FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

FH Aachen

Fachbereich
Elektrotechnik und Informationstechnik

Bachelorarbeit

Prognose der Anwesenheit von Personen für die Gebäudeautomatisierung mittels Umweltsensordaten

Alexander Loosen
Matr.-Nr.: 3167353

Referent: **Prof. Dr-Ing. Ingo Elsen**

Korreferent: **Prof. Dr.-Ing. ...**

24. Februar 2022

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Aachen, 24. Februar 2022

Geheimhaltung - Sperrvermerk

Die vorliegende Arbeit unterliegt bis [Datum] der Geheimhaltung. Sie darf vorher weder vollständig noch auszugsweise ohne schriftliche Zustimmung des Autors, des betreuenden Referenten bzw. der Firma [Firmenname und -sitz] vervielfältigt, veröffentlicht oder Dritten zugänglich gemacht werden.

Inhalt

1. Einleitung

Gebäudeautomatisierung bezeichnet die automatische Steuerung und Regelung von Gebäudetechnik wie Heizung, Lüftung oder Beleuchtung. Während sie bisher hauptsächlich für die Optimierung der Energieeffizienz von gewerblichen und öffentlichen Gebäuden genutzt wurde, welche in Zuge solcher Optimierungsschritte als „Smart Buildings" bezeichnet werden, rückt sie in den letzten Jahren zunehmend unter dem Begriff „Smart Home" auch in den privaten Bereich. Die beiden Begriffe stehen in den letzten Jahren so im Vordergrund, weil eine Verbesserung der Energieeffizienz durch bauphysikalische Maßnahmen, wie verminderte Wärmeverluste durch bessere Isolation, an ihre Grenzen gestoßen sind.
Zur weiteren Steigerung der Energieeffizienz ist es also nötig, die Gebäudetechnik automatisch anzusteuern, sodass sog. Performance-Gaps vermieden werden. Performance-Gaps stellen eine Diskrepanz im Energieverbrauch eines Gebäudes zwischen einem theoretischen Soll-Wert zu einem tatsächlichem Ist-Wert dar.

1.1. Motivation und Aufgabenstellung

Für nahezu alle Bereiche der Gebäudeautomatisierung stellt die Anwesenheit von Personen eine zentrale Variable dar. Da die direkte Messung von Anwesenheit über z.B. Infrarotsensoren nicht verlässlich und rechtlich problematisch ist, soll in dieser Arbeit untersucht werden, inwiefern Machine-Learning Algorithmen genutzt werden können, um eine genaue Erwartung über die Anwesenheit von Personen anhand von CO2-Werten in der Raumluft zu treffen.
Die Motivation der Optimierung der Gebäudeautomatisierung exisitert, da ein steigender CO2-Gehalt der Raumluft nachweislich mit einer Abnahme der menschlich kognitiven Leistung einhergeht. Mehrere Studien konnten belegen, dass sowohl sprachliche als auch logisch- mathematische Fähigkeiten abnehmen, sobald der CO2-Gehalt der Raumluft bestimmte Werte überschreitet.
Um eine angemessene Datengrundlage zu schaffen, wurden in diversen Büro-Räumen der FH Aachen Temperatur-, Luftfeuchtigkeits-, Infrarot- und CO2-Sensoren angebracht, deren Messungen kontinuierlich auf einer Datenbank gespeichert wurden. Der Zeitraum der Messwerte begann Mitte 2021. In allen Räumen sind täglich ein oder mehrere Personen im Rahmen eines ca. 8 stündigen Arbeitstages anwesend, weshalb die Temperatur-, Luftfeuchtigkeits- und CO2-Werte als aussagekräftige Indikatoren für menschliche Präsenz angesehen werden können. Es gab keine Einschränkungen hinsichtlich dessen, welche konkreten Machine-Learning Algorithmen benutzt werden sollten.
Als Programmiersprache für das Projekt wurde Python gewählt. Python ist wegen seiner umfassenden Machine-Learning Bibliotheken und einfachen Auswertungstechniken anhand von z.B. Graphen und Statistiken gut für diesen Anwendungsfall geeignet.

1.2. Vorgehensweise

Das Projekte beschäftigte sich im Schwerpunkt mit den folgenden Arbeitsschritten:

- Datenbeschaffung durch Datenbankzugriffe per SQL
- Analyse und Vorbereitung der Daten (Pre-Processing)
- Trainieren von Machine-Learning Models anhand der vorbereiteten Datensets
- Ergebnisauswertung durch Gegenüberstellung verschiedener Datensets und Models

Da es zwischen allen verfügbaren Datensets der einzlnen Räume und Machine-Learning-Models eine Vielzahl an Kombinationsmöglichkeiten gibt, war es ein Anspruch der Projektarbeit, ein möglichst übersichtliches, gut gekapseltes Python Programm zu erstellen, mit dem man einfach und schnell verschiedene Datensets verarbeiten und mit einer dem Forschungszweck angemessenen Anzahl von Machine-Learning Models auszuwerten. Um einen Vergleich der Ergebnisse zu ermöglichen, sollen diese klar und verständlich dargestellt werden. Da nicht bei allen Algorithmen die gleichen Leistungsindikatoren genutzt werden, sollen hauptsächlich nur jene Indikatoren betrachtet werden, die bzgl. aller Algorithmen auch gleiche Bedeutung haben. Falls Model- spezifische Leistungsindikatoren als besonders Erkenntnisreich erachtet werden, wird dies in dieser Arbeit angemerkt.

2. Grundlagen

2.1. Machine Learning

Grundsätzlich beschreibt Machine Learning das Entwickeln mathematischer Modelle zur statistischen Auswertung von Daten. Dabei wird dem Modell anhand von Daten zu einem bestimmten Sachverhalt beigebracht, in einem Datenset Schemata zu erkennen, womit sich eine Erwartung über die Umstände des Datensets treffen lässt. Beispielsweise könnte ein solches Model aus einem Datenset mit der aktuellen Jahreszeit, Uhrzeit und Position der Sonne am Himmel trainiert werden, sodass es auch schließlich in einem anderen Datenset aus Jahreszeit und Position der Sonne Rückschlüsse auf die Uhrzeit treffen kann.
Als Vorbild für diesen „Lernvorgang" dient das menschliche Gehirn, welches ebenfalls versucht zwischen bestimmten Input-Parametern wie z.B. der Form und Farbe eines Gegenstandes eine Beziehung herzustellen, um das beobachtete Objekt in Zukunft schneller kategorisieren zu können.
Da eine Vielzahl von effektiven Machine Learning Algorithmen existiert, ist es essenziell, sich mit den Stärken und Schwächen einzelner Herangehensweisen zu befassen.

Im Wesentlichen kann Machine Learning in zwei Unterkategorien unterteilt werden:

- *Supervised Learning*
- *Unsupervised Learning*

Supervised Learning bedeutet zwischen bestimmten Feldern eines Datensets eine Beziehung zu einem sog. Label herzustellen, welches als eine Art Ergebnis aus den Eingabewerten gesehen werden kann. Ein so traniertes Model kann dann neue, ihm vorher unbekannte Datensets, mit einem Label versehen - etwa wie in dem o.g. Beispiel wo Jahreszeit und Sonnenposition die Eingabewerte und die Uhrzeit das Label darstellen. Der Begriff „*supervised*" ergibt sich daraus, dass das Datenset, mit dem das Model traniert wird, diese Labels gegeben hat, sodass das Modell sich bei jedem Schritt des Lernvorgangs selbst korrigieren kann, falls eine Fehleinschätzung getroffen wurde. Bei einer sog. „*Klassifizierung*" sind diese Labels fest vorgegeben, während sie in der „*Regression*" kontinuierlicher Natur sind. Im Kontext dieser Arbeit wäre das Ergebnis einer Klassifizierung eine „1" für Anwesenheit und eine „0" für Abwesenheit, während das Ergebnis einer Regression eine Wahrscheinlichkeit auf Anwesenheit zwischen 0.0 und 1.0 darstellen würde.

Beim „*Unsupervised Learning*" versucht das Modell ohne Referenz zu einem bestimmten Label, Zusammenhänge zwischen bestimmten Feldern des Datensets herzustellen. Solche Modelle arbeiten vorrangig mit „*Clustering*" und „*Dimensionality Reduction*".
„*Clustering*"-Algorithmen versuchen ein Datenset in kleinere Bereiche einzuteilen und so aus den Feldern des Datensets bestimmte Abhängigkeiten abzuleiten.

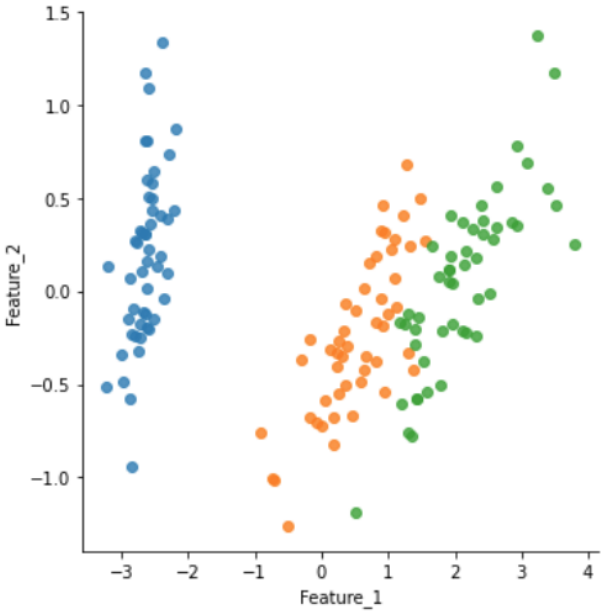

Abbildung 2.1.: Beispiel für Clustering

Bei der „*Dimensionality Reduction*" versucht der Algorithmus das Datenset in einer Dimensionalität, also seiner Anzahl an Feldern, zu reduzieren. Es wird also die Frage gestellt, ob das bestehende Datenset auch mit weniger Feldern Abhängigkeiten feststellen

lässt. Dieser Schritt wird vorallem für Modelle benutzt, die sensibel gegenüber hoher Dimensionalitäten sind, sodass das Datenset vor dem Training in seiner Dimensionalität heruntergebrochen werden kann.
Im Rahmen des Projektes wurden hauptsächlich Klassifizierungs-Algorithmen genutzt, da ein Großteil der Datensets Labels zur Überprüfung hatte. Um einen Vergleich herzustellen werden später trotzdem noch einzelne Ergebnisse von Clustering und Dimensionality Reduction betrachtet. Im folgenden sollen die genutzten Modelle erklärt werden.

2.1.1. Random Forest Classifier

Random Forests stellen eine Unterkategorien der „*Decision Trees*" dar. Decision Trees sind einfache Anordnungen von bestimmten Fragen, die über das Datenset gestellt werden, um eine Klassifikation zu erreichen.

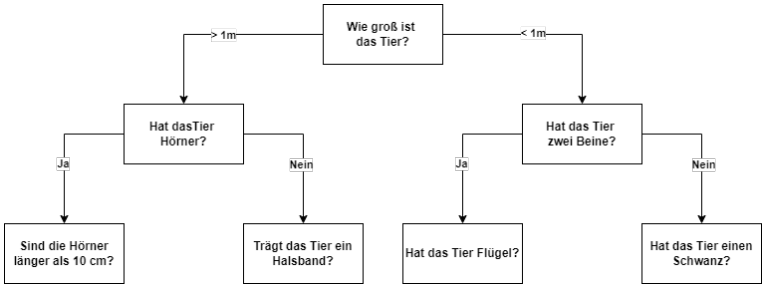

Abbildung 2.2.: Beispiel eines Decision Trees

Erstellt man ein „*Ensemble*" aus Decision Trees die Erwartungen über einen zufällig gewählten Teil des Datensets treffen können, entsteht ein Random Forest. Der Random Forest Classifier versucht, eine Menge einfacher Schätzfunktionen über einen komplexeren Sachverhalt „abstimmen" zu lassen. Während sich in einem einzelnen Entscheidungsbaum Fehleinschätzungen entwickeln können, sinkt die Chance auf eine solche Fehleinschätzung, je mehr unabhängige Entscheidungsbäume man befragt.

2.1.2. Support Vector Classifier

Der Support Vector Classifier(SVC) versucht in einem Datenset anhand von bestimmten Cut-Off-Values klare Grenzen zwischen Werten zu finden, sodass man alle Messwerte ober- und unterhalb der Grenze eindeutig Klassifizieren kann.

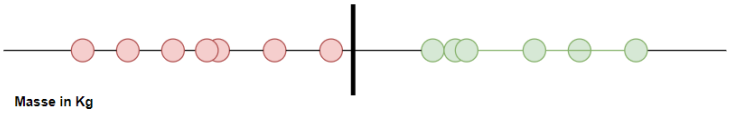

Abbildung 2.3.: Beispiel eines Support Vector Classifiers

In Abb. 2.3 ist der SVC ein Punkt auf einer eindimensionalen Linie, auf der das Gewicht in Kg von z.B. Mäusen in „Unter-" und „Übergewichtig" unterteilt wird. Dieser Punkt ist Ergebnis aller Verhältnisse der einzelnen Datenpunkte zueinander. Durch sog. „*Kernel*

Funktionen" versucht der Algorithmus nun Beziehungen in höheren Dimensionen zu finden, wie z.B. $Masse^2$, $Masse^3$ usw. .
Da der SVC die Verhältnisse aller Datenpunkte zueinander betrachtet, ist er sehr anfällig für Ausreißer in den Daten, was während der Projektarbeit, auch bezüglich anderer Klassifizierungsmethoden, berücksichtigt wurde.

2.1.3. Gradient Boosting Classifier

2.1.4. Bagging Classifier

2.1.5. Neural Networks

2.1.6. Long Short Term Memory

2.2. CO2 als Anwesenheitsindikator

Der CO2-Gehalt der Raumluft ist als sehr guter Indikator für menschliche Präsenz anzusehen. Anders als andere Umweltindikatoren wie Temperatur oder Luftfeuchtigkeit hat der CO2-Gehalt die Eigenschaft, dass es in geschlossenen Räumen keine äußeren Einflussfaktoren für diesen Messwert gibt. In einem Büroraum kann der Mensch als alleinige Quelle für CO2 angesehen werden.
Der Anteil an CO2 in der Atemluft beträgt zwischen 350 und 450 ppm. Es gibt in Deutschland und auch Europa keine grundsätzlich festgelegten Grenzwerte für akzeptable Raumluft, vielmehr raten Gesundheitsämter verschiedener Länder Grenzwerte zwischen 1200 und 1500 ppm einzuhalten. Bei der Obergrenze von 1500 ppm entstehen beim Menschen erste Müdigkeitserscheinungen, weshalb dieser Wert in der Literatur als maximaler Richtwert für Innenräume gilt.

2.3. Outlier Detection

$$(p_i * p_j)(n) = \sum_{k \in \mathbb{D}} p_i(k) \cdot p_j(n-k) \tag{2.1}$$

$$p_{total} = p_0 * p_1 * \ldots p_{n-1}; \forall n \tag{2.2}$$

Hier ist nur eine einfache Formel mit der `equation`-Umgebung für die Minkowski Metrik:

$$D\left(X, Y\right) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{2.3}$$

Wie in Gleichung 2.3 zu erkennen ist, ergibt sich die L2-Norm (Euklidische Distanz), wenn man den Exponenten $p = 2$ wählt.
Support Vector Machines [Haykin, 1999] nutzen die Euklidische Distanz (oder äquivalent) das Skalarprodukt.

3. Kapitel 3

Tabelle 3.1.: Messergebnisse

Stellung	$\frac{T_U}{°C}$	$\frac{T_c}{°C}$	$\frac{\Delta T}{°C}$
senkrecht (0°)	$27,3$	$69,8$	$42,5$
waagerecht (90°)	$26,6$	$70,6$	$44,0$

Wie in Tabelle 3.2 zu sehen ist, ist es besser, Trennlinien nur dort einzusetzen, wo logische Grenzen liegen.

Tabelle 3.2.: Smartphone Sensordaten

Sensorinformation	Format	frequency [s^{-1}]
App identifier for vendor	int64	once per transfer
WIFI and network carrier IP addresses	int128	once per transfer
battery level	int8	0.1
Position information: latitude, longitude, altitude, speed, course, vertical position accuracy, horizontal position accuracy, floor level information	float32[8]	1
Heading information: heading.x, heading.y, heading.z, true heading, magnetic heading, heading accuracy	float16[6]	1
Accelerometer information: acceleration.x, acceleration.y, acceleration.z	float16[3]	2
Gyroscope information: rotationRate.x, rotationRate.y, rotationRate.z	float16[3]	2
altimeter information: relative altitude, pressure	float16[2]	1
timestamp	uint32	once per transfer
Temperature [°C]	float16	1

4. Zusammenfassung und Ausblick

Quellenverzeichnis

[Hartnett, 2018] Hartnett, K. (2018). Machine learning confronts the elephant in the room. Quanta Magazine, Online. https://www.quantamagazine.org/machine-learning-confronts-the-elephant-in-the-room-20180920/.

[Haykin, 1999] Haykin, S. (1999). *Neural Networks - A Comprehensive Foundation*. Prentice Hall, 2 edition.

[Le, 2018] Le, J. (2018). How to do semantic segmentation using deep learning. Online. https://medium.com/nanonets/how-to-do-image-segmentation-using-deep-learning-c673cc5862ef.

Abkürzungsverzeichnis

g	Gravitation in Nähe der Erdoberfläche
Nu	Nußelt-Zahl
ν_{Luft}	Kinematische Viskosität von Luft
Pr	Prandtl-Zahl
$\dot{Q}$	Wärmestrom
Ra	Rayleigh-Zahl
ρ_{Luft}	Dichte von Luft
T	Temperatur
T_∞	Umgebungstemperatur

Abbildungsverzeichnis

Tabellenverzeichnis

A. Quellcode

1. Source 1
2. Source 2

B. Rohdatenvisualisierungen

1. Graustufen
2. Verteilungen