt und zwei Parkplätzen pro Wohnung in ländlicheren Gegenden liegt. Ein zentraler Parkplatz am Siedlungsrand hält die eigentliche Siedlung autofrei, wodurch die Erschließungskosten für die Verkehrsinfrastruktur geringer ausfallen. Für solche Stellflächen ist eine maximale Entfernung zwischen Wohnung und Parkplatz von 100 m und eine gestalterische Eingliederung in das städtebauliche Umfeld maßgebend. Im Gegensatz zu den ausgelagerten Parkflächen besteht die Möglichkeit von Stellplätzen im Straßenraum. Ist nur ein niedriger Stellplatzschlüssel erforderlich, kann durch die Ausweisung von Mischflächen Parkraum am Straßenrand direkt vor den Wohnungen geschaffen werden. Eine weitere Variante ist die Tiefgarage. Nachteilig wirken sich hier die hohen Kosten und die starke Abhängigkeit zu den oberhalb befindlichen Bauwerken aus. Bei Tiefgaragen unter vorhandenen Gebäuden sind während der Planung besonders die statischen Aspekte zu beachten. Vorteilhaft ist, dass der Nutzer trockenen Fußes von seiner Wohnung zu seinem PKW gelangen kann. Die Anordnung der Tiefgarage unter einer ohnehin versiegelten Erschließungsstraße kann durch Überlagerungen der Eigentumsrechte mit der öffentlichen Straße erschwert werden. Zu beachten ist außerdem der Verlauf von Versorgungsleitungen, welche in einem Infrastrukturkanal sinnvoll gebündelt werden können. (Bott, et al., 1996 S. 42-43)

3.1.4. WOHNUNGSGRUNDRISS

Barrierefreiheit:

Aufgrund der verschiedenen Phasen des Lebenszyklus eines Menschen bestehen verschiedene Anforderungen an die dauerhafte Nutzbarkeit einer Wohnung. Der private Lebensraum soll so geschaffen sein, dass Kinder und Erwachsene, egal ob seh- oder gehbehindert, den Wohnraum problemlos nutzen können. Dies erfordert beispielsweise ausreichende Bewegungsflächen, die Vermeidung zu schmaler Durchgänge und nicht benötigter Höhendifferenzen. Zugleich müssen Bedienungsvorrichtungen wie Schalter, Steckdosen sowie Tür und Fenstergriffe in angemessener Höhe angeordnet werden. In der DIN 18025-2 (Barrierefreie Wohnungen und Planungsgrundlagen) sind genaue Vorschriften zur Umsetzung niedergeschrieben.

Erschließung:

„Die Erschließung beschreibt einen Wegraum zwischen dem öffentlichen und dem privaten Bereich, der mit zwei Schwellen beginnt: Die erste bildet den Übergang in das Haus, die zweite führt in die eigene Wohnung.“

(Heckmann, et al., 2018 S. 42)

Die Erschließung hierarchisiert die Räume eines Gebäudes durch deren Abfolge. Statt lediglich ein Verkehrsraum zu sein, erfüllen die Erschließungsflächen soziale Aufgaben. Sie dienen dazu, halböffentliche Begegnungen zu ermöglichen und gleichzeitig einen räumlichen Puffer der individuellen Lebenswelten im Privaten zum anonymen öffentlichen Raum darzustellen. Im Massenwohnungsbau werden Erschließungsräume sehr ökonomisch entworfen, weshalb die räumlichen und sozialen Funktionen auf ein Minimum reduziert werden. (Heckmann, et al., 2018 S. 42)

Die horizontale Erschließung von Wohnungen kann über ebenerdige Zugänge, Laubengänge oder Mittelgänge realisiert werden. Ebenerdige Zugänge werden häufig bei bis zu dreigeschossigen Reihenhäusern genutzt. Dabei sind die Obergeschosse über Treppen erreichbar, welche sowohl innerhalb als auch außerhalb des Gebäudes liegen können. Laubengänge sind halböffentliche Räume und benötigen einen Flächenüberschuss. Sie rentieren sich nur, soweit Flächen in Treppenhäusern eingespart worden sind. Bei Gebäuden mit großer Tiefe bietet sich ein flächensparsamer Mittelgang an. Er eignet sich besonders für Kleinwohnungen

im Hotelprinzip. Durch seine Abgeschlossenheit zur Außenwelt ist er aber eine weniger attraktive Variante der Wohnungerschließung. Die vertikale Erschließung einer Wohnung erfolgt punktförmig über Treppenhäuser. Je nach Anzahl der erschlossenen Wohnungen n pro Geschoss werden diese auch als n -Spänner bezeichnet. Einspänner sind dabei unwirtschaftlicher als Vielfachspänner, welche in Hochhäusern beispielsweise eine Zentralerschließung ermöglichen. Auch Mischformen sind denkbar, bei denen in mehrgeschossigen Objekten kleine Wohnungen über Laubengänge und große Wohnungen direkt an den Treppenhäusern erschlossen werden. (Bott, et al., 1996 S. 47-49)

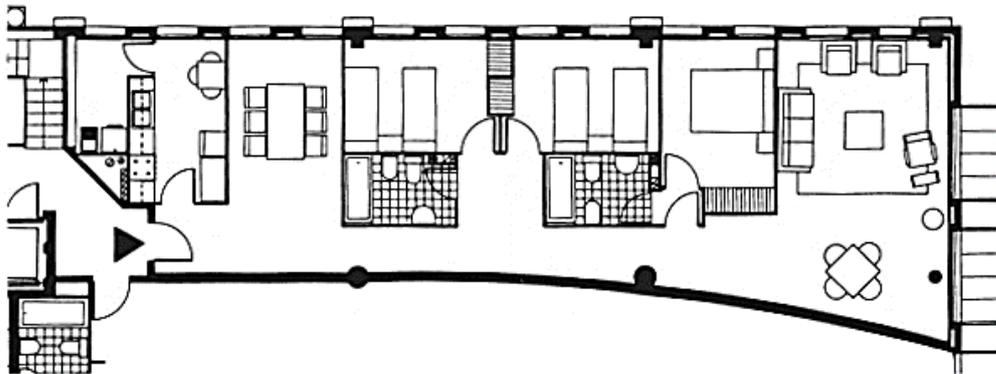


Abbildung 17: Erschließung mit offenem Ende in einem Gemeinschaftsraum (Heckmann, et al., 2018 S. 48)

Orientierung:

Die Belichtung einer Wohnung ist ein entscheidendes Qualitätsmerkmal. Der Sonneneinfall beeinflusst nicht nur das Wohngefühl, sondern auch das energetische Konzept im Hinblick auf den sommerlichen Wärmeschutz und die solaren Wärmegewinne im Winter. Die Orientierung einer Wohnung wird im einfachsten Sinne durch die Lage der Fenster beschrieben, da mithilfe der Transparenz der Scheiben eine Verbindung zur Außenwelt hergestellt werden kann. Sowohl der Lichteinfall als auch die Sichtachse der Bewohner nach außen ist von der Konstellation der Fensterflächen abhängig. Als Faustregel gilt, dass die Fensterfläche ein Achtel der Grundfläche betragen soll und eine maximale Raumtiefe von 5,00 m bei einer Raumhöhe von 2,40 m eingehalten werden muss, um eine ausreichende, direkte und natürliche Belichtung zu ermöglichen. (Neufert, 2019)

Bei freistehenden Gebäuden ist eine allseitige Orientierung anzustreben, um den Wohnraum umfassend in die Umgebung einzugliedern. Eine zweiseitige Orientierung kommt meistens in Zeilengebäuden zur Anwendung. Durch die

gegenüberliegenden Öffnungen der Wohnung treten unterschiedliche Belichtungen und Ausblicke auf die städtebauliche Situation auf. Im Gegensatz zur früher bevorzugten Ausrichtung der wichtigen Räume zur Öffentlichkeit und damit zur Straße, ist heutzutage die Orientierung nach den Himmelsrichtungen maßgebend. Als spezieller Fall treten Innen- und Außeneckenorientierungen auf. Dabei ergibt sich bei Außenecken ein panoramaartiges Sichtfeld, wohingegen man bei Innenecken mehr Tiefe im Grundriss wahrnimmt. Die sehr häufig verwendete einseitige Orientierung tritt bei Aneinanderreihung von Wohnungen mit zentralem Erschließungsweg auf. Tiefe Grundrisse erfordern künstliche Belichtung und mechanische Belüftungsanlagen zur Querlüftung. Die reine Innenorientierung, beispielsweise auf ein Atrium, ermöglicht eine maximale Grundstücksausnutzung. Dabei schützt das Gebäude den Innenhof auch in dichten zentralen Stadtlagen vor Blicken und Lärm. (Bott, et al., 1996 S. 50-51)

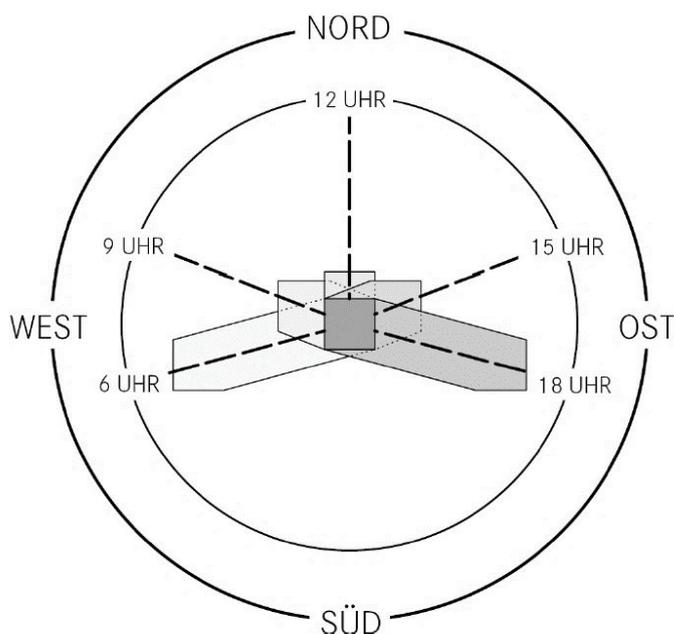


Abbildung 18: Sonneneinfall nach Tageszeit und Himmelsrichtung (Krebs, et al., 2014 S. 16)

Die Lage der jeweiligen Räume ist von der Nutzung selbiger abhängig. Neben- und Erschließungsräume benötigen weder eine spektakuläre Sichtachse und noch eine besondere Belichtung, weshalb diese an der Nordseite angeordnet werden sollten. Eine östliche Orientierung ist für Bereiche zu empfehlen, die morgens mit der aufgehenden Sonne genutzt werden, wie die Küche, das Schlafzimmer oder das Bad. Tagsüber kommt das meiste Licht aus Süden, weshalb Kinder-, Ess- und Wohnbereiche, sowie Terrassen eine südliche bis westliche Orientierung erhalten sollten. (Krebs, et al., 2014 S. 15)

Gliederung:

Zur Gliederung einer Wohnung finden mehrere Unterscheidungsmodelle Anwendung. Das erste Konzept unterscheidet nach der Nutzungsanpassung der einzelnen Räume. Dabei ermöglichen *Neutrale Grundrisse* durch gleichwertige und unabhängig erschlossene Räume Flexibilität. Sie benötigen aber einen gewissen Flächenüberschuss, um in jedem Raum ausreichend Fläche vorzuhalten. Die Mindestgröße ist dabei durch die Abmessungen eines Ehebetts zuzüglich Transferraum und Doppelkleiderschrank gegeben. Es wird also nur die Nutzung der einzelnen Räume durch den jeweiligen Bewohner geändert, anstatt den Grundriss umzuwandeln. In *variablen Wohnungen* können Anpassungen mit Hilfe von Abtrennungen und Einbauten realisiert werden. Einbaumöbel oder Schiebetüren sind zu bevorzugen, da mobile Trennwandsysteme schwer zu handhaben oder bautechnisch unzureichend sind. Es entstehen flexible Grundrisse, welche an kontinuierlich ändernde Nutzungsanforderungen (finanzielle Situation, Arbeit, Wunsch nach Nähe oder Abstand, etc.) der Bewohner angepasst werden können. Auch unter dem Aspekt zunehmender Fluktuation, das heißt dem Wechsel der Wohnung in unterschiedlichen Lebensphasen, sind spezifische Wohnungen mit maßgeschneiderten Grundrissen eine geeignete Lösung. Durch die Partizipation bei der Planung oder den Selbstausbau der Wohnung wird ein *nutzerbestimmter Grundriss* geschaffen. Je nach Lebenssituation erhält der Bewohner eine Wohnung, bei dem der optimale Kompromiss zwischen Wohnkomfort und Nachhaltigkeit gegeben ist. (Bott, et al., 1996 S. 52-53)

Allgemeiner Bereich	↔	Privater Bereich
Arbeitsbereich	↔	Freizeitbereich
Wohnbereich	↔	Funktionsbereich
Verkehrsbereich	↔	Wohnbereich
Erwachsene	↔	Kinder
Bezug zum Außenraum	↔	Bezug zum Innenraum
Vertikale Bezüge	↔	Horizontale Bezüge
Extrovertiert	↔	Introvertiert
Offen	↔	Geschlossen
Laut	↔	Leise
Hell	↔	Dunkel
Tag	↔	Nacht

Abbildung 19: Gegensätzliche Kriterien zur Zonierung einer Wohnung (Krebs, et al., 2014)

Das zweite Konzept beruht auf der Anordnung der Erschließungswege, denn der Wohnungsgrundriss entscheidet über die Art und Weise des Wohnens. Durch die Verbindung und Gruppierung von Räumen, aber auch durch ihre Vereinzelung werden Funktionen, Wege und Sichtachsen bestimmt. Aus der Anordnung eines Flurs folgt eine klare Aufteilung, da die einzelnen Räume abgetrennt und unabhängig erschlossen sind. Auf diese Weise ergibt sich ein neutraler Grundriss. Am besten endet der Flur in einem Gemeinschaftsraum, wodurch ein kontinuierlicher Anstieg der Licht-, Raum- und Blickqualität beim Durchschreiten entsteht. Zu empfehlen sind Einbuchtungen in linear verlaufenden Fluren, denn diese erweitern die Nutzbarkeit, beispielsweise als Garderobe oder Loggia. Im Gegensatz dazu gibt es den flurlosen Grundriss, bei welchem ein Gemeinschaftsraum als klar definierter Mittelpunkt auftritt und als Verteiler zu den umliegenden Räumen dient. Dadurch wird die Fläche des wegfallenden Korridors eingespart und es entsteht ein äußerst kommunikativer Ort in der Wohnung, da nahezu alle Wege durch das zentrale Zimmer führen. (Heckmann, et al., 2018 S. 47)

Ein drittes Konzept ist die Zonierung, also die Trennung der Wohnbereiche. Hier stehen die gemeinschaftlichen Räume, wie Küche, Esszimmer und Wohnraum, den individuellen Zimmern, wie Bad, Schlaf- oder Arbeitszimmer gegenüber. Mit dieser Trennung sollen die einzelnen Funktionen ungestört und parallel ablaufen können. Für jeden Bewohner soll so möglichst viel privater Freiraum entstehen. Die Zonierung erfolgt mit Hilfe von parallelen oder geschachtelten Verteilern bzw. Fluren über welche die einzelnen Bereiche zugänglich sind. Eine Zonierung kann auch in dienende Räume und Wohn- oder Individualräume vorgenommen werden. Die entstehenden Servicezonen aus geräuschintensiven Räumen liegen meist an einem gebündelten Versorgungsschacht und können dann als Pufferbereich zur Erschließungszone der Wohnung angeordnet werden. Eingestellte Elemente in Form eines Installationskerns, eines Wandstücks oder einer Maisonettetreppe in einem großen Raum lassen diesen großzügiger und offener wirken. Gleichzeitig trennen diese Elemente einzelne Funktionsbereiche ab und ermöglichen einen Rundgang, womit sie die Bewegungen der Bewohner führen. Zugleich wirkt die Wohnung vielfältiger, da durch den Rundgang neue Blickwinkel entstehen. Um einen dynamischen Raum zu erhalten, kann ein fließender Grundriss entworfen werden. Dafür sind keine Türen notwendig, sondern ausschließlich eine durchdachte Anordnung weniger Wände. Durch diese wird jede Sichtbeziehung geregelt und die Wohnung in spezifische Orte gegliedert, die dennoch in einander übergehen und einen großzügigen, offenen Raum ohne direkte Verkehrsfläche

schaffen, der Bewohner und Besucher zum Durchschreiten einlädt. (Heckmann, et al., 2018 S. 48)

3.1.5. RAUMENTWURF

In Neufert (2019 S. 313-328) sind die fundamentalen Planungsgrundlagen nach Funktion und Raum zusammengefasst. Für alle Aufenthaltsräume gilt eine lichte Raumhöhe von mindestens 2,40 m und eine minimale Fensterfläche von einem Achtel der Nettozimmerfläche. Außerdem muss eine Belüftung möglich sein. Weiterhin ist zu beachten, dass die Räumgröße und das Verhältnis der Seitenlängen dem Verwendungszweck entsprechend anzupassen sind. Beispielsweise sollten Aufenthaltsräume nicht zu groß sein, da sie sonst leer wirken, und nicht zu langgezogen sein, da die Möblierung so schwerer fällt.

Schlafzimmer:

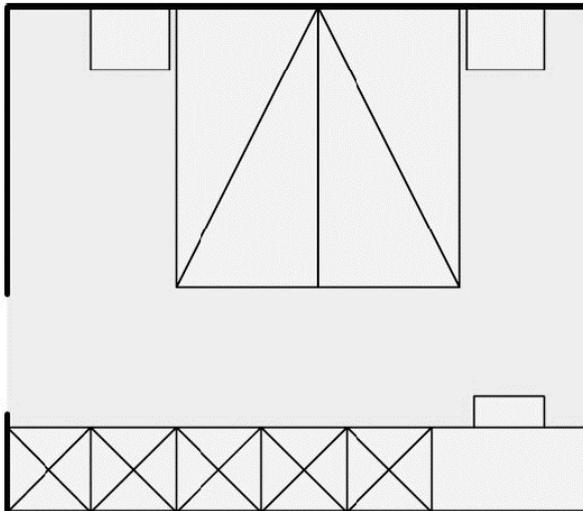


Abbildung 20: Musterschlafzimmer (Krebs, et al., 2014 S. 37)

Um ein natürliches Erwachen zu ermöglichen, sollte das Schlafzimmer mit Blickrichtung zum Sonnenaufgang ausgerichtet sein. Durch die Mindestmöblierung von zwei Kleiderschränken mit je 1,00 m Breite, einem Ehebett mit den Maßen 2,00 m x 2,00 m und je Seite einen Nachttisch mit 0,70 m Breite, welcher gleichzeitig den Bewegungsraum abdeckt, ergibt sich eine Mindestbreite des Schlafzimmers von 3,40 m und eine Mindestfläche von 14 m².

Küche:

Küchen benötigen eine nach Osten orientierte Lage mit kurzen Laufwegen zum Eingang, zur Abstellkammer und eventuell zum Esszimmer oder dem Balkon bzw. der Terrasse. Um einen ungehinderten Arbeitsablauf zu ermöglichen und um gleichzeitig unnötige Wege zu vermeiden, müssen mindestens 1,20 m Bewegungsraum vor den Arbeitsflächen Platz finden. Bei der Annahme einer Tiefe der Arbeitsplatten und Küchenschränke von 60 cm und einer umlaufenden Anordnung dieser resultiert eine Mindestbreite von 2,40 m. Die einzelnen Arbeitsflächen sollten zudem gut ausgeleuchtet werden, beispielsweise durch die Lage am Fenster. Ebenso sollten sie sinnvoll angeordnet werden, beispielsweise durch die benachbarte Position von Abstellflächen neben dem Kochbereich. An die Küche sollte nach Möglichkeit ein Abstellraum angeschlossen sein, welcher mindestens 1,00 m² bei 0,75 m lichter Breite vorweist. Als Faustregel sind 2 % der Wohnungsfläche als Abstellraum anzulegen.

Badezimmer:

Die jeweilige Bauordnung eines Landes gibt vor, dass jede Wohnung ein Bad mit Badewanne oder Dusche und einer Toilette besitzen muss. Bei einer größeren Bewohneranzahl sollte ein zusätzliches WC bedacht werden. Das Badezimmer benötigt keine direkte Sonneneinstrahlung und kann deshalb nördlich orientiert sein. Empfohlen wird eine natürliche Möglichkeit zur Belüftung durch ein Fenster. Funktionell betrachtet ist die Lage nahe der Schlafräume am günstigsten. Aus technischer Sicht sollten sich Bad und Küche eine Wandseite bzw. einen Versorgungsschacht teilen, da so ein geringerer Installationsaufwand entsteht.

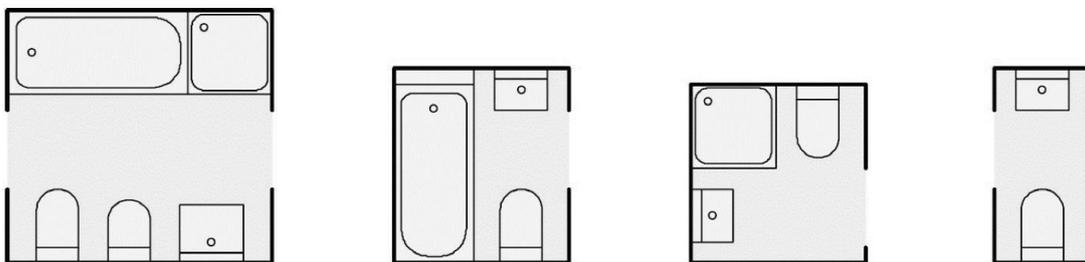


Abbildung 21: Flächenoptimierte Badezimmer bieten oft wenig Aufenthaltsqualität (Krebs, et al., 2014 S. 50)

Wohnbereich:

Der Wohnbereich fungiert als gemeinschaftlicher Mittelpunkt der Wohnung und sollte deshalb an die Hauptverkehrsachsen angebunden sein. Er sollte eine südlich bis westliche Orientierung haben und unabhängig von der Bewohneranzahl mindestens vier Personen Platz bieten, resultierend aus der Bewohnerschaft und eventuellen Gästen.

Erschließungsräume:

In der Regel kommt man mit dem Betreten einer Wohnung zuerst direkt in einen Erschließungsraum. Speziell der Eingangsbereich muss mehrere Funktionen übernehmen. Dazu gehört der Empfang, also die Begrüßung und Verabschiedung der Gäste. Es muss genügend Platz für mindestens zwei Personen und zum Abstellen von Paketen oder Koffern vorhanden sein, sowie zum Abnehmen der Mäntel und deren Verstauung in einer erreichbaren Garderobe. Des Weiteren sollte der Eintrittsbereich eine erste Orientierung ermöglichen und direkten Zugang zum Wohnbereich und zu den wichtigsten Serviceräumen wie Küche und Bad ermöglichen. Der Rest des Verteilerraumes besitzt eine Mindestbreite von 0,9 m bei geringem Verkehrsaufkommen und 1,20 m bei regem Verkehr. Weiterhin sollte er eine möglichst vielfältige Begrenzung besitzen und wie ein eigenes Zimmer eingerichtet sein. Natürliches Licht wäre wünschenswert, ist aber in den meisten Fällen eher zweitrangig.

3.2. ENTWURFSPROZESS

Die Vorgehensweise für den Gesamtentwurf folgt meistens der Erschließung und beginnt, genauer betrachtet, schon bei der Stadt- und Raumplanung. Je nach Art des Vorhabens ist die Lage zur überregionalen Erschließung wichtig. Davon ausgehend wird die lokale Infrastruktur der Umgebung in Abhängigkeit von der Nutzung ermittelt. Gewerbegebiete benötigen beispielsweise einen besseren verkehrstechnischen Anschluss als ruhige Wohngebiete. Die einzelnen städtebaulichen Nutzungsgebiete müssen wiederum entworfen und unterteilt werden. Dabei spielt weiterhin die Verkehrsanbindung, aber auch die Lage zentraler Versorgungselemente eine große Rolle. Von der Anordnung in den einzelnen Vierteln und Straßenzügen geht es dann zum einzelnen Grundstück, auf welchem das jeweilige Gebäude errichtet wird. In diesem folgt dann die richtige Positionierung der Räume, die sich an der internen Erschließung orientiert. Und

selbst in jedem einzelnen Zimmer ist die Einrichtung vom Zugang zu den Nachbarräumen oder der Position der Fenster abhängig. Diese sukzessive detaillierter werdende Kette folgt in jedem Teilabschnitt den Prinzipien „vom Großen ins Kleine“ und gleichzeitig „von der Öffentlichkeit ins Private“. Für den Entwurf dieser Abschnitte gibt es folgende zwei Vorgehensweisen, welche am Beispiel der Raumanordnung im folgenden Absatz dargestellt werden:

„Vor dem Gebrauch“ ist eine morphologische Vorgehensweise, bei der anfangs bewusst auf funktionale Aspekte verzichtet wird. Zur Strukturierung der Baukörpertiefe werden Flächen als Platzhalter für Räume parallel oder senkrecht zur Außenwand auf der Gebäudegrundfläche angeordnet. Nachfolgend sind Tiefe, Breite und Orientierung einzelner Einheiten festzulegen und daraus resultiert dann die Positionierung tageslichtbedürftige Räume. Des Weiteren werden geschlossene (private) und offene (gemeinschaftliche) Bereiche unterschieden. Aus den Varianten ergeben sich verschiedene räumliche Sequenzen, bei denen die Nachbarschaftsbeziehungen der einzelnen Räume eine wichtige Rolle spielen. Im Gegensatz zur gerade beschriebenen Vorgehensweise ist „Vor der Form“ eher eine organisatorische Vorgehensweise in Bezug auf Funktion, Aktivität und Beziehungen der einzelnen Räume. In einem Funktionsdiagramm findet eine Zuordnung der einzelnen Funktionen zu einem oder mehreren Räumen statt, welche somit gruppiert werden. Damit können ohne räumliche Anordnung Rückschlüsse auf die Offenheit, Multifunktionalität und Wichtigkeit für das Gesamtgefüge der Wohnung gezogen werden. In der folgenden Abbildung ist die erste Zuordnung von Funktionen und Räumen in einem Funktionsdiagramm dargestellt. (Heckmann, et al., 2018 S. 54-56)

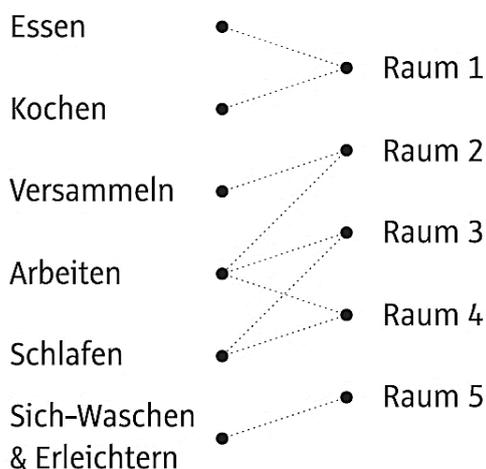


Abbildung 22: Funktionsdiagramm (Heckmann, et al., 2018 S. 56)

4. SOZIALER WOHNUNGSBAU

Immobilien werden heutzutage verstärkt als Anlageobjekte gesehen. Zur Renditemaximierung werden die Mietpreise so hoch wie möglich angesetzt. Durch die Gentrifizierung werden unsanierte, von heterogenen Bevölkerungsschichten besiedelte Wohngebiete zu teuren, nur von Gruppen mit hohem Einkommen bewohnbaren Vierteln. Zusätzlich entziehen private Vermietungsplattformen wie Airbnb den Innenstädten den Wohnraum. All diese Faktoren führen zu einer sozialen Segregation und Wohnungsmangel für die schlechter situierten Bevölkerungsteile. Um diesen Prozess aufzuhalten, gibt es staatliche Maßnahmen, wie den sozialen Wohnungsbau, aber auch private Möglichkeiten günstigen und angemessenen Wohnraum zu schaffen.

Wohnmodelle, die durch soziale, räumliche und technische Innovationen mit geringem Kostenaufwand überdurchschnittlichen Wohnwert schaffen, fallen unter den Begriff „Social Design“ (Dömer, et al., 2016 S. 7). Die folgende Abbildung stellt ein kompaktes und effizientes Gebäude dar, welches mit Hilfe von gestalterischen Elementen an der Außenfassade individualisiert wurde. Dadurch wirkt dieses Studentenwohnheim in Wien nicht monoton, sondern bietet Facettenreichtum und damit verbunden mehr Aufenthaltsqualität bei gleichzeitiger Erhaltung der Nutzungsfreundlichkeit.



Abbildung 23: Studentenwohnheim Wien 1020 (STUWO, 2012)

Sozialer Wohnungsbau darf nicht nur durch kostengünstiges Bauen erfolgen, indem beispielsweise Materialien mit geringer Qualität oder in unzureichender Menge verwendet werden. Daraus ergeben sich durch häufigere Reparaturen und einen erhöhten Energieverbrauch höhere Betriebs- und Instandhaltungskosten und ein geringerer Wohnkomfort. Folglich besteht das größte Problem bei der Schaffung sozialen Wohnraums darin, ein optimiertes Gleichgewicht zwischen Nutzen und Kosten zu finden. Eine Wohnung gilt als erschwinglich, wenn ihre Monatsmiete nicht mehr als 30 % des Nettoeinkommens des Mieters übersteigt. (Dömer, et al., 2016 S. 10)

4.1. STRATEGIEN FÜR ERSCHWINGLICHEN WOHNRAUM

Der Wohnungsmarkt ist durch mehrere Faktoren zu einem beliebten Spekulationsobjekt geworden. Mit der Entstehung von Anlageprodukten im Rahmen der Finanzialisierung, wie beispielsweise Privat Equity Real Estate Fonds oder Real Estate Investment Trusts, wurde es möglich, auch ohne den Kauf einer Immobilie renditegetriebene Anlagestrategien auf dem Wohnungsmarkt zu verfolgen. Da Immobilien in der Bevölkerung als reale, werthaltige und sichere Investitionsmöglichkeiten gegenüber Aktien oder Zertifikaten gesehen werden, ist die Investition in Immobilienfonds zu einem beliebten Konzept der privaten Altersvorsorge erwachsen. Das Immobiliengeschäft hat sowohl wegen der Niedrigzinspolitik der vergangenen Jahre als auch wegen der Volatilität und der Risiken auf dem Aktienmarkt einen zusätzlichen Anreiz für die Kapitalanleger erhalten. Die aktuellen Immobilienpreise sind nicht mehr aus Mieteinnahmen zu erklären, sondern beruhen auf Spekulationen über den Wertzuwachs, sodass teilweise sogar auf die Vermietung verzichtet und der reine Zeitgewinn rentabler wird, als Sanierungs- und Verwaltungskosten zu tragen.

Ein Weg zu erschwinglichem Wohnraum ist es daher, die Wohnungen schon vor der Entstehung vom freien Markt und dessen Prinzipien zu isolieren. Über genossenschaftliche Projekte können die Finanzierung und der Eigenkapitaleintrag auf mehrere Schultern verteilt werden. Durch Partizipation der späteren Bewohner bei der Planung und Errichtung können diese die Art des Zusammenlebens im Gebäude an sich anpassen. Dabei ist eine Konzeptionierung ohne Renditeziele möglich, da nur die Eigennutzung der Wohnungen im Vordergrund steht. Allerdings wird durch eine aufwendigere Entscheidungsfindung der Planungsprozess langwieriger. Durch gemeinsam genutzte Infrastruktur und Wohnflächen im Gebäude können Mieten gesenkt werden. Dieser Bestandteil der Minimalisierung

des Flächenverbrauches wird Mehrfachnutzung genannt. Eine Steigerung dieses Prinzips ist die Externalisierung. Dabei wird beispielweise nicht nur eine in der Wohnung vorhandene Küche von mehreren Bewohnern einer Wohngemeinschaft genutzt, sondern diese sogar aus dem Gebäude ausgelagert und beispielsweise ein Restaurant bevorzugt. (Dömer, et al., 2016 S. 20-21, 50-51)

COLIVING FOR THE COST CONSCIOUS

Say goodbye to hidden living expenses.

	ALTA+ by ollie	Traditional Studio
	\$1,380 - \$2,063	\$2,300*
Gym with Pool	Included	\$50
Social Club & Community Events	Included	\$50
Multifunctional Furniture	Included	\$150
Housekeeping	Included	\$200
Linen & Towel Service	Included	\$20
Bath Amenities	Included	\$20
Premium TV (350 Channels)	Included	\$35
High-speed Wifi (330 Mbps)	Included	\$35
Total Monthly Cost	\$1,380 - \$2,063	\$2,860

*As of March 2018

Abbildung 24: Co-Living-Spaces im Vergleich zu Einzelabnehmern (ollie.co, 2019)

Des Weiteren gibt es heutzutage sogenannte Co-Living- oder Co-Working-Spaces. Diese treten in Form von Hochhäusern in Großstädten auf der ganzen Welt auf und werden zumeist von einem Unternehmen betrieben. Die Bewohner schlafen in Ein- oder Mehrbettzimmern und nutzen Gemeinschaftsküchen, Arbeitsräume und sogar Fitnessanlagen mit allen anderen Mitmietern. Durch die Masse an Bewohnern in einem Hochhaus rentieren sich vergleichsweise sogar möblierte Zimmer mit Zimmer- und Handtuchservice, welcher für Einzelabnehmer wesentlich teuer wäre, wie aus Abbildung 24 ersichtlich wird. (ollie.co, 2019)

Die Hochhausbauweise ist in den teuren, engen Innenstädten die effektivste Art den vorhandenen Platz zu nutzen. Das wird erreicht, indem die Wohnungen so eng wie möglich beisammen liegen. Nach demselben Prinzip bringt die Verdichtung einer Vorstadtsiedlung einen ökonomischen und ökologischen Nutzen durch Flächeneinsparung und fördert gleichzeitig die sozialen Funktionen in Form von Nutzungsvielfalt und Erlebnisdichte. Die Geschossflächenzahl (GFZ) ergibt sich aus der Fläche aller Vollgeschosse geteilt durch die Fläche des Baugrundstückes. Ein Quartier mit 6500 Einwohnern und einer Geschossflächenzahl von 0,8 ist gerade nötig, um eine Schule, zwei Kindergärten und eine Ladengruppe optimal auslasten zu können. Dabei nimmt die Siedlung unter Einbeziehung eines 35-prozentigen Gewerbeanteils eine quadratische Fläche von 115 ha mit einer Seitenlänge von 1072 m ein. Bei zentraler Lage des Siedlungszentrums mit den genannten Einrichtungen wird von keiner Wohnung aus eine maximale Entfernung von rund 700 m überschritten. Gleichzeitig werden durch die dichte Anordnung auch Versorgungsleitungen kurzgehalten und damit Baukosten und Leitungsverluste verringert. Die geschlossener Bauweise bei einer hohen Geschossflächenzahl trägt auch einen kleinen Teil zur Einsparung von Transmissionswärmeverlusten bei. Nachteilig sind der höhere Versiegelungsgrad und die steigende Stellplatzzahl pro Quadratmeter Baugrund. Der steigende Verschattungsgrad tritt auch beim tiefsten Wintersonnenstand erst bei einer Geschossflächenzahl von über 0,8 auf. Bei Nutzung der in Deutschland typischen Geschossflächenzahl 0,4 steigt der Flächenbedarf auf das Doppelte an, die Seitenlänge des Quadrates erhöht sich auf 1516 m und die maximale Entfernung zum Mittelpunkt steigt auf 1061 m. (Bott, et al., 1996 S. 44-46)

4.2. FÖRDERUNGSKRITERIEN

Wie in Kapitel 4.1 beschrieben kann erschwinglicher Wohnraum auf viele Arten und Weisen geschaffen werden. Allerdings bedeutet das noch lange nicht, dass dieser

durch öffentliche Institutionen subventioniert wird. Dafür müssen verschiedene Kriterien der Richtlinie zur sozialen Mietwohnraumförderung realisiert worden sein. 2006 wurden im Zuge der Föderalismusreform die Zuständigkeiten der Finanzierung und Umsetzung vom Bund an die einzelnen Länder übertragen (BMI, 2019). Diese mussten auf Grundlage des Wohnraumförderungsgesetzes (WoFG) eigene Richtlinien unter Anpassung an die regionalen Situationen erstellen, wie in § 5 Abs. 2 WoFG beschrieben wird. Das Land Sachsen erließ die „Richtlinie des Sächsischen Staatsministeriums des Innern zur Förderung der Schaffung von mietpreis- und belegungsgebundenem Mietwohnraum (RL gebundener Mietwohnraum – RL gMW)“. Diese besagt, dass die Neuerrichtung oder Änderung von mietpreis- und belegungsgebundenem Wohnraum in Gebieten mit entsprechendem Bedarf solcher Wohnungen förderungsfähig ist. Der Bedarf besteht bei einer „Gefährdung der Versorgung von einkommensschwachen Haushalten mit ausreichendem Wohnraum zu angemessenen Bedingungen“ (Abschnitt II, Abs. 2a) und ist zu erkennen bei:

- steigender Haushaltsanzahl in den letzten drei Kalenderjahren,
- gleichzeitigem Rückgang der dem Wohnungsmarkt neu zur Verfügung gestellten Wohnungen in den letzten drei Kalenderjahren,
- einer Leerstandsquote von unter 4 %,
- einer 5 % höheren Miete der Gemeinde als die Miete in Sachsen (jeweils auf die Mediane bezogen) und
- einer vergleichsweise höheren Mietbelastung (Mietkosten pro Nettohaushaltseinkommen, bezogen auf die Mediane) zur Durchschnittsmietbelastung in Sachsen.

Für Erstellungs- und Änderungsmaßnahmen müssen mindestens 600 Euro pro Quadratmeter aufgewendet werden. Maximal sind für einen Neubau Kosten in Höhe von 2.200 Euro pro Quadratmeter und für eine Sanierung Kosten in Höhe von 1.800 Euro pro Quadratmeter zulässig. Dabei sind Grundstücks- und Grunderwerbskosten gemäß DIN 276 nicht anrechenbar.

Diese Rahmenbedingungen werden beispielsweise in Dresden durch die „Dresdner Richtlinie zur sozialen Mietwohnraumförderung“ ergänzt. Darin sind unter Abschnitt I Abs. 2 folgende Zielsetzungen notiert:

- die Wohnraumversorgung der nach § 1 Absatz 2 Satz 1 Nr. 1 WoFG vordringlich unterzubringenden Bevölkerungsgruppen sichern und verbessern,

- das Wohnungsangebot nachhaltig erhöhen,
- eine ansprechende architektonische und städtebauliche Qualität aufweisen,
- den Wohnbedürfnissen Alleinerziehender und Familien mit Kindern besonders Rechnung tragen,
- neue Wohnformen ermöglichen, insbesondere Angebote des generationenübergreifenden und seniorengerechten Wohnens,
- barrierefreie und rollstuhlfahrgerechte Wohnungen schaffen,
- die Schaffung und Erhaltung sozial stabiler Bewohner- und ausgewogener Siedlungsstrukturen fördern,
- den Vorrang auf die Innenentwicklung und integrierte Wohnungsbaustandorte legen,
- die Anforderungen des ökologischen Bauens, insbesondere die nachhaltige Reduzierung des Energiebedarfs und der CO₂-Emission berücksichtigen,
- eine ÖPNV-orientierte Verkehrserschließung schaffen, die auf die Reduzierung motorisierten Individualverkehrs ausgerichtet ist,
- einen besonderen Beitrag zum Klima- und Umweltschutz leisten,
- als Bauvorhaben einen besonderen Beitrag zur ökologischen Erneuerung und Entwicklung Dresdens leisten durch:
 - sparsame Flächeninanspruchnahme, durch das Nutzen vorhandener Baurechte (Baulücken, Verdichtungsmaßnahmen),
 - Erschließung neuer Nutzungsmöglichkeiten im Innenbereich von z. B. Brachflächen,
 - wirksame Konzepte des Wasser- und Energiesparens, des Immissionsschutzes, des Naturschutzes sowie der Landschaftspflege.

Genauer werden diese Ziele in der „Anlage 1 zur Dresdner Richtlinie zur sozialen Mietwohnraumförderung“ durch den „Kriterienkatalog zur Priorisierung für den miet- und belegungsgebundenen Mietwohnungsbau in Dresden“ beschrieben. Dabei sollen sich die anfänglichen Bruttokaltmieten an den aktuellen Angemessenheitsrichtwerten der „Leistungen für Unterkunft und Heizung (KdU)“ orientieren. Diese sind beispielhaft für 2019 und 2020 in der folgenden Abbildung dargestellt:

<u>Haushaltsgröße</u>	<u>Bruttokaltmiete</u>	<u>Haushaltsgröße</u>	<u>Bruttokaltmiete</u>
1 Person	378,39 Euro/Monat	4 Personen	603,63 Euro/Monat
2 Personen	444,51 Euro/Monat	5 Personen	775,54 Euro/Monat
3 Personen	518,76 Euro/Monat	jede weitere P.	81,63 Euro/Monat

Abbildung 25: Angemessenheitsrichtwerte Bruttokaltmiete (Landeshauptstadt Dresden, 2019)

Die Anfangsmiete darf erst nach sieben Jahren erstmalig erhöht werden. Die Belegungsbindung besteht für mindestens 15 Jahre. Zur Förderung der Diversifizierung müssen Gebäude sowohl belegungsgebundenen (geförderten), als auch nicht belegungsgebundenen (nicht geförderten) Wohnraum enthalten und dürfen nicht in Bereichen mit überdurchschnittliche häufigen Vorkommen an belegungsgebundenen Wohnungen errichtet werden, um einer möglichen Segregation entgegen zu wirken.

Ein Wohnraum für eine einzelne Person muss eine Mindestfläche von 10 Quadratmetern aufweisen, für zwei oder mehr Personen erhöht sich diese auf 14 Quadratmeter. Dabei dürfen Schlaf- und Kinderzimmer keine Durchgangsräume sein. Freisitze an einer Wohnung sind ausdrücklich erwünscht, soweit diese gestalterisch und funktionell vertretbar sind. Einseitig orientierte Wohnungen dürfen nicht zur Lärm- oder Nordseite zugewandt sein. Jeder Wohnstätte soll über einen genügend großen Abstellraum innerhalb oder außerhalb der eigenen Wände verfügen. Hinter wasserführenden Armaturen und dem Herd sind ausreichend hohe, wasserfeste Wandbeläge anzuordnen. Des Weiteren sollen Wohnungen ab drei Zimmern nach dem Durchwohnprinzip entworfen werden, d. h. von einer Gebäudeseite bis zur gegenüberliegenden. Beim Bewohnen von mehr als vier Personen muss die Unterkunft über ein zweites WC verfügen. Jede fünfte Neubauwohnung soll barrierefrei und jede zwanzigste uneingeschränkt mit einem Rollstuhl nutzbar sein. In diesem Sinne darf beispielsweise das Rohbaumaß jeder Tür nicht weniger als 88,5 Zentimeter betragen und der Zugang muss barrierefrei gestaltet sein.

Auch die maximale Wohnfläche in Abhängigkeit der Bewohneranzahl ist festgelegt. Die Landeshauptstadt Dresden hat dazu folgende Vorgaben und Bedarfsanteile ermittelt. Hier fällt auf, dass Ein- und Zweiraumwohnungen als typische Studentenwohnungen und Unterkünfte ab 4 Personen als Wohnungen für Familien oder Wohngemeinschaften gefragt sind. Dreiraumwohnungen sind momentan nahezu ausreichend vorhanden.

Haushaltsgröße in Personen	1-Personen- haushalt	2-Personen- haushalt	3-Personen- haushalt	4-Personen- haushalt	5-Personen- haushalt	>5-Personen- haushalt
Wohnfläche	35 bis 45 m ²	bis 60 m ²	bis 75 m ²	bis 85 m ²	bis 95 m ²	+ 10 m ²
Zahl der Räume	1 und 2	2 und 3	3	3 und 4	4 und 5	5 und mehr
Bedarfsanteil	40 bis 50%	15 bis 25%	bis 5%	5 bis 15%	5 bis 15%	5 bis 10%

Abbildung 26: Maximale Haushaltsgrößen und deren Bedarfsanteil (Landeshauptstadt Dresden, 2018)

Das Gebäude an sich muss bei mehr als zwei Vollgeschossen hinreichende Abstell- und Gemeinschaftsräume für Kinderwagen, Fahrräder und Wäschetrocknung besitzen. Die Erschließung langer Zeilenbauten ist beidseitig zu realisieren, wobei die Hauseingänge witterungsgeschützt, zur Straße hin orientiert und inklusive Fahrrad- und Gehhilfenabstellplätze auszubilden sind. Die Gebäude sind so zu planen, dass Modernisierungsmaßnahmen wie der Anbau von Fahrstühlen, automatischen Türöffnern und Anlagen zur Gewinnung alternativer Energien im späteren Nutzungsverlauf möglich sind. Der Blockinnenbereich ist als allen Bewohnern barrierefrei zugängliche Grünfläche mit Spielanlagen anzulegen. Das bedeutet, dass die PKW-Stellplätze minimiert werden müssen, um gleichzeitig die Versiegelung zu minimieren und den Straßenverkehr samt Abgasen und Lärm fern zu halten.

5. PROTOTYPISCHE UMSETZUNG

Um den architektonischen Entwurfsprozess digital nachzubilden, müssen Formgestaltungsregeln in Computersprache repräsentiert werden. Dazu soll das Wissen aus den vorherigen Hauptkapiteln genutzt werden, um eine geeignete Nutzung des Machine Learning beim Vorentwurf zu gewährleisten. Das so entstandene Konzept wird anschließend prototypisch in Software umgesetzt und evaluiert.

5.1. KONZEPTION

Um analog zum EVA-Prinzip (Eingabe-Verarbeitung-Ausgabe) die Anforderungen an die Verarbeitung zu definieren, müssen zuerst In- und Output analysiert werden. Als Input sollen die Eingabeparameter der Architekten und Bauherren sowie die Umgebungsbedingungen des Bauplatzes dienen. Diese sind wiederum davon abhängig, was der Verarbeitungsmechanismus für seine Berechnungen benötigt. Der Output soll idealerweise einen kompletten, optimal an die Ausgangsfaktoren angepassten Gebäudeentwurf in Form eines BIM-Modells darstellen. Die Verarbeitung der Inputwerte soll mittels Machine Learning gelöst werden. Die genaue Konzeption des Verarbeitungsprozesses erfolgt unter Einbeziehung einiger Vorüberlegungen.

5.1.1. PROBLEMSTELLUNG

Um das Problem einer entwurfsunterstützenden Generierungssoftware greifbarer zu machen, wurden mehrere Teilprobleme extrahiert, die bei der Erstellung eines vollständigen Gebäudemodells auftreten. In Kapitel 3 sind der Entwurfsprozess und die wesentlichen Einflussparameter dargestellt und können folgendermaßen unterteilt werden.

Um analog zum Vorgehen eines Architekten vom Großen ins Kleine zu gehen, ist das erste Teilproblem die Anordnung der Gebäudegrundfläche auf dem Baugrundstück. Wichtige Einflussparameter sind hierbei:

- Gebäudegrundfläche
- Grundflächenzahl (GRZ)
- Geschossflächenzahl (GFZ)
- Bebauungslinien

- Abstandsflächen und die dafür benötigte Gebäudehöhe
- Verschattungsflächen der Nachbarbebauung
- Anbindung an die Versorgungsinfrastruktur und Verkehrswege
- Benötigte Freiflächen für:
 - Spielplätze
 - Entsorgung
 - Rettungswege
 - Erschließungsflächen
- Orientierung
- Grenzwerte der Gebäudetiefe und -länge

In die entstandene Gebäudegrundfläche können im Folgenden die Wohnungsgrundrisse integriert werden. Ein Zwischenschritt zur Einteilung einzelner Wohnungsflächen ist notwendig, wenn mindestens zwei Wohnungen mit festgelegten Erschließungswegen angelegt werden sollen. Zur Anordnung der einzelnen Räume sind folgende Einflussparameter maßgebend:

- Raumanzahl
- Raumgrößen
- Maximale und minimale Raummaße
- Verhältnis Länge zu Breite
- Orientierung der Fenster
- Fensterflächen
- Lage im Inneren der Wohnung
- Erschließung
- Beziehungen untereinander
- Raumart
 - Verteilerraum
 - Individualraum
 - Serviceraum

Abschließend könnte noch die Einrichtung der einzelnen Zimmer je nach Raumart stattfinden. Dieses ist ein sehr detailliertes Teilproblem, bei dem ästhetische Kriterien den größten Einfluss haben. Im sozialen Wohnungsbau allerdings liegt das Hauptaugenmerk auf der Einpassung der Mindestmöblierung inklusive der zugehörigen Bewegungsflächen in den einzelnen Räumen. Die Umsetzung soll mit dem Ziel einer optimalen und platzsparenden Anordnung verfolgt werden.

Zur weiteren Betrachtung wurde das Problem der Grundrissgenerierung einer Wohnung ausgewählt, um daran prototypisch den Einsatz von Machine Learning in entwurfsunterstützender Software zu demonstrieren.

5.1.2. GRUNDRISSGENERIERUNG

Da der Softwareprototyp für den sozialen Wohnungsbau gedacht ist, können einige Vereinfachungen erfolgen. Aus den Richtlinien zum gebundenen Mietwohnraum Dresden und einer persönlichen Konsultation mit Architekten für soziale Wohnungsbauprojekte der IPROconsult GmbH, konnten wichtige Erkenntnisse gewonnen werden. Unter anderem sei zur Umsetzung eines kostengünstigen und praktischen Gebäudeaufbaus eine rechteckige Geometrie ratsam. Diese ermögliche die effektivste Nutzung der Wohnflächen. Abstraktere Grundrisse gingen mit einem erhöhten Platzverbrauch und geringerer Multifunktionalität einher. Deshalb soll der Softwareprototyp rechteckige Räume in einem rechteckigen Wohnungsgrundriss erstellen. Des Weiteren ist es sinnvoll, wenn die Gliederung im Sinne eines neutralen Grundrisses erfolgt, um die Nutzungsoffenheit der jeweiligen Räume sicherzustellen. Zu Beginn soll deshalb neben der gewünschten Anzahl an Räumen lediglich die Länge und Breite der Wohnungsgrundfläche abgefragt werden.

Die Anordnung von Gebäuden oder Räumen auf einer gegebenen Fläche nach kreativen und funktionellen Maßstäben wird als Layout-Problem bezeichnet. Es besteht ein Unterschied zwischen der Optimierung eines vorgegebenen Grundrisses und dem Entwurf neuer Grundrisslayouts. Bei letzterem fließt Kreativität mit ein, welche durch nicht-operationale Kriterien wie beispielsweise Ästhetik ausgedrückt wird. Durch deren Einfluss kann keine subjektiv richtige Lösung ermittelt werden. Es wird ein flexibler und adaptiver Entwurfsraum benötigt, da der Entwurfsprozess in verschiedenen hierarchischen Maßstäben stattfindet. Knecht (2011) schlägt deshalb neutrale und multifunktionale Kriterien vor, um auf allen Maßstabsebenen eine passende Evaluation der Lösungen durchführen zu können. Hinzu kommt, dass durch das Vorhandensein etlicher Parameter ein multikriterielles Optimierungsproblem entsteht. Deshalb müssen bei der Lösung Kompromisse zwischen den einzelnen Merkmalen gefunden werden. Soll dies eine Maschine übernehmen, ist eine vorherige Wichtung der Parameter erforderlich, was wiederum umfangreiches Vorwissen über deren genauen Einfluss voraussetzt. Alternativ könnte die Auswahl verschiedener Kompromisse abschließend durch einen Menschen erfolgen.

„Kreative Lösungen sind allerdings die Folge schlecht definierter Situationen.“

(Donath, et al., 2012 S. 35)

Um ein möglichst kreatives System zu schaffen, ist es einerseits notwendig, die Entwurfsregeln nicht zu eng zu fassen. Auf diese Weise soll ein möglichst großer Lösungsraum ermöglicht werden. Andererseits ist aus Sicht des Rechenaufwands eine Beschränkung der möglichen Lösungen durch genaue Kriterien ratsam, um so zügig Ergebnisse liefern zu können. Das Ziel besteht also darin, eine Methode zu finden, die das Optimum beider Herangehensweisen liefert. So kann der Lösungsraum beispielsweise nur aus wenigen sehr guten, aber vor allem unterschiedlichen Entwürfen bestehen. Aus diesem Lösungsraum kann dann der beste Entwurf durch die Wichtung einzelner Parameter ausgewählt werden. Daraus resultieren zwangsweise eine geringe Redundanz und eine hohe Varianz der einzelnen Lösungen.

Zur randomisierten Erzeugung der Grundrisse werden in KREMLAS (Donath, et al., 2012) verschiedene Möglichkeiten beleuchtet. Die Methode der dichten Packung versucht, vorgegebene Räume auf einer vordefinierten Grundfläche passend anzuordnen. Dieses Problem tritt beispielsweise bei der optimalen Beladung eines Frachttransporters in der Logistik auf. Dabei können die Raumgrenzen in geringem Maße modifiziert werden, um ein garantiertes Füllen der Grundfläche zu gewährleisten. Zur Ermittlung der dichtesten Packung werden nach einer zufälligen Anordnung die Überlappungsflächen der einzelnen Räume untereinander ermittelt und diese über ein Verschieben der kollidierenden Raumbegrenzungen entgegen der Überschneidung verkleinert. Die Kollision wird hierbei mit Hilfe des Separating-Axis-Theorem überprüft, bei welchem die betrachteten Flächen auf eine Linie projiziert werden. Sollte keine Überschneidung vorliegen, entsteht zwischen den Projektionen ein Zwischenraum.

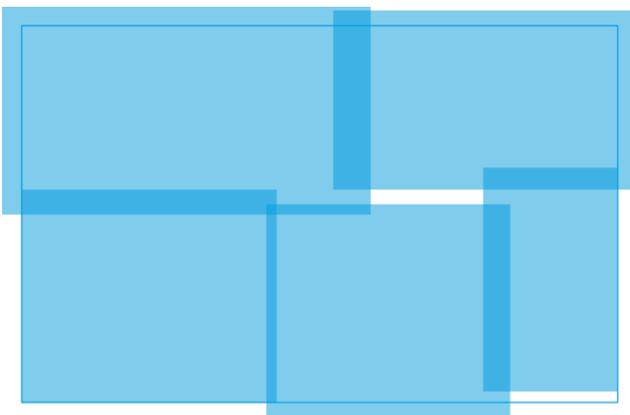


Abbildung 27: Layout mittels dichtester Packung (eigene Darstellung)

Eine zweite Methode zur Erzeugung der Räume kann mittels k-dimensionaler Bäume umgesetzt werden. Dabei stellen Zufallspunkte die späteren Räume dar. Nun wird durch Generierung einer Ebene im Mittelwert zwischen zwei Punkten oder direkt durch einen Punkt eine Wand erstellt. Dieser Prozess wird so oft wiederholt, bis alle Punkte durch Wände getrennt sind oder jeder Punkt eine Wand darstellt. Bei der Nutzung des Mittelwertes entspricht die Anzahl der Punkte der Anzahl der späteren Räume. Hingegen ergibt sich bei der zweiten Variante die Raumanzahl aus der um eins erhöhten Punktzahl.

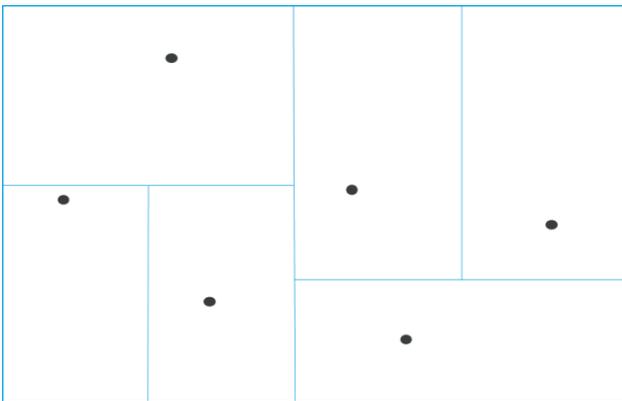


Abbildung 28: Layout mittels k-dimensionaler Bäume (eigene Darstellung)

Mit Hilfe der Voronoi-Diagramme (disjunkte Bereiche nächster Nachbarn) können n -eckige Räume entstehen. Diese werden zuerst durch zufällige Punkte dargestellt. Danach wird der Abstand zu den benachbarten Punkten ermittelt und in der Mitte zwischen dem Ausgangspunkt und dem jeweiligen Nachbarpunkt eine senkrechte Ebene erstellt. In den Schnittpunkten der Ebenen, dem sogenannten Voronoi-Knoten, treffen immer genau drei Trennflächen zusammen. Damit entsteht ein Gebiet, in welchem jeder Punkt näher zu dem Ausgangspunkt des betrachteten Raumes ist als zu einem der anderen Punkte.

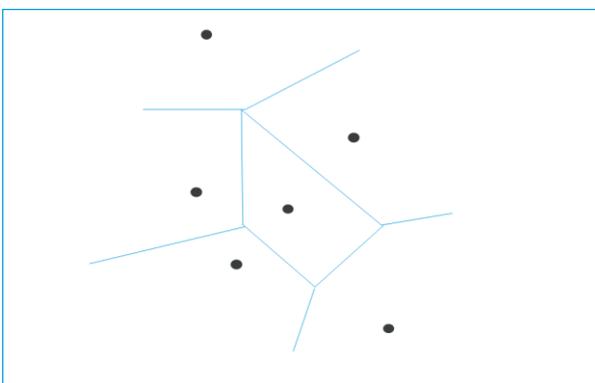


Abbildung 29: Voronoi-Diagramm (eigene Darstellung)

In Abgrenzung zu den gerade genannten Methoden soll diese Arbeit einen eigenen Algorithmus nutzen, welcher besser auf die Anforderungen des sozialen Wohnungsbaus abgestimmt ist. Dabei sind die spitz- und stumpfwinkligen Räume aus Voronoi-Diagrammen unbrauchbar. Dafür sollen die zufälligen Raumpunkte der k-dimensionalen Bäume und die Anpassung der Raumgrenzen aus dem Konzept der dichtesten Packung Einsatz finden. Daraus ergibt sich die folgende Methode zur Erstellung zufälliger Grundrisse:

Je nach Raumanzahl werden entsprechend viele Zufallspunkte erstellt. Die Punkte werden um die Mindestraumbreite in x- und y-Richtung erweitert und auf Kollision überprüft. Bei einer Überschneidung ist ein neuer Datensatz zu bilden. Sollten die Ausgangsräume kollisionsfrei sein, erfolgt eine Ausdehnung der einzelnen Räume in beiden Richtungen bis zum nächsten benachbarten Raum. Sobald damit die Kernzone verdichtet ist, werden die Randbereiche ausgefüllt, indem die Räume bis zur Wohnungsbegrenzung expandiert werden. Am Ende erfolgt ein Vergleich zwischen der Gesamtfläche der Räume und der Wohnungsfläche, um eventuelle Leerstellen zu vermeiden.

5.1.3. WAHL DER MACHINE LEARNING METHODE

Das klassische Konzept des Machine Learning, aus bereits vorhandenen Daten zu lernen, erfordert große digitale Informationsmengen mit homogener Struktur. Diese könnten einerseits durch aufwändige Umformungen bereits bestehender heterogener Daten erstellt werden. Andererseits könnte man durch die Verwendung einschlägigen Fach- bzw. Domänenwissens zufällige, homogene Datensätze erzeugen und aus diesen dann Schlüsse ziehen. Dabei erinnert die erste Methode eher an eine Fleißarbeit. Aufgrund des in den Anfangskapiteln erlernten, umfangreichen Expertenwissens zum Gebäudeentwurf ist es daher sinnvoller, eigene Datensätze zu generieren. Gleichzeitig soll der kreative Aspekt erhalten bleiben. In Anbetracht dieser Überlegungen soll als Machine Learning Methode eine evolutionäre Strategie zum Einsatz kommen. Diese Variante gehört zum Supervised Learning, da ein Bewertungskonzept vom Lehrer vorgegeben werden muss. Gleichzeitig ist es ein Lazy-Learner, da es sich jedes Mal aufs Neue auf die lokalen Parameter neu einstellen muss und damit von Grund auf ein neues Modell erstellt wird.

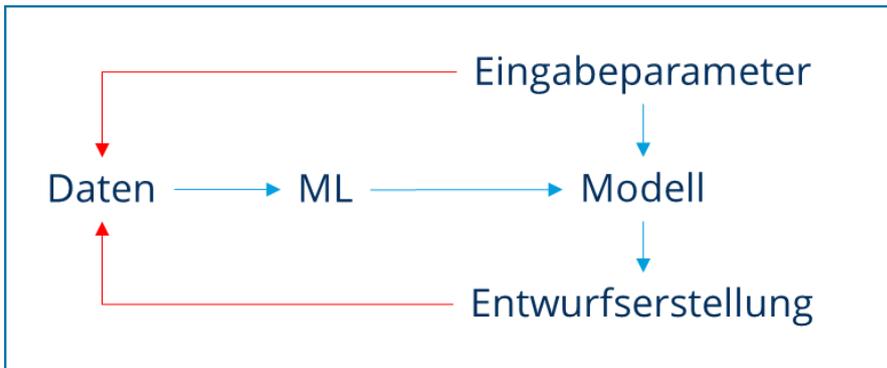


Abbildung 30: Traditionelles Vorgehen (blau), Erzeugung eigener Daten (rot) (eigene Darstellung)

Wie in der vorherigen Grafik verdeutlicht, sollen die zum Lernen benötigten Datensätze der Grundrisse unter Einbeziehung einiger Eingabeparameter randomisiert erzeugt werden. Die zufällig entstandenen Entwürfe müssen dann im Rahmen der Machine Learning Methode (ML) bewertet werden, um die fittesten Individuen für die nachfolgende Generation auszuwählen. Das vorhandene Modell wird so iterativ verbessert und lernt aus den vorherigen Daten.

Das genaue Vorgehen der evolutionären Strategie soll wie folgt aussehen:

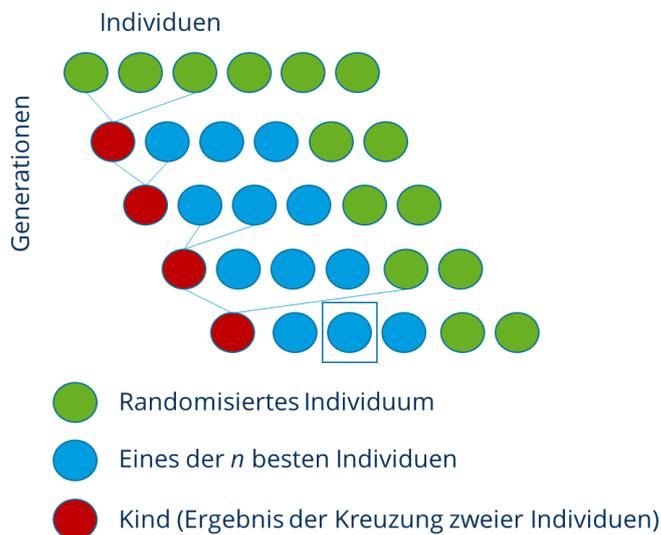


Abbildung 31: Evolutionäre Strategie (eigene Darstellung)

Alle Individuen der ersten Generation werden aus zufälligen Datenpunkten erzeugt. Nach einer Bewertung der Fitness der jeweiligen Grundrisse werden die n fittesten Individuen in die nächste Generation übernommen. Die beiden am besten bewerteten Grundrisse werden durch eine Vermischung ihrer Räume gekreuzt, sodass ein Kind-Individuum entsteht. Dieses wird ebenfalls in die nächste Generation übernommen. Zur Umsetzung der Mutation werden zusätzlich so viele

zufällige Individuen generiert, bis in der zweiten Generation genauso viele Individuen wie in der ersten Generation existieren. Nun findet eine Bewertung der zweiten Generation statt und der Algorithmus beginnt von vorn. Eine exemplarische Darstellung und intensivere Erläuterung folgen im Kapitel 5.2. zur Implementierung.

5.1.4. BEWERTUNGSKRITERIEN

Über die Bewertungskriterien werden im Software-Prototyp einige Einflussparameter quantifiziert. Dazu sind die gewünschten Räume zu erstellen, welche je nach Entwurfsziel über die gegebenen Bewertungsparameter angepasst werden können. Der Benutzer soll außerdem einen Gewichtungsfaktor für jeden Parameter eingeben können, um seine persönlichen Präferenzen einfließen zu lassen.

Die Form des Wohnungsgrundrisses muss vorgegeben werden. Der Benutzer soll nach eigenem Ermessen eine geeignete Form finden und allgemeingültige Kriterien berücksichtigen, bspw. dass zu tiefe Grundflächen nicht ausreichend natürlich belichtet werden können. Genauso muss bei der Parametereingabe das Domänenwissen des Benutzers einfließen, da er unter anderem die minimalen und maximalen Geometrien der einzelnen Räume vorgibt.

Die Bewertungskriterien zur Raumgenerierung sollen hauptsächlich dazu beitragen, die zufällig erzeugten Datensätze in die richtige Richtung zu leiten, damit genügend kreative Freiheit erhalten bleibt. Anhand dieser Kriterien findet die Bewertung für die Kreuzung statt, bei der nur die vielversprechendsten Grundrisse in die nächste Generation weitergegeben werden.

Mindestfläche:

Die minimale Raumfläche gibt die Mindestfläche des betrachteten Raumes in Quadratmetern an. Dieser Parameter ist wichtig, da bestimmte Mindestbewegungsflächen notwendig sind.

Maximalfläche:

Um übergroße Räume zu vermeiden, soll ein Parameter der Maximalfläche eingeführt werden. Die maximale Raumfläche gibt die größtmögliche Fläche des betrachteten Raumes in Quadratmetern an.

Fenster:

Dieses Kriterium ist von der Lage des Raumes in der Wohnung abhängig und kann die Werte 0 (false) und 1 (true) annehmen. Wenn ein Zimmer also an einer Außenwand liegt und damit ein Fenster besitzt, bekommt es bei dem Wert 1 eine positive Bewertung.

Minimallänge:

Um zu dünne Räume zu vermeiden, gibt der Parameter Minimallänge die minimale Seitenlänge der kürzeren Seite eines Zimmers mit einem bestimmten Wert in Metern an.

Maximallänge:

Dieses Bewertungskriterium begrenzt die Länge der größeren Seite eines Raumes auf einen bestimmten Wert in Metern.

Seitenverhältnis:

Um schlauchförmige Räume auszuschließen, gibt dieses Kriterium das maximale Seitenverhältnis größer eins an. Dabei wird die größere Seite auf die kleinere bezogen.

Am Ende des Algorithmus soll ein Grundriss mit der entsprechenden Raumanzahl zurückgegeben werden. Jeder Raum wird erneut nach raumart-spezifischen Kriterien bewertet, um die bestmögliche Nutzung der einzelnen Räume zu ermitteln. In diesem Sinne erfolgt dann eine Zuordnung der Raumtypen, welche aus der Zusammensetzung der besten Bewertungen resultiert. Dabei werden die bisherigen Kriterien durch bspw. die Orientierung eines Zimmers oder Nachbarschaftsbeziehungen erweitert.

Ein weiterer Entwurfsparameter auf Grundlage der benachbarten Räume könnte das Vorhandensein von sogenannten gefangenen Räumen bestimmt werden. Dies ist von der gewünschten Erschließungsform und der einer eventuellen Gliederung der Wohnung abhängig. Ein gefangener Raum ist nicht direkt durch einen Erschließungsraum erreichbar, sondern der Zugang erfolgt durch mindestens einen weiteren Raum.

5.2. IMPLEMENTIERUNG

In diesem Kapitel wird die Implementierung des Teilproblems der Generierung des Wohnungsgrundrisses behandelt. Zuvor war es nötig ein grobes Prozessdiagramm als Entwicklungsstrategie zu erstellen.

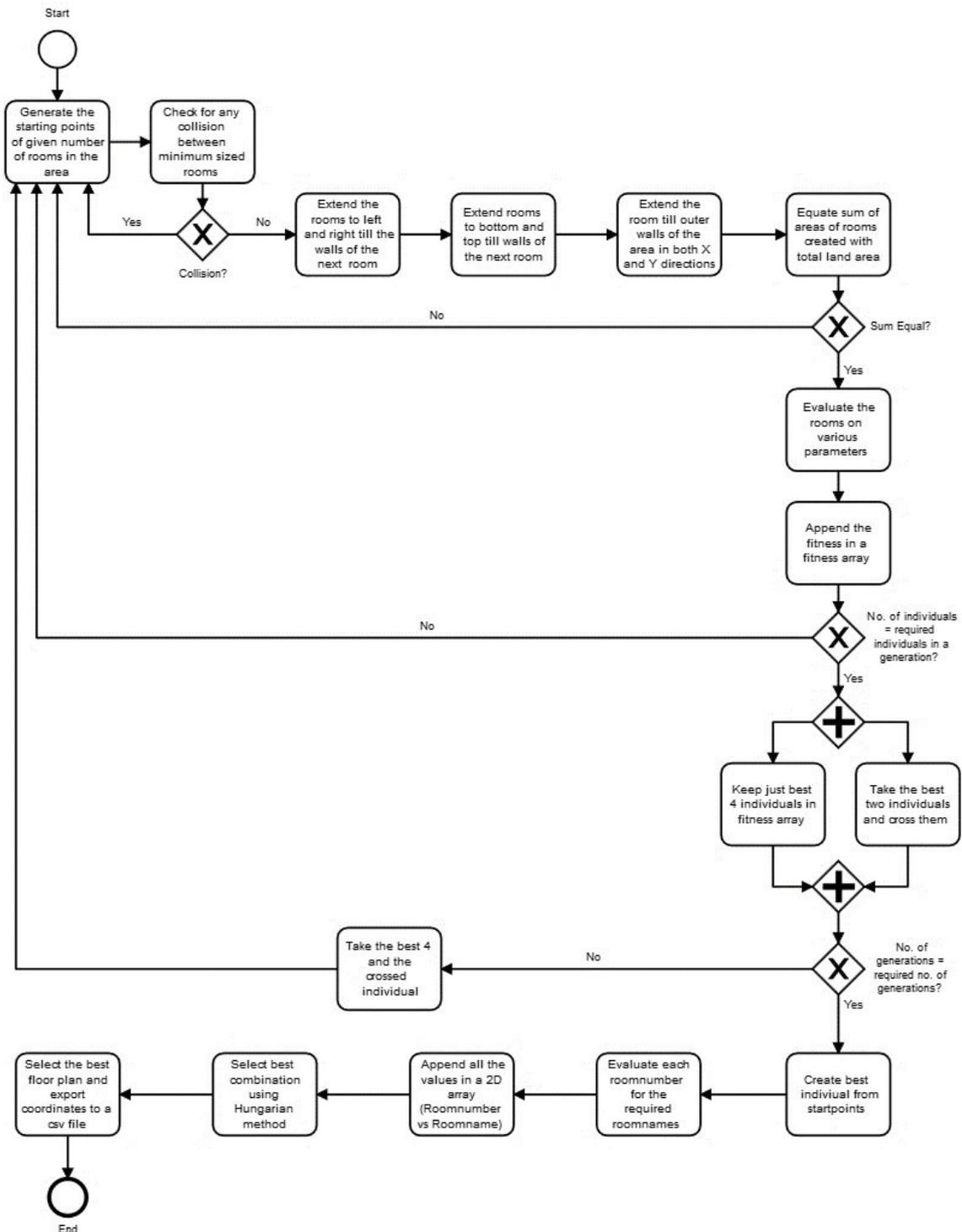


Abbildung 32: Konzeptionierung des Softwareprototypen (eigene Darstellung)

In einem ersten Schritt musste eine Möglichkeit zur Interaktion des Nutzers mit dem Programm gefunden werden. Dazu wurde die Eingabe einiger ausgewählter Einflussparameter mit Hilfe einer übersichtlichen Excel-Tabelle realisiert.

INPUTPARAMETER	width(x):	10	generations:	30						
	length(y):	15	individuals:	1000						
								roomnumber:	7	
	factor	unit	gene	hall	bath	kitchen	bed	working	living	children
minimal area	1	m ²	4	5	8	6	14	14	14	14
maximum area	10	m ²	20	10	15	15	20	20	20	20
window	1	0/1	1	0	1	1	1	1	1	1
minimum length	1	m	1	3	3	3	4	3	5	3
maximum length	10	m	6	5	6	6	5	5	5	5
maximum ratio > 1	200	1+	1,3	2,5	2	2	1,3	1,3	1,5	1,5
orientation	1	NESW	-	-	E	N	E	N	S	W

Abbildung 33: Input-Tabelle für den Benutzer (eigene Darstellung)

Alle veränderbaren Werte sind in roter Schrift eingetragen, nicht zu ändernde Werte in schwarz. Obligatorisch sind die Werte für Breite und Länge der Wohnungsgesamtfläche sowie die Angaben dazu, wie viele Generationen durchlaufen und wie viele Individuen pro Generation gebildet werden sollen. In die Spalte *gene* sind die Parameter für die Grundrissgenerierung einzufügen. Je nach Raumart werden diese für die Zuordnung zu den passenden Räumen in den folgenden Spalten spezifiziert und ergänzt. Um dem Nutzer genügend Spielraum zu lassen, können beliebig viele Räume ergänzt werden. Die Raumanzahl wird durch den Parameter *roomnumber* automatisch aus den gefüllten Spalten und damit aus den gewünschten Räumen ermittelt. Mit Hilfe des Zusatzpakets *xlrd* für Python kann die Input-Tabelle ausgelesen werden.

```

10 # Using excel file for data input
11 loc = ("D:/_DA/Dynamo/PyCharm/venv/Inputmask.xlsx")
12 wb = xlrd.open_workbook(loc)
13 sheet = wb.sheet_by_index(0)
14 roomnumber = int(sheet.cell_value(0, 10))
15 l = int(sheet.cell_value(0, 3))
16 b = int(sheet.cell_value(1, 3))
17 areaSize = (l,b)
18 lr = int(sheet.cell_value(7, 3))
19 br = lr
20 ground_size = l*b
21
22 # Initializing Evolution Parameters
23 generations = int(sheet.cell_value(0, 6))
24 individuals = int(sheet.cell_value(1, 6))

```

Abbildung 34: Datenimport aus einer Excel-Arbeitsmappe (eigene Darstellung)

Zu Beginn der Umsetzung stellte sich das Problem der Darstellung der Räume, welches folgendermaßen gelöst wurde. Die Rechtecke werden durch die x- und y-Koordinate des unteren, linken Eckpunktes und die Größen der Länge und der Breite digital abgebildet. Somit wird ein Raum durch den Datensatz $rooms = [x\text{-Wert}, y\text{-Wert}, Breite(x), Länge(y)]$ dargestellt, wie in Abbildung 35, Zeile 55 zu sehen. Alternativ hätte auch der Mittelpunkt als Referenzpunkt genommen werden können. Darüber hinaus wäre auch eine Darstellung über Vektoren oder die reinen Eckpunkte der Räume denkbar gewesen. Die rechteckige Grundfläche der Wohnung erstreckt sich vom Koordinatenursprung in (0,0) bis zur Wohnungsbreite in x-Richtung ($areaSize[0]$) und Wohnungslänge in y-Richtung ($areaSize[1]$). Des Weiteren muss eine Mindestraumausdehnung in x- und y-Richtung über den Parameter *minimum length* in der Spalte *gene* festgelegt werden.

```

49 rooms = []
50 for i in range(roomnumber):
51     x = random.randint(0, areaSize[0]-lr)
52     y = random.randint(0, areaSize[1]-br)
53     b = lr
54     h = br
55     rooms.append([x, y, b, h])

```

Abbildung 35: Generierung zufälliger Starträume (eigene Darstellung)

In Abhängigkeit von der gewünschten Raumanzahl (*roomnumber*) werden entsprechend viele Zufallspunkte auf der Grundfläche erstellt, abzüglich der Mindestraumgröße (*lr,br*) am oberen und rechten Rand (blau). Die jeweiligen

Punkte (x, y) werden in den positiven Koordinatenrichtungen um die zugehörige Mindestraumausdehnung ($b=lr$, $h=br$) erweitert.

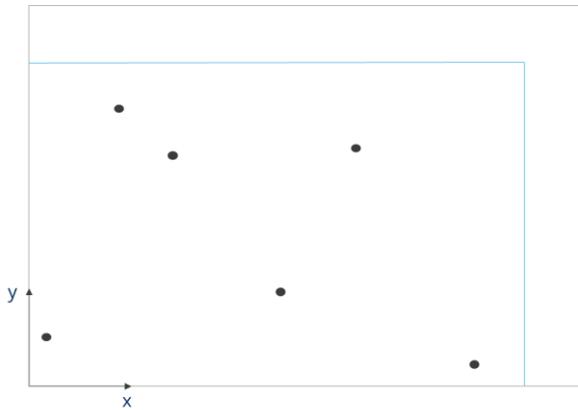


Abbildung 36: Visualisierung der Startpunkte auf der Wohnungsfläche (eigene Darstellung)

Anschließend folgt eine Überprüfung auf Kollision und zwar solange, bis eine kollisionsfreie Lage der minimalen Ausgangsräume gefunden wurde. Bei der Untersuchung der Kombinationen von jeweils zwei Räumen reicht es aus, die obere Dreiecksmatrix ohne Hauptdiagonale zu betrachten, da die Kombinationsmatrix symmetrisch ist. Ansonsten würden die Räume zweifach oder mit sich selbst geprüft werden, woraus unnützer Rechenaufwand entstehen würde.

```

57 # To check for collision between various rooms
58 jk = -1 # Start value for upper triangular matrix
59 for k in range(roomnumber):
60     jk += 1
61     for j in range(jk, roomnumber):
62         if not (j == k):
63             nocollision = not (((rooms[k][0] + rooms[k][2]) > rooms[j][0]) and (
64                 (rooms[k][1] + rooms[k][3]) > rooms[j][1]) and (
65                 (rooms[j][0] + rooms[j][2]) > rooms[k][0]) and (
66                 (rooms[j][1] + rooms[j][3]) > rooms[k][1]))
67             if nocollision == False:
68                 collision += 1
69     n += 1
70     if (collision == 0):
71         startpunkte = copy.deepcopy(rooms)

```

Abbildung 37: Kollisionsalgorithmus in Python (eigene Darstellung)

Zur Kollisionserkennung wird die Lage der Referenzpunkte in einem bestimmten Bereich überprüft. Dieser Ausschnitt ergibt sich aus dem gewählten Mindestraum (rot) und dessen einmaliger Erweiterung in negativer x - und y -Richtung (blau). Ein Beispiel für eine Kollision ist in Abbildung 38 (links) zu sehen. Sollte eine Überschneidung stattfinden, wird der Datensatz verworfen und ein neuer generiert, bis eine kollisionsfreie Anordnung gefunden ist. Für die spätere Kreuzung werden diese Startwerte des Grundrisses als Tiefenkopie unter dem Namen *startpunkte*

gespeichert, da in den folgenden Schritten die Punkte in *rooms* weiterverwendet und damit überschrieben werden.

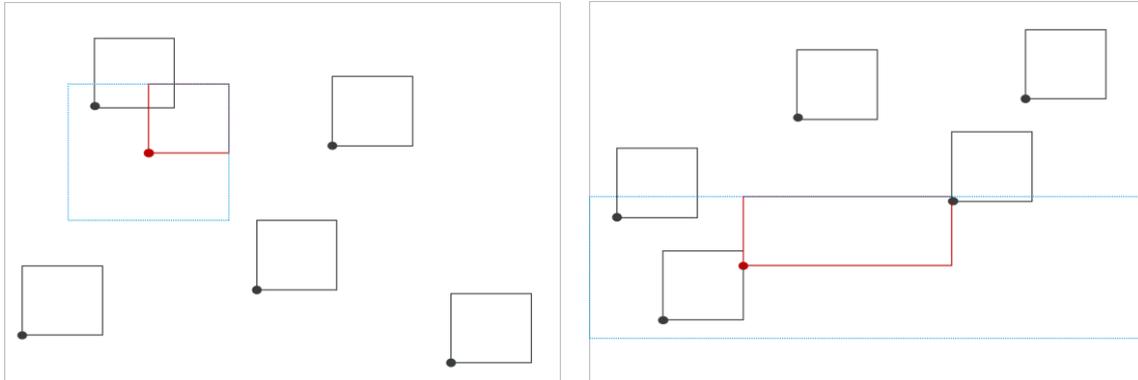


Abbildung 38: Visualisierung der zu überprüfenden Kollisionsbereiche (links) und die erste Ausdehnung (rechts) (eigene Darstellung)

Die Mindesträume werden im nächsten Schritt in x-Richtung ausgedehnt, wie in Abbildung 38 (rechts) dargestellt. Hierfür wird die Distanz zu den betroffenen Nachbarn ermittelt. Dafür ist die Lage der Referenzpunkte in einem bestimmten Bereich (rot) um den betrachteten Raum entscheidend, weshalb diese geprüft werden muss. Anschließend wird die geringste Distanz gewählt und der aktuelle Raum um dieses Maß in die jeweilige Richtung ausgedehnt. Dazu ist eine Unterscheidung in positive und negative Koordinatenrichtung notwendig, um den Ausgangsdatensatz entsprechend zu modifizieren. Das ist deshalb erforderlich, da sich bei einer Expansion nach links der Referenzpunkt in der linken unteren Ecke mit verschiebt.

```

75 # Extending the room till the starting point of next room in X-Direction
76 def Xdirection_extension(rooms):
77     xdistli = numpy.zeros((roomnumber, roomnumber)) - 1
78     xdistre = numpy.zeros((roomnumber, roomnumber)) - 1
79
80     for k in range(roomnumber):
81         for j in range(roomnumber):
82             if j != k:
83                 # checking for distances on the left
84                 if ((rooms[k][0] >= (rooms[j][0] + rooms[j][2])) and (
85                     rooms[j][1] > (rooms[k][1] - rooms[j][3])) and (
86                     rooms[j][1] < (rooms[k][1] + rooms[k][3]))):
87                     xdistli[k][j] = rooms[k][0] - rooms[j][0] - rooms[j][2]
88                 # checking for distances on the right
89                 if ((rooms[j][0] >= (rooms[k][0] + rooms[k][2])) and (
90                     rooms[j][1] > (rooms[k][1] - rooms[j][3])) and (
91                     rooms[j][1] < (rooms[k][1] + rooms[k][3]))):
92                     xdistre[k][j] = rooms[j][0] - rooms[k][0] - rooms[k][2]
93
94                 # Left extension
95                 if (all(i < 0 for i in xdistli[k])):
96                     minxdistli = int(0)
97                 else:
98                     minxdistli = int(min(i for i in xdistli[k] if (i > -1)))
99                     rooms[k][0] -= minxdistli
100                    rooms[k][2] += minxdistli
101                 # Right extension
102                 if (all(i < 0 for i in xdistre[k])):
103                     minxdistre = int(0)
104                 else:
105                     minxdistre = int(min(i for i in xdistre[k] if (i > -1)))
106                     rooms[k][2] += minxdistre
107     return rooms
    
```

Abbildung 39: Algorithmus zum Berechnen des Abstandes zu den nächsten Nachbarn und der Ausdehnung zu diesen (eigene Darstellung)

Nach derselben Vorgehensweise wird auch die y-Richtung angepasst. Damit ergibt sich ein dichtes Konstrukt im Kern der Wohnung, wie in der folgenden Abbildung (links) zu sehen.

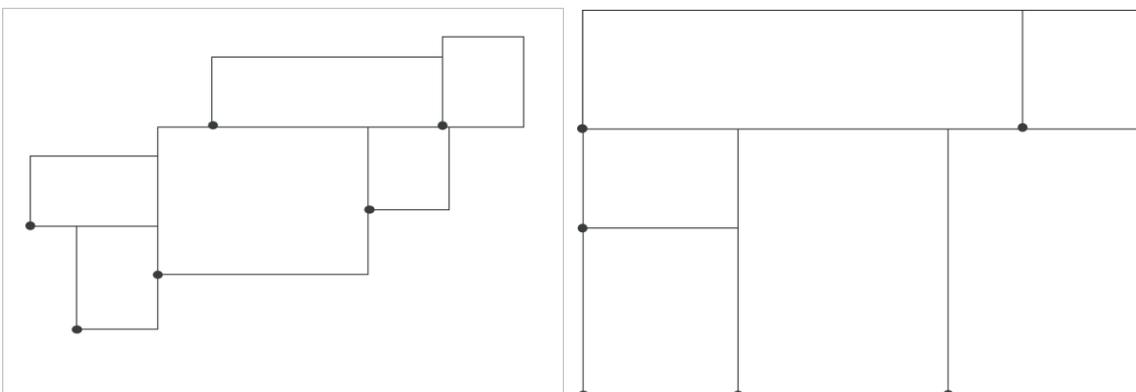


Abbildung 40: Visualisierung des verdichteten Kerns (links) und des Endergebnisses (rechts) (eigene Darstellung)

Nun folgt die Ausdehnung bis zu den Außenwänden. Dabei wird analog nach dem vorherigen Prinzip vorgegangen. Allerdings wird diesmal der Abstand zu den Außenwänden ermittelt und der jeweils geringste Wert zur Expansion genutzt. Es entsteht ein fertiger Grundriss wie in Abbildung 40 (rechts) zu sehen.

```

149 # Extending the rooms till the outer walls in X-Direction
150 def Xdirection_wall_extension(rooms):
151     xdistli = numpy.zeros((roomnumber, roomnumber+1)) - 1
152     xdistre = numpy.zeros((roomnumber, roomnumber+1)) - 1
153     for k in range(roomnumber):
154         for j in range(roomnumber):
155             if j != k:
156                 # Checking for distances on the left
157                 if (((rooms[k][0] >= (rooms[j][0] + rooms[j][2]))) and (
158                     (rooms[j][1] > (rooms[k][1] - rooms[j][3])) and (
159                         rooms[j][1] < (rooms[k][1] + rooms[k][3])))):
160                     xdistli[k][j] = rooms[k][0] - rooms[j][0] - rooms[j][2]
161                 # Checking for distances on the right
162                 if (((rooms[j][0] >= (rooms[k][0] + rooms[k][2]))) and (
163                     (rooms[j][1] > (rooms[k][1] - rooms[j][3])) and (
164                         rooms[j][1] < (rooms[k][1] + rooms[k][3])))):
165                     xdistre[k][j] = rooms[j][0] - rooms[k][0] - rooms[k][2]
166
167     xdistli[k][roomnumber] = rooms[k][0]
168     xdistre[k][roomnumber] = (areaSize[0] - (rooms[k][0] + rooms[k][2]))
169
170     # Adjustment on the left
171     if (all(i < 0 for i in xdistli[k])):
172         minxdistli = int(0)
173     else:
174         minxdistli = int(min(i for i in xdistli[k] if (i > -1)))
175         rooms[k][0] -= minxdistli
176         rooms[k][2] += minxdistli
177
178     # Adjustment on the right
179     if (all(i < 0 for i in xdistre[k])):
180         minxdistre = int(0)
181     else:
182         minxdistre = int(min(i for i in xdistre[k] if (i > -1)))
183         rooms[k][2] += minxdistre
184     return rooms

```

Abbildung 41: Algorithmus zur Ausdehnung der Kernzone zu den Außenwänden (eigene Darstellung)

Wenn alle Räume bis zum Rand ausgedehnt worden sind, folgt eine Überprüfung der Flächen. Durch Lückenbildung kann es vorkommen, dass die Gesamtfläche der Räume in Summe (*sum_ground*) nicht der Wohnungsfläche (*ground_size*) entspricht. In diesem Fall muss der aktuelle Vorschlag verworfen werden und der Prozess von vorn beginnen. Stimmen die Flächen überein, werden die Startpunkte als Gene in einem Array (*gene*) gespeichert (vgl. Zeile 236).

```
228 # Equating sum of areas of the rooms created with the total area of land
229 def Equate_areas(rooms, gene, startpunkte):
230
231     sum_ground = 0
232     for i in range(roomnumber):
233         sum_ground += (rooms[i][2] * rooms[i][3])
234     if ground_size == sum_ground:
235         gueltig = True
236         gene.append(startpunkte)
237     else:
238         gueltig = False
239     return gueltig
```

Abbildung 42: Überprüfungsfunktion auf valide Grundrisse (eigene Darstellung)

Nach der erfolgreichen Erstellung eines Grundrissvorschlags muss dieser im Rahmen der evolutionären Strategie bewertet werden. Für die Entwurfsgenerierung wurden lediglich vier Kriterien gewählt, um ausreichend Spielraum für kreative Ergebnisse zu lassen.

In der Inputtabelle werden die benötigten Einflussparameter abgefragt. Diese gelten grob für alle Räume der zu erstellenden Wohnung, um die Nutzungsoffenheit von Räumen in sozialen Wohnungsbauten zu gewährleisten. Die Raumfläche der erzeugten Räume wird im Abschnitt *# Area* in der Abbildung 43 mit der minimalen und maximalen Fläche aus der Inputtabelle verglichen und bei Unter- bzw. Überschreitung mit einer schlechten (negativen) Bewertung geahndet. Dabei wird bei Nichteinhaltung die Differenz zum Vorgabewert mit einem Faktor multipliziert und ergibt so den Bewertungsbetrag. Der Bewertungsmaßstab erstreckt sich also auf den Bereich negativer Zahlen, wobei das bestmögliche Ergebnis null ist.

Nach demselben Prinzip wird bei der Evaluierung der Länge vorgegangen. Die Mindestlänge der kürzeren Seite wird gleichzeitig als Startgröße bei der Raumgenerierung verwendet.

Zur Bestimmung der Außenwandlage und der damit verbundenen Möglichkeit eines Fensters im Raum werden im Abschnitt *# Window* die Eckpunkte der Räume benötigt. Liegt mindestens ein Punkt auf den Koordinatenachsen bzw. auf den Achsen der Außenwände ist das Kriterium für die Anordnung eines Fensters erfüllt. Als Auswahlmöglichkeit hat der Benutzer nur „viele Fenster“ (*true/1*) oder „wenig Fenster“ (*false/0*). Für jeden Raum, der nicht die gewählte Anforderung einhält, wird der Wert des zugehörigen Wichtungsfaktors subtrahiert.

```

241 # Evaluation the rooms created and give the whole plan a fitness value that can be compared with other
242 def Evaluate(rooms,fitness):
243     evaluation = 0
244
245     # Area
246     for i in range(roomnumber):
247         area = rooms[i][2] * rooms[i][3]
248         if (area < int(sheet.cell_value(4,3))):
249             evaluation -= (int(sheet.cell_value(4,3)) - area) * int(sheet.cell_value(4, 1))
250         elif (area > int(sheet.cell_value(5,3))):
251             evaluation -= (area - int(sheet.cell_value(5, 3))) * int(sheet.cell_value(5, 1))
252
253     # Length
254     for i in range(roomnumber):
255         minlength = min(rooms[i][2],rooms[i][3])
256         maxlength = max(rooms[i][2],rooms[i][3])
257         if (minlength < int(sheet.cell_value(7,3))):
258             evaluation -= (int(sheet.cell_value(7,3)) - minlength) * int(sheet.cell_value(7, 1))
259         elif (maxlength > int(sheet.cell_value(8,3))):
260             evaluation -= (maxlength - int(sheet.cell_value(8, 3))) * int(sheet.cell_value(8, 1))
261
262     # Window
263     for i in range(roomnumber):
264         if (int(sheet.cell_value(6,3)) == 1):
265             if not((rooms[i][0] == 0) or (rooms[i][1] == 0) or (rooms[i][0] + rooms[i][2] == 1)
266                  or (rooms[i][0] + rooms[i][3] == b)):
267                 evaluation -= int((sheet.cell_value(6, 1)))
268         elif (int(sheet.cell_value(6,3)) == 0):
269             if ((rooms[i][0] == 0) or (rooms[i][1] == 0) or (rooms[i][0] + rooms[i][2] == 1)
270                or (rooms[i][0] + rooms[i][3] == b)):
271                 evaluation -= int((sheet.cell_value(6, 1)))
272
273     # Ratio
274     for i in range(roomnumber):
275         ratio = float(max(rooms[i][2],rooms[i][3]))/min(rooms[i][2],rooms[i][3])
276         evaluation -= ((abs(float(sheet.cell_value(9,3))-ratio+1))) ** int(sheet.cell_value(9, 1))
277
278     fitness.append(evaluation)
279     return fitness

```

Abbildung 43: Bewertungsfunktion der Layoutentwürfe (eigene Darstellung)

Für das Verhältnis der Raummaße wird die größere Spannweite durch die kleinere geteilt, um einen Wert größer oder gleich eins zu erhalten. Für die Bewertung wird die Differenz zum Vorgabewert um eins erhöht und dann mit dem Faktorwert potenziert. Damit wird der Einfluss eines größeren Fehlers intensiviert, je weiter dieser vom Idealwert entfernt ist. Dies ist notwendig, da bei der gewählten Methode zur Grundrissgenerierung gehäuft langgezogene Räume auftreten. Dieses Verhalten soll mit einer stärkeren Wichtung unterdrückt werden.

Mit dem jeweiligen Einflussfaktor aus der Inputtabelle können die Kriterien je nach Präferenz angepasst werden, sodass bestimmte Einflussparameter für die Bewertung und damit Erstellung der Layouts mehr oder weniger ins Gewicht fallen.

Wurden in der ersten Generation genügend Individuen erstellt und bewertet, kann die Selektion, Kreuzung und Mutation für die nächste Generation erfolgen. Dazu werden die vier Individuen mit der höchsten Fitness ausgewählt und deren Gene,

also ihre ursprünglichen Startpunkte, als Elternteil-Datensatz gespeichert. Diese werden mit in die nächste Generation übernommen. Die beiden fittesten Elternteile werden gekreuzt und es entsteht ein Kindelement. Dazu werden die Startpunkte der beiden Eltern in einem Array zusammengefasst und aus diesem in Abhängigkeit der Raumanzahl zufällige Einträge gewählt (vgl. Zeile 38). Diese werden im Nachhinein einer Kollisionsprüfung unterzogen. Beim Vorliegen einer Überlagerung der Mindestflächen wird eine neue Auswahl aus dem Array *kind* getroffen.

```
35 while collision > 0:
36     collision = 0
37     if (generation > 0 and individuum == 0):
38         kind = random.sample(eltern1 + eltern2, roomnumber)
39         rooms = copy.deepcopy(kind)
40     elif (generation > 0 and individuum == 1):
41         rooms = eltern1
42     elif (generation > 0 and individuum == 2):
43         rooms = eltern2
44     elif (generation > 0 and individuum == 3):
45         rooms = eltern3
46     elif (generation > 0 and individuum == 4):
47         rooms = eltern4
48     else:
49         rooms = []
50         for i in range(roomnumber):
51             x = random.randint(0, areaSize[0]-lr)
52             y = random.randint(0, areaSize[1]-br)
53             b = lr
54             h = br
55             rooms.append([x, y, b, h])
```

Abbildung 44: Kreuzungsprozess und Übernahme der vier fittesten Individuen (eigene Darstellung)

Der in der letzten Generation als fittestes Individuum auserwählte Grundriss wird erneut anhand der Gene rekonstruiert, da nur die Startpunkte und die Fitnesswerte der Individuen pro Generation gespeichert werden. Das fitteste Element soll nun dargestellt werden. Dazu dient die Methode *Exporting_coordinates(rooms)* mithilfe derer der ursprüngliche Datensatz zur Beschreibung eines Raumes (*[x-Wert, y-Wert, Breite(x), Länge(y)]*) in die Koordinaten der vier Eckpunkte transformiert wird. Dabei werden die Eckpunktkoordinaten ausgehend vom Referenzpunkt im mathematisch positiven Drehsinn in einer Liste in der Form (*[x,y,z],[x,y,z],[x,y,z],[x,y,z]*) gespeichert.

```
281 # Exporting the coordinates in a txt file that can be used by Dynamo
282 def Exporting_coordinates(rooms):
283     pointsdy = []
284     rooms = (rooms)
285     for i in range(roomnumber):
286         pointsdy.append([str(rooms[i][0]) + [str(rooms[i][1])] + ['0'])
287         pointsdy.append([str(rooms[i][0] + rooms[i][2]) + [str(rooms[i][1])] + ['0'])
288         pointsdy.append([str(rooms[i][0] + rooms[i][2]) + [str(rooms[i][1] + rooms[i][3])] + ['0'])
289         pointsdy.append([str(rooms[i][0]) + [str(rooms[i][1] + rooms[i][3])] + ['0'])
290
291     with open('coordinates.txt', mode='w') as coord:
292         writer = csv.writer(coord)
293         writer.writerows(pointsdy)
294     return
```

Abbildung 45: Funktion zur Transformation und zum Export der Koordinaten (eigene Darstellung)

Da der Prototyp ein BIM-Modell ausgeben soll, stellte sich die Frage, wie der in Python berechnete Wohnungsgrundriss in die zur BIM-Modellierung genutzte Software Autodesk Revit übergeben werden könnte. Hierfür bietet sich ein Algorithmus in einem Autodesk Dynamo Skript an. Das Python-Skript exportiert die Koordinaten der Eckpunkte der jeweiligen Räume in eine TXT-Datei. Diese kann über Dynamo eingelesen werden. Nach einer passenden Formatierung in Unterlisten werden aus den einzelnen Teillisten die Koordinaten der Raumecken im Revit-Projekt erstellt und durch den gewählten Wandtyp verbunden.

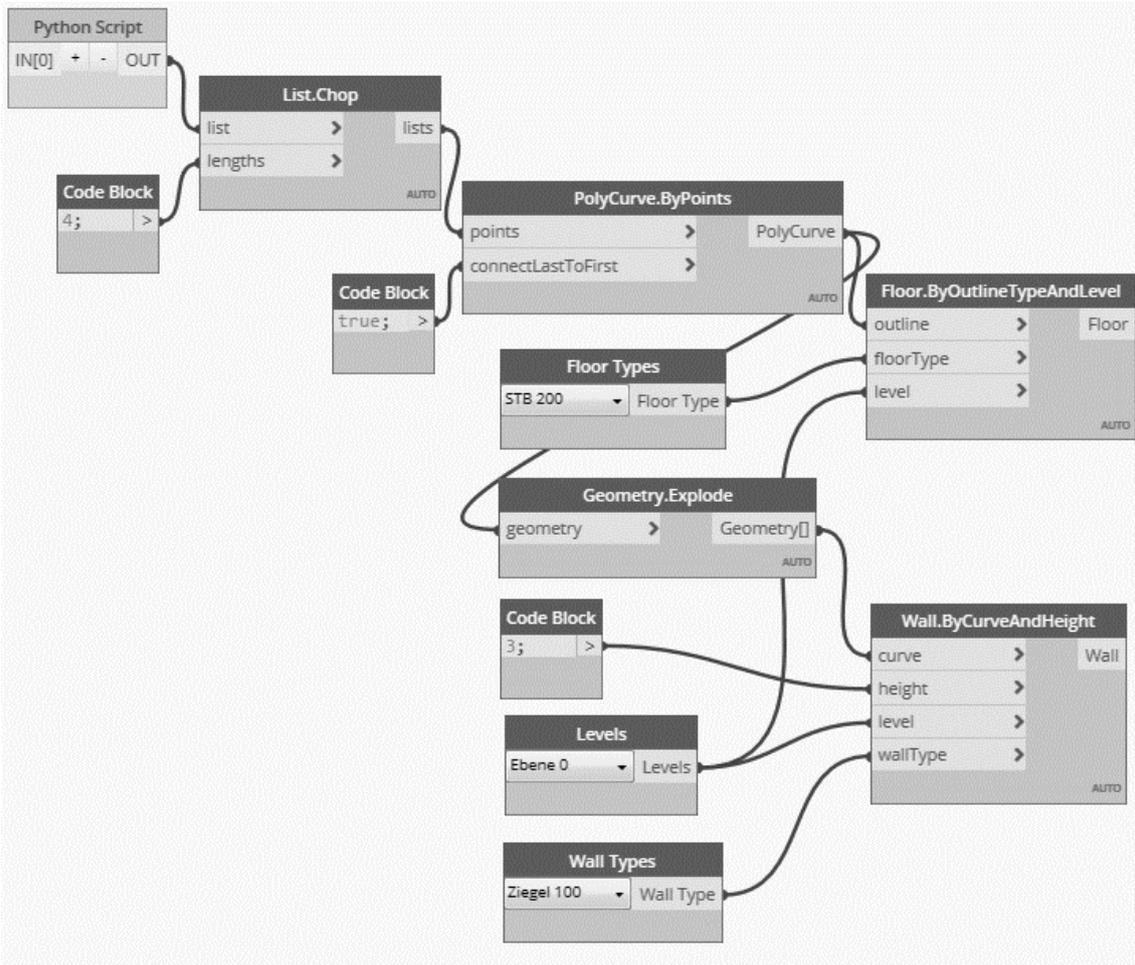


Abbildung 46: Dynamo-Script zum Erstellen des BIM-Modells in Revit (eigene Darstellung)

Über den in Revit integrierten Dynamo-Player ist es möglich den BIM-Grundriss in jedes beliebige Projektmodell zu laden. Dazu können vor dem Starten des Dynamo-Scripts der Wandtyp und die Ebene zum Einfügen gewählt werden. Nach erfolgreicher Ausführung erscheint das BIM-Modell des Grundrisses in Revit.

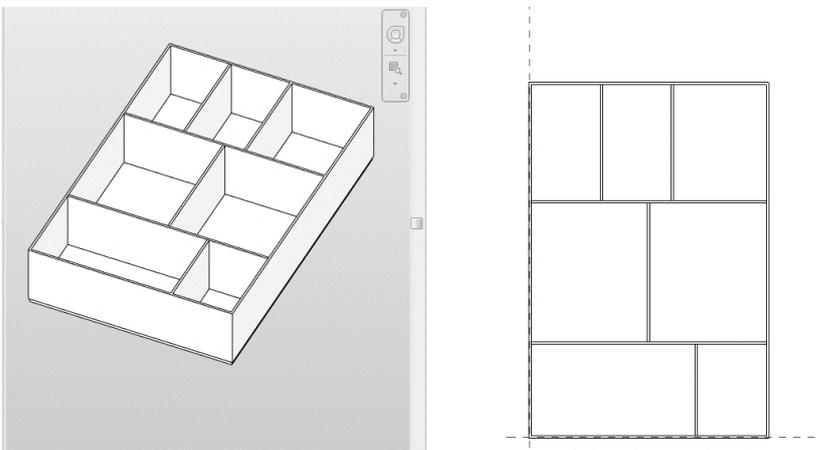


Abbildung 47: 3D-Ansicht (links) und Grundriss (rechts) des erzeugten Wohnungslayouts (eigene Darstellung)

Des Weiteren soll der Software-Prototyp einen Vorschlag zur optimalen Belegung der Zimmer ausgeben. Hierfür ist eine Vorgabe der Idealwerte der einzelnen Räume notwendig, welche über die Eingabetabelle realisiert wird. Jeder Raum wird auf alle aufgelisteten Kriterien geprüft. Die Bewertung erfolgt diesmal über Werte im positiven Bereich, da dies für den Algorithmus zur Auswertung der Bewertungsmatrix notwendig ist. Null bleibt weiterhin das beste Ergebnis, das zu erreichen ist.

Die Bewertung wurde hier um ein Kriterium ergänzt. Wie in Kapitel 3 beschrieben ist die Orientierung der Fenster ein entscheidendes Kriterium für die Nutzung der einzelnen Räume. Dazu wird im Voraus ermittelt, welche Außenwände an den betrachteten Raum angrenzen. Je nach Orientierung wird diese mit den Anfangsbuchstaben der Himmelsrichtungen (N, E, S, W) gekennzeichnet. Bei Räumen, welche fensterlos sein sollen, ist ein Querstrich einzutragen.

```
376 # To check the orientation of the windows
377 def evaluate_orientation(room_fitness, rooms, i, j, l, b):
378     if (sheet.cell_value(10, j+4) == '-'):
379         return room_fitness
380     else:
381         if (rooms[i][1] == 0):
382             window_direction = 'S'
383         elif (rooms[i][0] == 0):
384             window_direction = 'W'
385         elif (rooms[i][0] + rooms[i][2] == 1):
386             window_direction = 'E'
387         elif (rooms[i][0] + rooms[i][3] == b):
388             window_direction = 'N'
389
390         if not (window_direction == sheet.cell_value(10, j+4)):
391             room_fitness += int((sheet.cell_value(10, 1)))
392
393     return room_fitness
```

Abbildung 48: Bewertungsfunktion der Fensterorientierung (eigene Darstellung)

Die aus der Bewertung entstandene Matrix soll nun effizient ausgewertet werden, um die bestmögliche Verteilung der Räume in Abhängigkeit ihrer Nutzung und unter Erfüllung der vorgegebenen Kriterien zu finden. Dazu eignet sich der Kuhn-Munkres-Algorithmus, welcher auch als Ungarische Methode bezeichnet wird. Das Ziel des Zuordnungsproblems ist die Minimierung der Kostenfunktion. Es wird also die Zusammensetzung an bewerteten Räumen gesucht, die die geringste Summe ergibt, wobei jeder Raum genau einmal vorkommen muss.

Im ersten Schritt wird für jede Zeile und Spalte der Minimalwert identifiziert. Dieser wird dann von allen anderen Werten in der Spalte und in der Zeile subtrahiert.

Dadurch entstehen Einträge mit der Wertung null, welche jetzt genau so gewählt werden sollen, dass in jeder Zeile und jeder Spalte nur ein Eintrag markiert ist.

In Python wurde dieser Algorithmus bereits unter dem Zusatzpaket *Munkres* implementiert und kann in Bezug auf das *fitness_array* angewendet werden.

```
346 # Function to choose best possible combination from all possible combinations
347 def Hungarian(fitness_array):
348     m = Munkres()
349     Indices = m.compute(fitness_array)
350
351     return Indices
```

Abbildung 49: Anwendung der Ungarischen Methode in Python (eigene Darstellung)

5.3. EVALUIERUNG

Nach ausführlichen Tests des Softwareprototyps können einige Schlussfolgerungen getroffen werden. Die Laufzeit des Python-Scripts ist von den Einflussparametern abhängig. Maßgebend ist in erster Linie die Anzahl der Generationen und der Individuen pro Generation. Bei einer Änderung der Parameter ist jedes Mal aufs Neue ein Herantasten an die effektivste Anzahl erforderlich, da anfangs zwar schnell bessere Lösungen gefunden werden, hingegen die letzten Versuche bei hohen Anzahlen der Gesamtindividuen aber keine große Verbesserung des Ergebnisses bewirken. Auch das Verhältnis zwischen Mindestraumfläche und Wohnungsgesamtfläche hat entscheidenden Einfluss auf die Dauer für die Erzeugung eines Grundrissentwurfes. Bei zu großen Mindesträumen gibt es weniger Möglichkeiten für eine kollisionsfreie Anordnung. Sollten die Starträume jedoch zu klein gewählt werden, entstehen viele schlauchförmige Räume, da die minimalen Ausgangsräume bei der Expansion nicht behindert werden.

Ungewöhnlich ist die verhältnismäßig lange Laufzeit des Dynamo-Scripts, welches einen relativ kurzen und einfachen Aufbau besitzt. Wichtig ist das Speichern der Eingabetabelle vor Benutzung des Python-Script da sonst die Änderungen der Parameter nicht übernommen werden können.

Auch eine Testreihe zur Validierung des Kreuzungsverfahrens wurde durchgeführt. Dazu wurden 500 Generationen mit lediglich je fünf Individuen verwendet. Eines davon entstand immer aus der Kreuzung der zwei fittesten Grundrisse der vorherigen Generation und die restlichen vier waren die vier fittesten Individuen der letzten Generation. Hierbei verbesserte sich die Bewertung eines Grundrisses signifikant auch ohne die Untermischung zufällig erzeugter Individuen im Rahmen

der Mutation. Die effektive Verbesserung erfolgt hauptsächlich in den ersten Generationen und konvergiert dann gegen einen bestimmten Wert.

Generation	Bewertung
1	259,6277
3	258,3119
14	53,8198
29	47,8441
51	46,0000
76	45,3881
108	45,3178
169	41,8155
188	40,3906
500	40,3906

Abbildung 50: Verbesserung der Bewertung der Grundrisse durch reine Kreuzung (eigene Abbildung)

6. SCHLUSSBETRACHTUNG

6.1. ZUSAMMENFASSUNG UND ERGEBNISSE

Maschinelles Lernen versucht aus bestehenden Daten zu lernen und hierdurch entweder die Qualität seiner Ergebnisse zu verbessern oder auf einem schnelleren Weg zu gleichwertigen Ergebnissen zu gelangen. Trotz der aktuellen medialen Aufmerksamkeit sind die Prinzipien schon länger bekannt und nur durch die gestiegene Rechenpower umsetzbar. Im Groben wird zwischen Supervised und Unsupervised Learning sowie zwischen Regression und Klassifikation unterschieden. Besonders zu beachten ist das Maß der Lernintensität, sodass nicht zu allgemein, aber auch nicht zu detailgenau bzw. auswendig gelernt wird, wie es beim Overfitting der Fall ist. Trotz der gegenwärtig zugänglichen Rechenleistung ist eine vorherige Bereinigung der Daten im Rahmen des Feature Engineering unerlässlich, da sich die benötigte Leistung in höheren Dimensionen vervielfacht.

Zum Vorentwurf von Bauwerken gibt es verschiedene Arten der Randbedingungen. Gesetzliche Vorschriften betreffen beispielsweise Mindestgrößen von Wohnungs- und Zimmerflächen oder Regelungen zur Barrierefreiheit. Auch Abstandsflächen und Brandschutzbestimmungen sind von höchster Bedeutung. Auf der anderen Seite haben sich sogenannte Entwurfsmuster durchgesetzt. Diese sind von den Architekten zu beachten, um ein angenehmes Wohngefühl und ein soziales Miteinander zu gewährleisten. Ein wichtiges Werkzeug dazu ist die Führung des Erschließungsweges vom öffentlichen in den privaten Bereich mit einer entsprechenden Anordnung der jeweiligen Räume im Sinne ihrer Nutzung. Schon die Anordnung der Zimmer in Abhängigkeit von der Tagesbelichtung fällt bei nicht freistehenden Grundrissen sichtlich schwer. Natürlich dürfen die örtlichen Randbedingungen des Bauplatzes nicht vergessen werden. Anschlussleitungen, Erschließungswege für den Straßenverkehr oder Abstandsflächen sind unter allen Umständen einzuhalten. Ästhetische und funktionelle Parameter können meist nicht vollumfänglich und zufriedenstellend umgesetzt werden, da sie sich teilweise gegenseitig ausschließen. Deshalb ist die Suche nach einer optimalen Kompromisslösung von großer Bedeutung.

Für den sozialen Wohnungsbau dient ein vom Bund erlassenes Gesetz als Grundlage, welche landesrechtlich durch Richtlinien vom jeweiligen Bundesland und der zuständigen Kommune ergänzt werden kann. Darin sind beispielsweise

maximale Wohnungsgrößen, Ausstattungsstandards und Bedarfswerte verschiedener Wohnungstypen enthalten.

Um den computergestützten Gebäudeentwurf zu realisieren, muss die Problemstellung in mehrere Teile zerlegt werden, wobei es sich anbietet die Teilprobleme nach den Entwurfsschritten eines Gesamtgebäudes zu gliedern. Als Beispiel zum Einsatz des Machine Learning im Gebäudeentwurf wurde die Grundrissgenerierung einer Wohnung auf einer vorgegebenen Fläche gewählt. In Anbetracht der gegebenen Daten und des umfangreichen Domänenwissens aus den Kapiteln 2 und 3 bietet es sich an, die benötigten Daten selbst zu generieren. Darauf aufbauend wurde eine evolutionäre Strategie als Machine Learning Methode festgelegt. Die Generierung der Zufallsgrundrisse erfolgte durch zufällige Platzierung von Punkten und deren 2-dimensionalen Expansion. Zur Umsetzung der evolutionären Strategie musste ein Bewertungssystem erstellt werden, in welchem die Quantifizierung der Eingangsparameter umgesetzt wurde. Bei den dafür benötigten Bewertungskriterien wurde darauf geachtet, dass genug Freiraum für Kreativität bleibt. Gleichzeitig schränken die Regelungen zum sozialen Wohnungsbau die Formenvielfalt ein, sodass ein Mittelweg gefunden werden musste. Nach einer ausführlichen Erläuterung der Implementierung des Software-Prototypen in Python folgte eine Evaluierung bezüglich der Qualität der Ergebnisse, der Laufzeit und der Einflussnahme der Parameter.

6.2. AUSBLICK

In den nächsten Schritten sind die weiteren Teilprobleme zu lösen, um dann im Hintergrund des ganzheitlichen Konzepts dieser Arbeit eine vollständige Software auszubilden. Sowohl für die Gebäudeplatzierung als auch für die Raumausstattung eignet sich eine evolutionäre Strategie, wie sie in dieser Arbeit angewendet wurde. Die Teillösungen bauen dafür aufeinander auf und folgen dem nativen Entwurfsprozess eines Architekten.

Trotz der verlockenden Vorstellung komplette Gebäude von Grund auf automatisiert generieren zu lassen, wird wohl weiterhin ein Mensch zur Erfassung der Randbedingungen und als Kontrollorgan benötigt werden. Die durch die Software ausgegebenen Entwürfe können immer nur Kompromisslösungen sein, da zumeist nicht alle Kriterien vollständig erfüllt werden können. Deshalb muss aus der Auswahl der je nach Geschmack des Architekten und des Auftraggebers passendste Entwurf nominiert werden.

Trotz all der neuen Anforderungen an den Wohnungsbau haben sich die Schemata großer Wohnungsbauprojekte nicht verändert. Zum einen sind die Bauträger auf Anlageobjekte fixiert, welche möglichst unspezifisch sein sollen, um eine serielle Produktion zu ermöglichen. Zum anderen werden die auf die Kernfamilie angepassten Wohnungsmuster unüberlegt wiederholt, ohne die demografischen Veränderungen zu berücksichtigen. Das lässt die Frage aufkommen, ob die Bauträger die Einsparungen durch computerunterstütztes Entwerfen an die späteren Bewohner weitergeben oder nur in eine Erhöhung der Renditeleistung stecken.

LITERATURVERZEICHNIS

Alexander, Christopher, Ishikawa, Sara und Silverstein, Murray. 2011. *Eine Muster-Sprache - Städte, Gebäude, Konstruktion; zweite und verbesserte Auflage.* Wien : Löcker Verlag GesmbH, 2011.

BBSR, Bundesinstitut für Bau-, Stadt- und Raumforschung. 2017. *Bundesinstitut für Bau-, Stadt- und Raumforschung - Wohnmieten 2016 flächendeckend gestiegen.* [Online] 2017. [Zitat vom: 22. 07 2019.] https://www.bbsr.bund.de/BBSR/DE/Service/Medien/Pressedownloads/entwicklung-angebotsmieten-kreistypen.jpg;jsessionid=0ED425144042D8F3BF2BB309174BBCCD.live21301?__blob=publicationFile&v=2.

BMI, Bundesministerium des Inneren, für Bau und Heimat. 2019. BMI - Soziale Wohnraumförderung. *BMI - Soziale Wohnraumförderung.* [Online] Bundesministerium des Inneren, für Bau und Heimat, 2019. [Zitat vom: 22. 05 2019.] <https://www.bmi.bund.de/DE/themen/bauen-wohnen/stadt-wohnen/wohnraumfoerderung/soziale-wohnraumfoerderung/soziale-wohnraumfoerderung-artikel.html>.

Boersch, Ingo, Heinsohn, Jochen und Socher, Rolf. 2007. *Wissensverarbeitung - Eine Einführung in die künstliche Intelligenz für Informatiker und Ingenieure (2. Auflage).* München : Spektrum Akademischer Verlag, 2007.

Bott, Helmut und v. Haas, Volker. 1996. *Verdichteter Wohnungsbau.* Stuttgart : W. Kohlhammer GmbH, 1996.

Dömer, Klaus, Drexler, Hans und Schultz-Granberg, Joachim. 2016. *Bezahlbar. Gut. Wohnen. - Strategien für erschwinglichen Wohnraum.* Berlin : Jovis Verlag GmbH, 2016.

Donath, Dirk, König, Reinhard und Petzold, Frank. 2012. *KREMLAS - Entwicklung einer kreativen Entwurfsmethode für Layoutprobleme in Architektur und Städtebau.* Weimar : Verlag der Bauhaus-Universität Weimar, 2012.

Dong, Guozhu und Liu, Huan. 2018. *Data Mining and Knowledge Discovery Series: Feature Engineering for Machine Learning and Data Analytics.* Boca Raton : CRC Press Taylor and Francis Group, 2018.

Eichinger, Eva. 2004. *Was ist schon Wohnen? - Abklärungsversuch zur Kultur des Wohnens.* Linz : Johannes Kepler Universität Linz, Institut für Soziologie, 2004.

Ertel, Wolfgang. 2016. *Grundkurs Künstliche Intelligenz - Eine praxisorientierte Einführung (4., überarbeitete Auflage)*. Wiesbaden : Springer Vieweg, 2016.

Frauenhofer-Gesellschaft. 2018. *Maschinelles Lernen - Eine Analyse zu Kompetenzen, Forschung und Anwendung*. München : Frauenhofer-Gesellschaft, 2018.

Frochte, Jörg. 2018. *Maschinelles Lernen - Grundlagen und Algorithmen in Python*. München : Carl Hanser Verlag, 2018. 978-3-446-45291-6.

Géron, Aurélien und Rother, Kristian. 2018. *Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow : Konzepte, Tools und Techniken für intelligente Systeme*. Heidelberg : O'Reilly, 2018.

Görz, Günther, Schneeberger, Josef und Schmid, Ute. 2014. *Handbuch der künstlichen Intelligenz (5. Auflage)*. München : Oldenbourg Wissenschaftsverlag GmbH, 2014.

Heckmann, Oliver und Schneider, Friederike. 2018. *Grundrissatlas Wohnungsbau (5. überarbeitete und erweiterte Auflage)*. Basel : Birkhäuser Verlag GmbH, 2018.

Helic, Denis und Kern, Roman. 2018. Vorlesung: Feature Engineering. *Modul: Knowledge Discovery and Data Mining 1*. Graz : Institute for Interactive Systems and Data Science, University of Technology Graz, 2018.

Keller, Hubert B. 2000. *Maschinelle Intelligenz - Grundlagen, Lernverfahren, Bausteine intelligenter Systeme*. Braunschweig/Wiesbaden : Friedrich Vieweg & Sohn Verlagsgesellschaft mbH, 2000.

Kern, Roman und Helic, Denis. 2017. Vorlesung: Feature Extraction. *Modul: Knowledge Discovery and Data Mining 1*. Graz : Institute for Interactive Systems and Data Science, University of Technology Graz, 2017.

Knecht, Katja. 2011. *Generierung von Grundriss-Layouts mithilfe von evolutionären Algorithmen und k-dimensionalen Baumstrukturen*. Weimar : Bauhaus-Universität Weimar, Professur Informatik in der Architektur, 2011.

Krebs, Jan und Bielefeld, Bert. 2014. *Entwerfen und Wohnen*. Basel : Birkhäuser Verlag GmbH, 2014.

Landeshauptstadt Dresden, Geschäftsbereich Arbeit, Soziales, Gesundheit und Wohnen. 2019. *Leistungen für Unterkunft und Heizung - Leistungen für Unterkunft und Heizung*. Dresden : Eigenverlag, 2019.

Landeshauptstadt Dresden, Stadtplanungsamt. 2018. *Förderprogramm gebundener Mietwohnraum -Bedarf an mietpreis- und belegungsgebundenem*

Mietwohnraum in der Landeshauptstadt Dresden differenziert nach Haushaltsgröße. Dresden : Eigenverlag, 2018.

Neufert, Ernst. 2019. *Bauentwurfslehre - Grundlagen, Normen, Vorschriften über Anlage, Bau, Gestaltung, Raumbedarf, Raumbeziehungen, Maße für Gebäude, Räume, Einrichtungen, Geräte mit dem Menschen als Maßstab.* Wiesbaden : Springer Vieweg Verlag, 2019.

Ognjanovski, Gavril. 2014. Towards Data Science - Everything you need to know about Neural Networks and Backpropagation. [Online] 2014. [Zitat vom: 22. 07 2019.] <https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a>.

ollie.co. 2019. Ollie: All Inclusive Coliving. *Long Island City Apartments - Ollie.* [Online] ollie.co, 20. 05 2019. [Zitat vom: 20. 05 2019.] <https://www.ollie.co/new-york/long-island-city-apartments>.

Schröder, Jörg. 2016. Vermessungsbüro Jörg Schröder - Öffentlich bestellter Vermessungsingenieur. [Online] 2016. [Zitat vom: 22. 07 2019.] <https://www.oebvi-schroeder.de/wissen/abstandsflaechen.html>.

SMI, Staatsministerium des Inneren Sachsen. 2019. Sachsen.de - Bauen und Wohnen - Wohnungswesen - Wohnraumförderung. [Online] 2019. [Zitat vom: 22. 07 2019.] https://www.bauen-wohnen.sachsen.de/download/Bauen_und_Wohnen/Wohnraumfoerderkonzept.pdf.

STUWO. 2012. STUWO Vorgartenstraße Wien 1020. [Online] 2012. [Zitat vom: 22. 07 2019.] <https://stuwo.at/studentenheime/wien/studentenheim-vorgartenstrasse/>.

Wartala, Ramon. 2018. *Praxiseinstieg Deep Learning - Mit Python, Caffe, TensorFlow und Spark eigene Deep-Learning-Anwendungen erstellen.* Heidelberg : dpunkt.verlag GmbH, 2018.

Weinberger, Th., Keller, H. B. und Jakob, W. 1994. *Modelle Maschinellen Lernens - Symbolische und konnektionistische Ansätze.* Karlsruhe : Kernforschungszentrum Karlsruhe, Institut für Angewandte Informatik, 1994.

SELBSTSTÄNDIGKEITSERKLÄRUNG

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und ist auch noch nicht veröffentlicht worden.

Dresden, den 06.09.2019

Johannes Frank Schüler