



Fachhochschul-Bachelorstudiengang  
**MEDIZIN- UND BIOINFORMATIK**  
A-4232 Hagenberg, Austria

# Bachelorarbeit

zur Erlangung des akademischen Grades  
Bachelor of Science in Engineering

Eingereicht von

**Philipp Krainer**

Hagenberg, Dezember 2016

# Inhalt

- |  |              |
|--|--------------|
| Teil 1:<br>Klassifikation und Clustering                                       | Seite Nr. 4  |
| Teil 2:<br>Outlook/Exchange Adapter für CGM G3 Clinical Information System MRP | Seite Nr. 55 |

### **Eidesstattliche Erklärung**

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Datum, 13.12.2016

Unterschrift



Fachhochschul-Bachelorstudiengang  
**MEDIZIN- UND BIOINFORMATIK**  
A-4232 Hagenberg, Austria

# **Klassifikation und Clustering**

Bachelorarbeit  
Teil 1

zur Erlangung des akademischen Grades  
Bachelor of Science in Engineering

Eingereicht von

**Philipp Krainer**

Begutachter: DI Dr. Stephan Winkler

Hagenberg, Dezember 2016

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>6</b>
<b>Abstract</b>	<b>7</b>
<b>1 Einleitung</b>	<b>8</b>
1.1 Einführung . . . . .	8
1.2 Aufgabenstellung . . . . .	8
1.3 Terminologien und Begrifflichkeiten . . . . .	9
<b>2 Clustering</b>	<b>12</b>
2.1 Einführung . . . . .	12
2.2 Datenrepräsentation . . . . .	12
2.3 Algorithmen . . . . .	14
2.3.1 Allgemein . . . . .	14
2.3.2 k-means Algorithmus . . . . .	16
2.3.3 Hierarchisch Clustering . . . . .	17
2.3.4 Self-organizing Maps . . . . .	19
2.3.5 Graph-based Clustering . . . . .	20
2.3.6 Spectral Clustering . . . . .	20
2.4 Anwendungsgebiete . . . . .	22
2.5 Visualisierung . . . . .	22
2.6 Zusammenfassung und Ausblick . . . . .	23
<b>3 Klassifikation</b>	<b>24</b>
3.1 Einführung . . . . .	24
3.2 Klassifikationsgüte . . . . .	25
3.2.1 Beschreibung . . . . .	25
3.2.2 Train und Test . . . . .	25
3.3 Algorithmen . . . . .	26
3.3.1 Bayes-Klassifikatoren . . . . .	26
3.3.2 Entscheidungsbäume . . . . .	28
3.3.3 ( $k$ )-Nächste-Nachbarn-Klassifikatoren . . . . .	31
3.4 Zusammenfassung und Ausblick . . . . .	32

<b>Inhaltsverzeichnis</b>	<b>5</b>
<b>4 Beispiele</b>	<b>33</b>
4.1 Tools . . . . .	33
4.2 Klassifikation . . . . .	33
4.2.1 Einführung . . . . .	33
4.2.2 Iris Datensatz . . . . .	33
4.2.3 Wisconsin Diagnostic Breast Cancer Datensatz . . . . .	40
4.2.4 Fazit . . . . .	47
4.3 Clustering . . . . .	47
4.3.1 Einführung . . . . .	47
4.3.2 Iris Datensatz . . . . .	48
4.3.3 Wisconsin Diagnostic Breast Cancer Datensatz . . . . .	49
4.3.4 Fazit . . . . .	50
<b>5 Schluss</b>	<b>51</b>
5.1 Zusammenfassung . . . . .	51
<b>Quellenverzeichnis</b>	<b>52</b>
Literatur . . . . .	52

# Kurzfassung

Die vorliegende Arbeit setzt sich mit den heuristischen Algorithmen der Klassifikation und des Clustering auseinander. Das Clustering, welches auch als unüberwachte Klassifikation bekannt ist, wird als eine Methode zur Einteilung von großen Datensätzen angewandt. Hier wird auf den unterschiedlichen Bezug der Daten auf das Clustering und dessen Algorithmen eingegangen. Verschiedene Algorithmen geben einen guten Einblick in die komplexe Welt des Clustering. Grundsätzlich gilt, dass jede Methode ihr eigenes Einsatzgebiet innehat und an die vorliegenden Daten anzupassen ist. In weiterer Folge wird mit der Klassifikation eine weitere Methode beschrieben Daten einzuteilen. Im Gegensatz zum Clustering werden bei Klassifikation Daten anhand vorgegebener Klassen zugeordnet. Auch hier wurden die verschiedenen Algorithmen unter der Berücksichtigung der Klassifikationsgüte beschrieben und verglichen. Die Algorithmen verwenden bereits vordefinierte Daten und Methoden um die Berechnung zu beschleunigen. Daher ist die Klassifikation weiterverbreiteter als das Clustering. Es wurden die Algorithmen und Methoden anhand von Beispielen entsprechend dargestellt. Dabei kamen das Heuristiclabtool sowie Daten aus UCI Repository zur Anwendung. Beispiele zeigen ziemlich gut welche Algorithmen für welche Datensätze geeignet sind und beste Ergebnisse liefern.

# Abstract

This thesis deals with the comparison of heuristic algorithms of classification and clustering. First we discuss clustering as a part of the unsupervised classification. This is used to divide big data into smaller and different parts. Different algorithms and Methods describe the common used strategies of clustering relatively well. Basically every data set is different and not every algorithm can be applied to it, so every method has to be customized to fit into the data set properties. In contrast to the Clustering the Classification separates data sets based on predefined classes in groups by using train data and test data. Different algorithms are described to show a number of different methods to use classification. A main measurement for the algorithms is basically classification quality. The algorithms use predefined data to boost computing performance. That means Classification is used more often than clustering. Finally some examples and tests of the different algorithms of classification and clustering are described. The HeuristicLab software is used for the calculation of the algorithms and the sample data is provided from the UCI repository. All samples show us which kind of algorithms can be used for the different data sets and the best results.

# Kapitel 1

## Einleitung

### 1.1 Einführung

In dieser Bachelorarbeit werden die Themen *Klassifikation* und *Clustering* in Bezug auf heuristische Methoden behandelt. In der heutigen Welt sind große Datenmengen an der Tagesordnung. Die Hardware ist zwar leistungsfähig, kann aber große Datensätze nicht auf einmal verarbeiten, da die zeitliche Datendurchsatzrate zu gering ist. Daher werden große Datenmengen in kleinere Einheiten mit Hilfe von Klassifikation und Clustering übergeführt. *Big Data* bestimmt die Welt der Analyse und der Verarbeitung von Informationen, welche überwiegend als mehrdimensionale Daten beschrieben werden können. Daher ist es wichtig geeignete Algorithmen und Verfahren zu finden, welche große Datenmengen in anwendbare Einheiten einteilen. Das Ziel der Analyse von eingeteilten Daten ist es eine verständliche und interpretierbare Konzeption zu erreichen, die auf einem geeigneten Modell basiert. Dieses Modell kann nur auf der Basis von eingeteilten Daten erreicht werden, da es sonst zu komplex wird. Das Clustering und die Klassifikation ermöglichen diese Vorgangsweise. Sie sind daher ein zentraler Bestandteil bei der Analyse von Daten und in der Heuristik und Statistik nicht mehr wegzudenken.

### 1.2 Aufgabenstellung

In der Klassifikation geht es Samples in verschiedene Klassen einzuteilen. Beim Clustering dagegen geht es Gruppen von Datenpunkten zu identifizieren. Ist es sinnvoll Daten vor dem Anwenden von Klassifikationsalgorithmen zu gruppieren, also Clustering und Klassifikation zu kombinieren? Lässt sich so die Prognosegenauigkeit erhöhen? Ziel dieser Arbeit ist es, Antworten auf diese Fragen zu finden; Frameworks wie das HeuristicLab und WEKA sowie international bekannte Benchmark-Datensätze können für die entsprechenden Tests verwendet werden.

### 1.3 Terminologien und Begrifflichkeiten

- **Heuristik:**

Aus dem Griechischen heuriskein = finden, entdecken, bezeichnet eine Erfinderkunst. Heuristik ist die Lehre von verschiedenen Verfahren zum Lösen von Problemen, welche nicht mit mathematischen Algorithmen bzw. Formeln gelöst werden können.

- **Datensatz:**

Ein Datensatz ist die Zusammenfassung von Daten, die in einer direkten Beziehung zueinander stehen oder gemeinsame Merkmale haben. Daten, die in einem Sinnzusammenhang stehen, können dabei in einem Ordnungssystem zusammengefasst sein.

- **Wahrscheinlichkeit:**

Die Wahrscheinlichkeit ist ein Maß zur Quantifizierung der Sicherheit bzw. Unsicherheit des Eintretens eines bestimmten Ereignisses im Rahmen eines Zufallsexperiments

- **Algorithmus:**

Ein Algorithmus ist eine eindeutige, ausführbare Folge von Anweisungen endlicher Länge zur Lösung eines Problems. Ein Algorithmus besteht aus einem Deklarationsteil und einem Anweisungsteil.

- **Sample (Sampling):**

Teilmenge einer Grundgesamtheit, die für eine Untersuchung ausgewählt wird.

- **Satz von Bayes:**

Der Satz von Bayes ist ein mathematischer Satz aus der Wahrscheinlichkeitstheorie, der die Berechnung bedingter Wahrscheinlichkeiten beschreibt. Formel:

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

- **Test:**

Tests sind Methoden, mit denen eine Entscheidung über die Beibehaltung oder Zurückweisung einer Nullhypothese  $H_0$  mithilfe eines Stichprobenbefundes getroffen wird.

- **Graph:**

Ein Graph ist in der Graphentheorie eine abstrakte Struktur, die eine Menge von Objekten zusammen mit den zwischen diesen Objekten bestehenden Verbindungen repräsentiert. Die mathematischen Abstraktionen der Objekte werden dabei Knoten des Graphen genannt. Die paarweisen Verbindungen zwischen Knoten stellen Kanten dar.

- **Daten:**

- numerisch:

Daten, die mit Ziffern und zusätzlichen Sonderzeichen dargestellt werden.

- nicht-numerisch:

Daten, die aus Buchstaben und Ziffern zusammengesetzt sind.

- **Zeitreihe:**

Ein Zeitreihe ist eine Serie von Messungen, Beobachtungen und Aufzeichnungen von Variablen an aufeinanderfolgenden Zeitpunkten. Zeitreihen ermöglichen eine strukturierte Darstellung von Daten. Eine visuelle Darstellung entspricht einer Kurve, die sich mit der Zeit entwickelt.

- **Dimension:**

Der Begriff Dimension bezeichnet im Allgemeinen lediglich ein unabhängiges Merkmal eines Datensatzes. Dimensionen haben mit den Daten selbst gar keine echte Verwandtschaft, sondern stellen meist ein unabhängiges Gedankenkonstrukt dar, das Analogien zum Datensatz herstellt, um es berechenbar oder messbar zu machen.

- **Distanzmaß:**

Als Distanzmaß wird ein Maß bezeichnet, wenn es die Unähnlichkeit zwischen zwei Objekten misst. Es besitzt die Eigenschaft, dass es mit zunehmender Unterschiedlichkeit zweier Objekte ansteigt.

- **Kostenfunktion:**

Als Kostenfunktion wird jene Funktion beschrieben, welche bestimmt wie komplex und aufwendig ein Algorithmus oder Verfahren ist. Meist wird diese Funktion mit der O-Notation gleich gesetzt, welche vor allen in der Laufzeitmessung angewandt wird.

- **Lagemaße:**

- Mittelwert:

Der Mittelwert beschreibt den statistischen Durchschnittswert und wird auch arithmetisches Mittel genannt.

- Median:

Der Wert, der genau in der Mitte einer Datenverteilung liegt, nennt sich Median oder Zentralwert. Die eine Hälfte aller Daten ist immer kleiner, die andere größer als der Median.

- Modus:

Der Modus gibt an, welche Merkmalsausprägung in einem Datensatz am häufigsten vorkommt.

- **Heatmap:**

Eine Heatmap ist ein Diagramm zur Visualisierung von Daten, deren abhängige Werte einer zweidimensionalen Definitionsmenge als Farben repräsentiert werden. Sie dient dazu, in einer großen Datenmenge intuitiv und schnell besonders markante Werte zu erfassen.

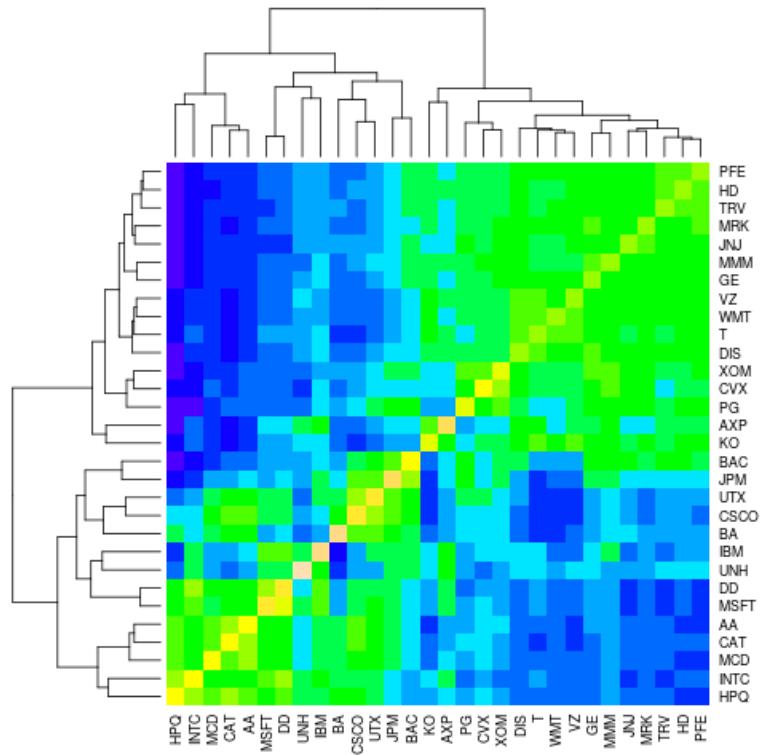


Abbildung 1.1: heatmap

- **Cluster:**

Als Cluster bezeichnet man in der Informatik und Statistik eine Gruppe von Datenobjekten mit ähnlichen Eigenschaften.

- **Link Arten:**

- Single-Link:

Beschreibt die kleinste Entfernung von den Clustern und wird auch als nächster Nachbar bezeichnet.

- Average-Link:

Beschreibt die mittlere Distanz von Clustern zueinander

- Complete-Link:

Beschreibt die maximale Entfernung von Clustern zueinander.

# Kapitel 2

# Clustering

## 2.1 Einführung

In der Informatik und Statistik geht es um große Datensätze. Diese können analysiert werden, wenn entsprechende Tools und Algorithmen zur Verfügung stehen. Durch das Clustering und der Bildung von eingeteilten Datensätzen wird das Arbeiten mit großen Daten erleichtert. Das Ziel ist ein geeignetes Modell für einen gegebenen Datensatz zu finden. Durch diese Methoden wird das Interpretieren von Merkmalen und Besonderheiten erleichtert und ermöglicht außerdem eine begünstigte Aufbereitung von den Daten. Um die richtigen Modellrepräsentationen zu finden und diese zu identifizieren ist es notwendig die Daten zuerst in Klassen einzuteilen. Damit beschäftigt sich die Klassifikation. Dies ist sehr hilfreich wenn ein klassenorientiertes Clustering vorgenommen wird, da die Einteilung zu relevanten Merkmalen zuerst erfolgen sollte. Dabei wird zwischen zwei Arten unterschieden: Die überwachte (*supervised*) und die unüberwachte(*unsupervised*) Klassifikation. Clustering gehört zu der unüberwachten Klassifikation, welche die Daten in relevante Klassen sogenannte Cluster einteilt. Das Einsatzgebiet von Clustering-Algorithmen ist vielfältig. Die Anwendungsbereiche sind vor allem im Gebiet von Data-Mining und Data-Analysis zu finden. Weitere Anwendungsbereiche sind die Bioinformatik, Analyse von Datenbanken, Textmining u. Neuronale Netzwerke. Clustering spielt bei der Datenanalyse eine große und bedeutende Rolle.[4, 2]

## 2.2 Datenrepräsentation

Als *Daten* wird eine Ansammlung bzw. eine Menge an Dingen oder Objekten bezeichnet. Diese Definition bezieht sich auf den Zusammenhang von Daten und Clustering. Jedes einzelne Objekt besitzt spezielle Eigenschaften, welche eindeutig per Objekttyp sein können. Diese Eigenschaften werden oft auch als Attribute, Merkmale oder Dimensionen bezeichnet.[12]

In einen Datenraum befinden sich Objekte oder Elemente mit einer endlichen Anzahl von Merkmalen, welche bei allen Objekten denen des Datenraumes gleichen. Jedoch können sich Daten die sich innerhalb dieses vorgegebenen Raumes befinden sich unterscheiden, da die Ausprägung nicht vorgegeben ist. [12, 6] Dabei gibt es eine mathematische Repräsentation:

$D, D_d$  : Datenraum (auch Merkmalsraum) der Dimension  $d$  (auch  $\mathbb{R}^d$ )

$S$  : Datenmenge,  $S \subset D$

$x_i$  :  $i$ -tes Objekt aus  $S$

$|S|, n$  : Mächtigkeit von  $S$  (Anzahl von Objekten)

$$\text{Objekte} \left\{ \begin{array}{l} x_1 = \underbrace{(x_{1,1} \quad \cdots \quad a_{1,d})}_{\text{Attribute}} \\ x_2 = (x_{2,1} \quad \cdots \quad a_{2,d}) \\ \vdots \qquad \vdots \quad \ddots \quad \vdots \\ x_n = (x_{n,1} \quad \cdots \quad a_{n,d}) \end{array} \right.$$

Jedes einzelne Objekt aus der Datenmenge  $S$  hat  $d$  Merkmale/Attribute. Diese werden durch die Art (des Typs) unterschieden. Dabei wird zwischen *numerisch* und *nicht-numerisch* Daten unterschieden. Erstens betroffen sind die Daten bzw. Objekte in Vektoren von reelwertigen Zahlen. Als Beispiel sei eine Zeitreihe von einer Messung genannt. Zweitens betrifft den Rest, welcher in ein numerisches Format übertragen werden kann. Im Allgemeinen können *nicht-numerische* Merkmale durch spezifische Codierungen in numerische Merkmale übertragen werden. Dabei wird jedes einzelne Merkmal durch ein oder mehrere Attribute repräsentativ dargestellt. Diese Methode wird verwendet um *numerische* und *nicht-numerische* Daten gleich zu behandeln, da es dabei keinen Unterschied in der Anwendung bzw. Auswertung gibt. Diese Methode ist notwendig um die Daten in ein Format zu bringen, welches für das Clustering verwendet werden kann. Dann können die Daten von einem geeigneten Algorithmus verarbeitet werden. In der realen Welt sind große und komplexe Daten an der Tagesordnung. Im Gegensatz dazu würde es sich sehr aufwändig gestalten, wenn Algorithmen mit hoher Dimensionalität an Daten verwendet werden. Da sich dabei die Rechenzeit erheblich erhöhen würde müssen Methoden angewendet werden um die Datenkomplexität zu reduzieren. Häufig sind bei Datensätzen zu viele Merkmale vorhanden, die sogar irrelevant für die Berechnung sind. Man-

che können den Algorithmus sogar in eine falsche Richtung führen. Es die Möglichkeit gewisse Merkmale auszuklammern d.h. diese werden nicht in die Berechnung aufgenommen um die Komplexität zu verringern. Dabei gehen keine wichtigen Attribute verloren. Dieses Verfahren wird als Merkmalsauswahl(*feature selection*) genannt. Es ist oft hilfreich nur gewisse Merkmale auszuwählen, um eine Selektion von den Besten zu ermöglichen (auch Elitismus genannt), denn dann sind die Algorithmen performant und liefern annehmbare Ergebnisse. Eine weitere Möglichkeit ist es gewisse Merkmale aus einer Ansammlung auszuwählen. Dadurch kann auch eine Featurerduktion erreicht werden. Dies ist ebenso so effizient wie die Auswahl der Merkmale. Diese Methode wird Merkmalsextraktion(*feature extraction*) genannt. Dabei werden nur die wichtigsten Merkmale herangezogen, um die Performance zu steigern. Neben den beiden Methoden ist es auch notwendig die Daten zu normalisieren d.h. es müssen die Daten auf die gleiche Weise umgerechnet werden, damit sie besser zusammenpassen. Dies wird durch skalieren und konvertieren erreicht. Die Normalverteilung der Daten wird angenommen um die Skalierung zu erleichtern; mit denselben Mittelwert ( $\bar{x}$ ) und der selben Standardabweichung ( $\sigma$ ), mit  $\bar{x} = 0$  und  $\sigma = 1$ .[12]

Bevor die Algorithmen angewendet werden können, muss ein geeignetes Maß für den Abstand gefunden werden. Die Maße sind metrisch u. es wird daher der Überbegriff der Ähnlichkeitsbestimmung verwendet. Diese Maße geben an wie ähnlich sich zwei Objekte sind und dabei wird nicht zwischen *numerisch* und *nicht-numerisch* unterschieden. Das Distanzmaß (*distance measure*) oder Ähnlichkeitsmaß (*similarity/proximity/affinity measure*) definiert die Beziehung bzw. die Funktion  $d : D \times D \rightarrow Z$ . Die Maße sind essentiell für das Clustering und müssen vor den eigentlichen Clustering ausgeführt werden.[4]

## 2.3 Algorithmen

### 2.3.1 Allgemein

Nachdem ein geeignetes Distanzmaß gefunden ist, kann ein bestimmter Algorithmus auf die Daten angewendet werden. Es gibt zwei Gruppen von Verfahren bzw. Algorithmen, welche das *Hierarchisches Clustering* und die *Partitionierung* darstellen. Bei der Partitionierung werden die Objekte bzw. Daten in Gruppen eingeteilt und diese Gruppen enthalten keine weiteren verschachtelten Cluster und besitzen nur eine Ebene. Beim *Hierarchischen Clustering* entsteht ein geschachtelter Aufbau bzw. eine Struktur, wo größere Cluster kleinere enthalten.[4]

Weiters können die beiden oben angeführten Methoden weiter aufgegliedert werden:[4]

- *divisiv:*

Bei dieser Methode werden alle Objekte einen Cluster zugeordnet sowie schrittweise verkleinert, indem schrittweise zerteilt wird, bis ein vordefiniertes Abbruchkriterium eintritt.

- *agglomerativ:*

Im Gegensatz zu der divisiven Methode wird bei der agglomerativen Methode mit kleinen Clustern begonnen. Jedes Objekt stellt einen Cluster für sich dar. Die kleinen Cluster werden schrittweise zusammengefügt bis ein Abbruchkriterium eintritt.

- *hard:*

Algorithmen welche das Prinzip von einer strikten Vorgehensweise (*hard*) verfolgen ordnen einen Cluster ein Objekt zu.

- *fuzzy:*

Im Gegenteil dazu gibt es das Prinzip von der ungenauen Vorgehensweise (*fuzzy*), dabei können Objekte verschiedenen Clustern zugeordnet werden.

- *stochastisch:*

Bei dem Begriff stochastisch kann davon ausgegangen werden, dass der Zufall eine Rolle spielt und die Auswahl verschiedener Objekte oder Attribute keiner Regel folgt.

- *deterministisch:*

Dabei handelt es sich um die Vorgabe keiner zufälligen Ereignisse. Hier muss alles vorgegeben sein damit es als deterministisch gilt.

- *monothetisch:*

Wenn bei der Verarbeitung nur ein Cluster bzw. ein Objekt verarbeitet wird, dann wird dieses Verfahren monothetisch bezeichnet. Aber die Algorithmen arbeiten nur bedingt nach diesem Prinzip, da dadurch die Berechnungszeit erhöht sein kann.

- *polythetisch:*

Bei der polythetischen Vorgangsweise werden Cluster bzw. Objekte oder Daten schneller verarbeitet, da Vorgänge gleichzeitig ausgeführt werden. In Bezug auf das Clustering bezieht sich die Gleichzeitigkeit auf die Distanzberechnung der Merkmale.

### 2.3.2 k-means Algorithmus

#### Beschreibung:

Der *k-means* Algorithmus gehört zu den Partitionierungs-Algorithmen. Die Implementierung ist einfach und liefert trotzdem gut interpretierbare Ergebnisse für einfache Aufgabenstellungen. Grundlegend versucht der Algorithmus eine Partition in den Daten zu finden und daraus dann Cluster zu bilden. Die Anzahl der Cluster wird durch den Anwender festgelegt und während der Laufzeit nicht mehr geändert. Die Formel nach dem die Cluster gebildet werden lautet: [12, 8, 10]

$$x_r^i : r\text{-te Element des Clusters } C_i$$

Diese Methode wird auch als Sum-of-Squares bezeichnet. Dabei werden die quadratischen Abstände minimiert. Beim Clustering bedeutet dies, dass die Ähnlichkeit der Attribute, Merkmale oder Objekte bestimmt wird. Damit basieren Cluster auf der oben angeführten Kostenfunktion.

#### Algorithmus:

1. Wähle zufällig  $k$  Cluster-Zentren  $\mu_1, \dots, \mu_k$ .
2. Berechne für jedes  $x \in S$ , zu welchen Clustermittelpunkt  $\mu_i$  es am nächsten liegt.
3. Berechne für jeden Cluster  $C_i$  die Kostenfunktion:

$$c(C_i) = \sum_{r=1}^{|C_i|} (d(\mu_i, x_r^i))^2$$

4. Berechne für jeden Cluster  $C_i$  den eigenen neuen Mittelpunkt:

$$\mu_i = \frac{1}{|C_i|} \sum_{r=1}^{|C_i|} x_r^i$$

5. Wiederhole 2., 3., 4. bis sich die Clusterzuordnung nicht mehr ändert.

Die Datensätze besitzen einen Mittelwert, da diese *numerisch* sind. Daher kann ein Mittelwert oder auch das arithmetisches bzw. geometrisches Mittel gebildet werden. Auch beim Clustering können *nicht-numerische* Daten-

sätze verwendet werden. Diese besitzen meistens keinen numerischen Mittelwert. Dennoch kann ein Lagemaß berechnet werden: der Median. Dieser gibt ähnlich wie der Mittelwert eine gute Aussage wie die Daten verteilt sind und es können auch damit *nicht-numerischen* Daten berechnet werden. Beim Clustering wird der ähnliche *k-medoids* Algorithmus angewandt, welcher nach dem oben genannten Prinzip funktioniert. Nur wird bei der Berechnung der Mittelwert  $\mu$  durch den Median ersetzt. [12, 8]

### Zusammenfassung

Dieser Clusteringalgorithmus ist einfach in seiner Komplexität, da er sich nur auf die Mittelwerte der einzelnen Attribute bezieht. Doch bei der Bildung von Clustern können einfach sphärische Cluster entstehen, da die Berechnung relativ zum Mittelwert geschieht. Weiteres kann ein vorhandenes Rauschen in den Daten (Störung in den Daten) bei diesem Algorithmus nicht beseitigt werden, da der Mittelwert sehr ausreißerempfindlich ist.

#### 2.3.3 Hierarchisch Clustering

##### Beschreibung:

Diese Methode wird verwendet wenn die Daten nicht offensichtlich in Gruppen bzw. Partitionen eingeteilt sind oder es keine separierte Cluster gibt. Diese Methode erstellt eine hierarchische Baumstruktur der Datenmenge. Dabei sind die einzelnen Knoten bzw. Enden jeweils eine Teilmenge des übergeordneten Knotens. Der Wurzelknoten repräsentiert die gesamte Menge und die Blätter die einzelnen Objekte. Bei diesem Algorithmus werden *bottom-up* und *top-down* als Verfahren unterschieden. Bei der *bottom-up*-Methode wird anfangs von kleinen Elementen bzw. Clustern ausgegangen. Diese werden immer weiter kombiniert bis ein gemeinsamer Megacluster entsteht, welcher den ganzen Datensatz enthält. Wenn ein großer Cluster entstanden ist, ist der Algorithmus durchlaufen bzw. das Clustering abgeschlossen.[7, 9]

Im Gegensatz dazu wird bei der *top-down* Methode von einen einzelnen Cluster ausgegangen, welcher den ganzen Datensatz enthält. Hier wird schrittweise der übergeordnete Cluster auch *parent* genannt in mehrere kleinere Cluster (*child*) zerlegt. Diese stellen den Eltern-Cluster dar. Wenn in jeden Cluster nur mehr ein Element vorhanden ist, ist das Clustering abgeschlossen. Beide Methoden verwenden eine Baumstruktur im Hintergrund, in der die einzelnen Cluster als Knoten repräsentiert werden. Dadurch kann mit größeren und komplexeren Datensätzen gearbeitet werden. [7, 9]

### Algorithmus

Der Algorithmus wird anhand der der *bottom-up* Methode erklärt. Dabei wird eine Vereinigung von zwei Clustern verwendet. Die zweite Methode *top-down* kann durch durch teilen der Cluster beschrieben werden:

1. Beginne mit  $n$  Clustern  $C_1, \dots, C_n$ ; wobei  $C_i = x_i$ .
2. Minimiere die Kostenfunktion  $c(C_i, C_j)$ , um die beste bzw. *günstigste* Vereinigung ( $C_i \cup C_j$ ) zu finden.
3. Ersetze  $C_i$  und  $C_j$  durch die Vereinigung  $C_i \cup C_j$ .
4. Wiederhole die Schritte 2 und 3 bis alle Cluster zusammengefasst sind.

Das hierarchische Clustering ist nur ein Verfahren, dass einzelne Algorithmen implementiert. Die Vorgehensweise unterscheidet sich nur in der Ausführung des Verfahrens. Es wird unterschieden zwischen *Single-Link*, *Average-Link* und *Complete-Link*. Dabei unterscheiden sich die Verfahren nur in der Kostenfunktion[4, 7, 9]:

- *Single-Link*:

$$c(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

- *Average-Link*:

$$c(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$

- *Complete-Link*:

$$c(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$

### Zusammenfassung

Die Datenmatrix bzw. der Datenvektor spielen bei dieser Methodik eine geringe Rolle, da eigentlich beim Clustering speziell beim hierarchischen meistens auf die Distanzen Rücksicht genommen wird und diese als Eingabe eingesetzt werden. Meist werden diese Distanzen durch eine Matrix repräsentiert. Die Dimensionen werden als  $n \times n$  dargestellt, welche bei großen

Datenmengen oder großen  $n$  zu Speicherproblemen führen können, da die Datenmenge sehr schnell ansteigen kann. Abhilfe kann geschaffen werden indem ein Schwellenwert festgelegt wird. Damit werden unbedeutende Wertepaare vernachlässigt. Weiters können auch die Anzahl der verwendeten Elemente begrenzt werden und dadurch die repräsentative Menge der Elemente reduziert werden. Auch können die Verlinkungen zu Nachbarn begrenzt werden um die Anzahl der nächsten Nachbarn zu begrenzen.

### 2.3.4 Self-organizing Maps

#### Beschreibung:

Bei diesen Clusteringverfahren wird ein mehrdimensionaler Datensatz mit einer bestimmten Dimensionalität auf ein Gitter mit wenig Dimensionalität meist ein - oder zweidimensional projiziert. Die Anzahl der Cluster ist festgelegt und kann durch die Spalten und Zeilen im Gitter festgelegt werden. Die Referenzvektoren werden beim SOM Clustering aus den Knoten im Gitter gebildet und durch eine iterative Annäherung anhand des vorgegebenen Algorithmus zu den Eingabevektoren geleitet. [13]

#### Algorithmus [13]:

1. Wähle ein Gitter mit  $k = k_v \times k_h$  Knoten  $v, v \in \{1, \dots, k\}$ .
2. Initialisiere  $k$   $d$ -dimensionale Vektoren  $f_0(v)$ , durch zufällige Wahl von Objekten  $x \in S$  oder vollständig zufällig
3. Iteration  $i$ :
  - (a) Für jedes Objekt  $x \in S$  bestimme den Knoten  $v_x$ , für den  $f_i(v_x)$  am nächsten zu  $x$  liegt.
  - (b) Aktualisiere alle Referenzvektoren wie folgt:

$$f_{i+1} = f_i(v) + \eta(d(v, x), i) \cdot (x - f_i(v))$$

$\eta(d, i)$ : Lernrate; Die Lernrate nimmt mit der Distanz zwischen den Knoten und der Iteration ab.

- (c) Wiederhole (a) und (b) bis keine Veränderung mehr eintritt.

### Zusammenfassung

Bei dem SOM-Algorithmus ist die Reduktion der Dimensionalität an erster Stelle und kann somit sehr effizient zum Berechnen von großen Datensätzen herangezogen werden, da die Dimensionalität bzw. die Anzahl der Merkmale gering ist. Auch eignet sich der Algorithmus um Daten aufzuteilen, da die Clusteranzahl vorbestimmt ist und so ein vereinfachter Algorithmus angewendet werden kann.

#### 2.3.5 Graph-based Clustering

Beim *Graph-based Clustering* wird von einem Graphen ausgegangen, welcher die Distanzmatrix repräsentiert. Die Knoten im Graphen werden Objekten aus der Datenmenge zugeordnet. Die verbindenden Linien oder auch Kanten entsprechen der Distanz zwischen den einzelnen Objekten. Diese können *gerichtet* oder *ungerichtet* sein.[4, 7]

Das Verfahren versucht den Graphen zu zerteilen und ist auch in der Heuristik als Graphpartitionsproblem bekannt. Meistens wird eine rekursive Bipartitionierung angenommen, da diese sehr effizient zu berechnen ist. Dabei sind die Graphen eine repräsentative Darstellung von Ähnlichkeitsbeziehungen der einzelnen Objekte.[7]

Eine weit verbreitete Methode beim Graph-based Clustering ist das *Clique-based Clustering*. Dabei wird durch die Cliquengraphen die Beziehungen zwischen Objekten und deren Ähnlichkeit gezeigt. Im Idealfall sind die Objekte in einen Cluster sehr ähnlich zueinander und die Objekte die zu anderen Clustern gehören sind unähnlich zueinander. Dabei sind die Knoten im Graph die Objekte und die Cliques die Cluster. Die Kanten symbolisieren dass die Elemente ähnlich zueinander sind.[4, 7]

Doch in der Praxis können Ähnlichkeitsbeziehungen nur bedingt durch Cliquengraphen dargestellt werden, da Kanten fehlen oder Kanten mehrfach vorhanden sein können. Beim *corrupted clique graph Model* stellen die Kanten die Wahrscheinlichkeiten dar und sind gewichtet. Es wird versucht vom *corrupted* zum originalen Graphen zu gelangen, welcher die richtigen Cluster repräsentiert.

[4]

#### 2.3.6 Spectral Clustering

##### Beschreibung:

Das *Spectral Clustering* ist ein Partitionierungsalgorithmus, welcher die Eigenvektoren der einzelnen Cluster verwendet und dann damit eine Beziehung zu anderen Clustern erstellt. Bei diesen Verfahren wird die Ähnlichkeitsmatrix herangezogen und die Anzahl der Cluster kann vom Benutzer festgelegt werden.[11]

**Algorithmus[5]**

Gegeben ist die Datenmenge  $S = \{x_1, \dots, x_n\}$  im  $\mathbb{R}^d$ ;  $k$ : Clusteranzahl

1. Berechne die Ähnlichkeitsmatrix  $A_{n \times n}$

$$A_{ij} = \begin{cases} e^{-\frac{|x_i - x_j|^2}{2\sigma^2}} & \text{falls } i \neq j \\ 0 & \text{sonst} \end{cases}$$

$\sigma^2$  : Skalierungsfaktor

2. Berechne die Diagonalmatrix  $D_{n \times n}$

$$D_{ij} = \begin{cases} \sum_{l=1}^n A_{il} & \text{falls } i = j \\ 0 & \text{sonst} \end{cases}$$

In der Diagonale  $D$  stehen die Zeilensummen von  $A$   
Berechne  $L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$

$$D^{-\frac{1}{2}} = \begin{cases} \frac{1}{\sqrt{D_{ii}}} & \text{falls } i = j \\ 0 & \text{sonst} \end{cases}$$

3. Finde  $v_1, \dots, v_k$ , die  $k$  größten Eigenvektoren von  $L$ , so dass alle  $v_i$  paarweise orthogonal sind. Erstelle daraus eine Matrix

$$X_{n \times k} = [v_1, \dots, v_k] \in \mathbb{R}^{n \times k}$$

4. Konstruiere Matrix  $Y_{n \times k}$  durch Normalisierung von  $X$

$$Y_{ij} = \frac{X_{ij}}{\sqrt{\sum_j X_{ij}^2}}$$

5. Jede Zeile  $Y_i$  von  $Y$  ist ein Punkt  $\mathbb{R}^k$  im Clustere diese Punkte mit einem beliebigen Clusteralgorithmus
6. Weise jedem Originalpunkt  $x_i$  den Cluster  $j$  genau dann zu, wenn die Zeile  $Y_i$  im Cluster  $j$  liegt.

Beim spektralen Clustering ist die Form des Cluster nicht so bedeutsam, da die Form von den eingegebenen Daten abhängt und es eine recht einfache Implementierung mit verschiedenen Sprachen gibt. Die Clusteranzahl muss vorher gewählt werden und dies kann sich unter Umständen schwierig gestalten.[4, 9, 1]

## 2.4 Anwendungsbereiche

Das Clustering hat viele Anwendungsbereiche, da in der Informatik und Statistik große Dateien und Datensätze vorkommen. Clustering und Klassifikation sind vor allen in der Heuristik sehr bedeutsam. Auch in der Medizin spielt Clustering eine wichtige Rolle, da in der Medizin Daten von Patienten in Klassen eingeteilt werden wie z.B. AML/ALL Klassifikation der Krebsmerkmale. Auch können in der Biologie große Datenmengen anfallen. Diese müssen aufgeteilt werden und in Gruppen eingeteilt werden wie z.B. die Zuordnung von Primer an der DNA/RNA. Wie schon oben beschrieben spielt das Clustering als ein Verfahren für die Klassifikation in vielen Anwendungsbereichen eine bedeutende Rolle und wird auch gerne als Hilfsmittel für diverse Berechnungen herangezogen. Das Clustering wird häufig in Verbindung mit der Klassifikation eingesetzt und damit wird noch eine breitere Anwendung ermöglicht. [5]

## 2.5 Visualisierung

Um Resultate und Ergebnisse betrachten und zu analysieren, müssen die Daten und deren zugehörigen Ergebnisse darstellt bzw. visualisiert werden. Dazu werden Diagramme, Graphen oder repräsentative graphische Darstellungen verwendet. Dadurch können repräsentative Ergebnisse dargestellt werden. Es wird für eine Reihe von Ergebnissen (z.B. eine Messreihe) eine zwei- oder dreidimensionale Darstellungsmethode gewählt, welche Heatmaps und Fitnesslandschaften darstellen. Dabei können die Beziehungen und Zusammenhänge gut dargestellt werden. Bei der Auswertung der Daten können hierarchische Daten bzw. Ergebnisse entstehen. Hier sollte eine andere Darstellungsform gewählt werden, welche die hierarchische Ordnung der Daten berücksichtigt. Diese Diagramme stellen Dendrogramme dar. Dabei wird die hierarchische Ordnung als Baumstruktur verwendet. Damit kann der Verlauf von einzelnen Clusteringschritten nachverfolgt werden.

## 2.6 Zusammenfassung und Ausblick

Das Clustering, welches hier dargestellt wurde, ist eine Methode der unüberwachten Klassifikation. Die verschiedenen Algorithmen helfen große und komplexe Datenmengen besser zu analysieren und erleichtern die nachträgliche Verarbeitung. Diese müssen interpretiert werden wobei die verschiedenen Visualisierungsmöglichkeiten hilfreich sind. Wie gut das Clustering interpretiert werden kann hängt sehr stark von den gewählten Parametern bzw. vom gewählten Algorithmus ab. Die verwendeten Daten entscheiden über die Art des gewählten Algorithmus, da bei manchen Datensätzen Algorithmen keine Ergebnisse liefern, da es für jede Methode Voraussetzungen gibt. So ist es wichtig zuerst abzuklären, welche Voraussetzungen gegeben sind. Dann sollte der richtige Algorithmus ausgewählt werden und nicht umgekehrt. Diese Algorithmen würden bessere Ergebnisse liefern, wenn die Daten besser angepasst wären, da es bei der Effizienz und Performance meist nur auf die Beschaffenheit der Daten ankommt. Die richtige Wahl der Parameter ist die größte Herausforderung beim Clustering und auch bei den heuristischen Algorithmen und Verfahren. Da häufig kein Wissen über das Verhalten von den Daten *a priori* bekannt ist, ist eine Vorhersage nur schwer möglich. In Zukunft wird die Entwicklung in Richtung selbst adaptiver Clusteringalgorithmen gehen, welche sich anhand der Daten anpassen und nicht mehr auf Annahmen basieren, welche meist nur sehr schlecht performante Ergebnisse liefern. Auch muss hier die Effizienz per Datensatz gesteigert werden um zeitlich bessere Ergebnisse zu erhalten.

# Kapitel 3

## Klassifikation

### 3.1 Einführung

Bei der Klassifikation werden große Datensätze in Klassen eingeteilt und können durch die Anwendung von Clustering weiterverarbeitet werden. Diese Methoden erlauben es komplexe Daten zu verarbeiten. Durch diese Schritt wird eine Erweiterung der Anwendungsbereiche ermöglicht, da bereits eingeteilte Daten einfacher zu bearbeiten und zu analysieren sind. Die Klassifikation ist ein Teil von Data Mining und Heuristik. Im Gegensatz zum Clustering ist die Klassifikation eine Methode, welche anhand vorgegebener Trainingsdaten die Auswahl des richtigen Algorithmus zu erleichtern. Die Zuordnung erfolgt dabei händisch. Damit lassen sich unbekannte Daten mit bestimmten Testdaten und Merkmalen eindeutig in Klassen einteilen. Dabei sind die Klassen und Trainingsdaten vorher bekannt und dies wird auch als überwachtes Lernen bezeichnet. [9]

Bevor mit der Klassifikation begonnen werden kann, müssen die Voraussetzungen definiert werden. Dabei wird von einer bestimmten Menge von Trainingsdaten ausgegangenen, welche bestimmte Merkmale bzw. Attribute aufweisen. Hier wird ein Klassentribut, welches die eindeutige Zuordnung zu einer bestimmten Klasse oder Gruppe besitzt vorgegeben. Das Attribut für die Zuordnung ist immer qualitativ, die restlichen Merkmale können auch quantitativ sein.[9]

Diese Verfahren laufen in zwei Phasen ab. Dabei wird in der ersten Phase anhand dem Vorliegen der Daten (Trainingsdaten) ein Klassenmodell aufgebaut. Dieses Modell wird in der zweiten Phase zur Zuordnung von den Daten angewandt. Das Klassenattribut ist an sich nicht bekannt um auch diese Daten in Klassen einzuteilen. Ziel der Klassifikation ist es anhand von vorgegeben Modellen Daten zuzuordnen. [9, 6]

## 3.2 Klassifikationsgüte

### 3.2.1 Beschreibung

Bei der Klassifikation ist das Einschätzen der Gütefunktion einfacher als beim Clustering. Da die Klassifikation die Objekte eindeutig zuordnen kann ist es möglich die *wahre Fehlerrate(true error rate)* zu berechnen und damit den Anteil der falsch klassifizierten Objekte zu bestimmen. Die textuelle mathematische Formel lautet:

$$\text{true error rate} = \frac{\text{Anzahl der falsch klassifizierten Objekte}}{\text{Anzahl aller Objekte}}$$

Doch wenn sich unter den Daten unbekannte Objekte befinden gibt es keine Methode die wahre Fehlerrate zu berechnen. Da keine Informationen über die Klassen vorhanden sind müssen andere Methoden gewählt werden, um eine etwaige Klassifikation zu bestimmen. Nur für die Trainingsdaten kann die wahre Fehlerrate *a priori* bestimmt werden, da diese vor der Berechnung die Klassenzugehörigkeit bekannt ist. Die Fehlerrate für die Trainingsdaten wird *offensichtliche Fehlerrate (apparent error rate)* genannt und lässt sich durch die folgende Formel beschreiben:[9]

$$\text{apparent error rate} = \frac{\text{Anzahl der falsch klassifizierten Trainingsobjekte}}{\text{Anzahl aller Trainingsobjekte}}$$

In der Statistik wird häufig beschrieben, dass sich die offensichtliche Fehlerrate der wahren Fehlerrate annähert. Wenn genügend Trainingsdaten vorhanden sind kann die wahre Fehlerrate mit der offensichtlichen Fehlerrate gleich gesetzt werden und so mit die Fitness bzw. Gesundheit der realen Daten bestimmt werden. Bei der Betrachtung von realen Problemstellungen ist die Anzahl der Trainingsdaten kleiner und daher müssen Varianten und Verfahren gesucht werden, welche die wahre Fehlerrate annähernd berechnen können.[15]

### 3.2.2 Train und Test

Die einfachste Methode ist die Eingabedaten in zwei Teile zu teilen und den einen Teil als Trainingsdaten und den anderen Teil als Testmenge zu verwenden. Die Trainingsmenge wird angewandt um den Klassifikationsalgorithmus die vorgegeben Klassen mitzuteilen und damit dann die Testdaten zu klassifizieren. Die beiden Datensätze müssen unabhängig voneinander sein, da

diese rein zufällig ausgewählt werden und damit kann die Fehlerrate recht gut angenähert werden. Die einzige Voraussetzung ist, dass die Testmenge relativ groß ist, da sonst die Klassifikation sehr schnell ungenau wird. Ansonsten muss auf andere Verfahren zurückgegriffen werden wie beispielsweise auf bestimmte Sampling Techniken. [9]

Auch andere Verfahren können die Klassifikationsgüte relativ gut aus dem Kontext des Anwendungsgebiets berechnen. Aber es ist es relativ schwer gute Ergebnisse zu erreichen und manchmal kann die Güte negativ von dem gewählten Verfahren beeinflusst werden. [15]

### 3.3 Algorithmen

#### 3.3.1 Bayes-Klassifikatoren

##### Beschreibung:

Bei der Bayes-Klassifikation wird auf die mathematische Grundlage der Wahrscheinlichkeitsberechnung der einzelnen Klassen aufgebaut. Diese folgen dem *Satz von Bayes*, welcher mit der Formel

$$P(X|Y) = \frac{P(Y|X) P(X)}{P(Y)}$$

beschrieben werden kann.

Durch diese Formel kann die Wahrscheinlichkeit, dass ein unbekanntes Objekt einer Klasse angehört, berechnet werden. Die Wahrscheinlichkeit *a posteriori* einer Hypothese X kann unter der Annahme von einer anderen Hypothese Y und anhand der Wahrscheinlichkeiten von X und Y erklärt werden. [3, 9]

##### Algorithmus

Dieser Algorithmus ist der *naive Bayes-Klassifikator*. Bei diesen Algorithmus werden unbekannte Objekte anhand deren Wahrscheinlichkeiten Klassen zugeordnet. Die Formel dazu lautet:

$C_i$ : Klasse

$x$ : Unbekanntes Objekt

$$P(C_i|x) = \frac{P(x|C_i) P(C_i)}{P(x)}$$

Das Objekt wird nur dann einer Klasse zugewiesen wenn die Wahrscheinlichkeit ( $P(C_i|x)$ ) ein Maximum darstellt. Die Wahrscheinlichkeit für jede einzelne Klasse ist immer gleich. Dadurch muss nur der Zähler im Bruch maximiert werden und dadurch entsteht eine neue Regel welche lautet:

$$\arg \max_{C_i \in \{C_1, \dots, C_k\}} P(x|C_i) P(C_i)$$

Die Wahrscheinlichkeit der einzelnen Klassen kann anhand der Trainingsdaten geschätzt werden. Mit Hilfe nachfolgender Formel kann die Anzahl der Trainingsobjekte pro Klasse bestimmt werden:

$$P(C_i) = \frac{|\{o \in T | o \in C_i\}|}{|T|}$$

Um die Wahrscheinlichkeit der Zuordnung zu den Klassen schätzen zu können muss die Annahme getroffen werden, dass der *naiver Bayes-Klassifikator* die Attribute der einzelnen Objekte als unabhängig betrachtet. Das bedeutet dass sich die Merkmale bzw. die Eigenschaften nicht gegenseitig behindern. Daher lässt sich die Klassenspezifische Wahrscheinlichkeit wie folgt berechnen:

$$P(x_j|C_i) = \prod_{j=1}^d P(x_j|C_i)$$

Dabei lässt sich die wahre Wahrscheinlichkeit der Objekte mit Hilfe der Trainingsdaten berechnen bzw. abschätzen.

$$P(x_j|C_i) = \frac{|\{y \in T | y \in C_i \wedge y_j = x_j\}|}{|\{y \in T | y \in C_i\}|}$$

[3, 9]

### Zusammenfassung

Der *naive Bayes-Klassifikator* geht grundsätzlich von dem Satz von Bayes aus und verwendet diesen Algorithmus um Objekte einer Klasse zuzuordnen. Er beruht auf dem Prinzip der Unabhängigkeit von Eigenschaften, da sonst

die Wahrscheinlichkeiten nicht vollständig aus den Testdaten berechnet werden können. Auch ist anzumerken, dass dieser Algorithmus nicht effizient im Vergleich zur Anzahl der Trainingsdaten fungiert. [3, 9]

### 3.3.2 Entscheidungsbäume

#### Beschreibung

Bei den Entscheidungsbäumen läuft die Klassifikation aufgrund von Einteilungen der Objekte anhand von Baumstrukturen ab und so kann eine hierarchische Klassenordnung erzeugt werden. Die einzelnen Knoten repräsentieren die Klassen, welche zu Beginn leer sind oder enthalten diverse Tests, die einem bestimmten Attribut zugeordnet sind. Doch in der Praxis sind die Attribute der einzelnen Objekte nicht eindeutig zuordenbar und so kann es vorkommen, dass manche Objekte mehrfach in verschiedenen Klassen vorhanden sind. Um ein Objekt mit dieser Methode zu klassifizieren, wird bei der Wurzel des Baumes begonnen und jedes Attribut pro Schritt durchgegangen, bis die gesamten Attribute in Klassen eingeteilt sind. Dies wird wiederholt bis die Objekte mit deren Attribute den richtigen Blättern zugewiesen sind.[3, 9]

#### Algorithmus

Es gibt nicht nur einen Algorithmus für Entscheidungsbäume sondern nur eine Richtlinie wie so ein solcher Algorithmus auszusehen hat und folgt meistens einem generalisierten Schema:

1. Es wird definiert, dass die Wurzel (hier als  $K$  bezeichnet) und der dazugehörige Baum (bezeichnet als  $B$ ) mit der Menge der Trainingsdaten (hier  $T$ ) die Ausgangssituation bildet.
2. Es wird das Attribut (hier  $A_i$ ) einem Test zugeordnet, welcher die Testmenge am besten in Objekte aufteilt und daraus die Teilmengen generiert (hier  $T_1, \dots, T_m$ ).
3. Die ganze Testmenge wird nach dem ausgewählten Test auf die Teilmengen hier  $T_j$  aufspaltet und daraus ein Knoten ( $K_j$ ) als Unterordnung vom Wurzelknoten generiert.
4. Für alle abhängigen Knoten, welche alle derselben Klasse angehören, wird ein Blatt im Baum erzeugt. Andernfalls wird weiter rekursiv durch den Baum gegangen und weiter aufgeteilt bis keine Zuordnung mehr möglich ist

Die Aufspaltung in die Teilmengen geschieht nach dem Prinzip eines Tests, welcher am besten die Testdaten aufspaltet. Damit stellt sich die Frage wie die Qualität von einem solchen Test bewertet werden kann. Der erste Ansatz ist das Prinzip der Reinheit der Daten; das bedeutet, dass die Teilmengen nur Objekte von einer Klasse beinhalten. Ein gutes Maß ist die Entropie, die angibt wie groß die Unordnung in einer Menge ist.[3, 9]

Die Formel lautet:

$$\text{entropie}(T) = - \sum_{i=1}^k p_i \log_2 p_i$$

$p_i$  ist die Wahrscheinlichkeit mit der ein Objekt, welches einer Teilmenge angehört in einer Klasse vorhanden ist. Diese Wahrscheinlichkeit lässt sich anhand der Trainingsdaten abschätzen:

$$p_i = \frac{|\{x \in T | x \in C_i\}|}{|T|}$$

Mit Hilfe der Berechnung der Entropie kann nun ein bedeutenderes Maß berechnet werden, welches sich *Informationsgewinn (gain)* nennt und die Abnahme der Entropie während eines ganzen Teilungsschrittes beschreibt. Die Formel dafür lautet:

$$\text{gain}(T, A) = \text{entropie}(T) - \sum_{i=1}^k \frac{|T_j|}{|T|} \text{entropie}(T_j)$$

Durch die Berechnung von dem *gain* kann nun bestimmt werden wie der Algorithmus die Testdaten aufteilt. Durch Verfeinerung der Methode (*gain*) kann durch die *gain ratio* eine Kombination aus *split info* und dem *gain* wie folgt mittels Formeln beschrieben werden:

1.

$$\text{split info}(T, A) = - \sum_{i=1}^k \frac{|T_j|}{|T|} \log_2 \frac{|T_j|}{|T|}$$

2.

$$\text{gain ratio}(T, A) = \frac{\text{gain}(T, A)}{\text{split info}(T, A)}$$

Um gewisse begünstigte Aufteilungen zu verbieten wird für die *gain ratio* ein Schwellenwert festgelegt. Es existiert neben der *gain ratio* auch ein weiteres Maß, der *Gini-Index*. Dieses Maß ist einfach zu bestimmen und liefert dennoch gut vergleichbare Ergebnisse. Dieser Index lässt sich nach einer einfachen Formel bestimmen:

$$gini(T) = 1 - \sum_{i=1}^k p_i^2$$

Mit Hilfe dieser Formel lässt sich der *Gini-Index* für die Gesamtheit der Klassen berechnen. Die einzelnen Partitionen stellen die  $T_n$  dar. Hilfreich ist diesbezüglich folgende Formel:

$$gini(T_1, \dots, T_m) = \sum_{i=j}^m \frac{|T_j|}{T} gini(T_i)$$

Nachdem mit den oben beschriebenen Strategien ein geeigneter Entscheidungsbaum aufgebaut worden ist, können alle Objekte anhand deren Attribute einer bestimmten Klasse zugeordnet werden. [3, 9]

### **Overfitting**

Ein Entscheidungsbaum kann mithilfe der Trainingsdaten korrekt aufgebaut werden. Aber es ist möglich, dass neue Daten nicht mehr vollständig klassifiziert werden können. Dadurch verschlechtert sich die Güte der Daten und es entsteht dann ein weniger komplexer Baum. Dieser Effekt wird dann als *Overfitting* bezeichnet. [9]

Fast alle Algorithmen implementieren eines der beiden Verfahren, mit welchen das *Overfitting* reduziert werden kann:

- Der Algorithmus wird vorher schon gestoppt, bevor ein *Overfitting* zu stande kommen kann. So wird vermieden, dass die Klassifikationsgüte zu stark verschlechtert wird. Dieser Vorgang wird als *pre-pruning* bezeichnet und ist weniger verbreitet, da es schwierig ist zu bestimmen wann der Aufbau des Entscheidungsbaums gestoppt werden muss.

- Der Entscheidungsbaum wird fertig aufgebaut und dann vereinfacht indem Knoten durch Blätter ersetzt werden. Diese Methode stellt *post-pruning* dar und ist einfacher in der Ausführung, da bereits ein Entscheidungsbaum vorliegt.

[9]

#### **Zusammenfassung:**

Die Entscheidungsbäume sind eine komfortable Methode um Daten zu klassifizieren, da deren Aufbau einfach gestaltet ist. Die meisten Algorithmen sind binär ausgeführt, da im Zusammenhang mit Entscheidungsbäumen jede Wurzel zwei Kindknoten oder Blätter besitzt. Die Aufteilung in Trainingsdaten und Testdaten erfolgt vor der eigentlichen Berechnung. Diese Aufteilung ist das Grundkonzept von Entscheidungsbäumen. Dabei werden verschiedene Methoden implementiert wie oben bereits beschrieben. Der eigentliche entscheidende Schritt ist dabei das Pruning, welches den Baum so optimiert, dass die Güte der Klassifikation ausreichend ist. Ein aussagekräftiges Kriterium ist auch die Anzahl der Attribute welche ein Objekt besitzt. Es ist dabei wichtig dass der Entscheidungsbaum mit genügend Trainingsdaten aufgebaut wird, da sonst die Güte darunter leidet. [3]

#### **3.3.3 ( $k$ )-Nächste-Nachbarn-Klassifikatoren**

##### **Beschreibung:**

Diese Methode ist auch beim Clustering bekannt und macht sich die Distanzberechnung der einzelnen Objekte zu Nutze. Durch die *Nähe* der einzelnen Nachbarn werden die einzelnen Objekte bestimmten Klassen zugeordnet. [3, 9]

##### **Algorithmus**

Die Methode beschreibt das Verfahren bei der unbekannte Objekte anhand der Distanz zu den Trainingsobjekten einer Klasse zugeordnet werden können. Mit dieser Formel lässt sich die Vorgangsweise beschreiben:

$$c(x) = c \left( \min_{y \in T} dist(x, y) \right)$$

Dabei bezieht sich das  $dist(x, y)$  auf die Euklidische Distanz, welche sich

nach folgender Formel berechnet:

$$\sqrt{\sum_{i=1}^1 (x_i - y_j)^2}$$

Eine andere Möglichkeit ist nicht nur die unmittelbaren Nachbarn sondern auch weiter entfernte Nachbarn für die Berechnung heranzuziehen und dadurch die Qualität zu steigern. Dabei wird ein unbekanntes Objekt einer Klasse zugeordnet. Die benachbarten Objekte gehören der jeweiligen Klasse an. Die Nächsten Nachbarn werden als  $y_n$  bezeichnet und können durch folgende Formel berechnet werden:

$$c(x) = \max_{C_i \in C} \sum_{j=1}^k \delta(C_i, c(y_i))$$

[3, 9]

#### **Zusammenfassung:**

Bei den ( $k$ )-Nächste-Nachbarn-Klassifikatoren werden Objekte anhand von Nachbarschaften bestimmten Klassen zugeordnet. Dabei sind die einzelnen Klassifikationsschritte unabhängig voneinander, da sich Nachbarn gegenseitig nicht beeinflussen. Die Wahl der richtigen Distanzfunktion ist entscheidend wie gut der Algorithmus arbeitet.

### **3.4 Zusammenfassung und Ausblick**

Die Klassifikation ist neben den Clustering eine wichtige Methode Objekte in Klassen einzuteilen. Bei der Klassifikation werden die Daten mithilfe verschiedener Methoden in Trainingsdaten und Testdaten aufgeteilt und anhand von den Testdaten eindeutig einer Klasse zugeordnet. Das Wichtigste bei der Klassifikation ist die richtige Wahl der Anzahl von den Trainingsdaten. Nur so ist sichergestellt, dass eine ausreichende Klassifikationsgüte erreicht wird. In Zukunft wird sich an den hier vorgestellten Prinzipien wenig ändern, da diese effizient und auch performant sind. Daher müssen in Zukunft neue noch bessere Algorithmen gefunden werden, welche noch größere und noch komplexere Daten klassifizieren können.

# Kapitel 4

## Beispiele

### 4.1 Tools

In den nachfolgenden Beispielen kam das HeuristicLab als Tool zur Anwendung, welches bei der Berechnung komplexer heuristischer Daten Anwendung findet. Diese Software wurde von der FH Oberösterreich entwickelt und steht kostenfrei zur Verfügung. Bei den Tests wurde diese Software zum Berechnen von Klassifikationen und Clustern genutzt um die verschiedenen Algorithmen zu vergleichen. Die Testdaten liegen in Format *.csv* vor und sind durch einen ; getrennt.

### 4.2 Klassifikation

#### 4.2.1 Einführung

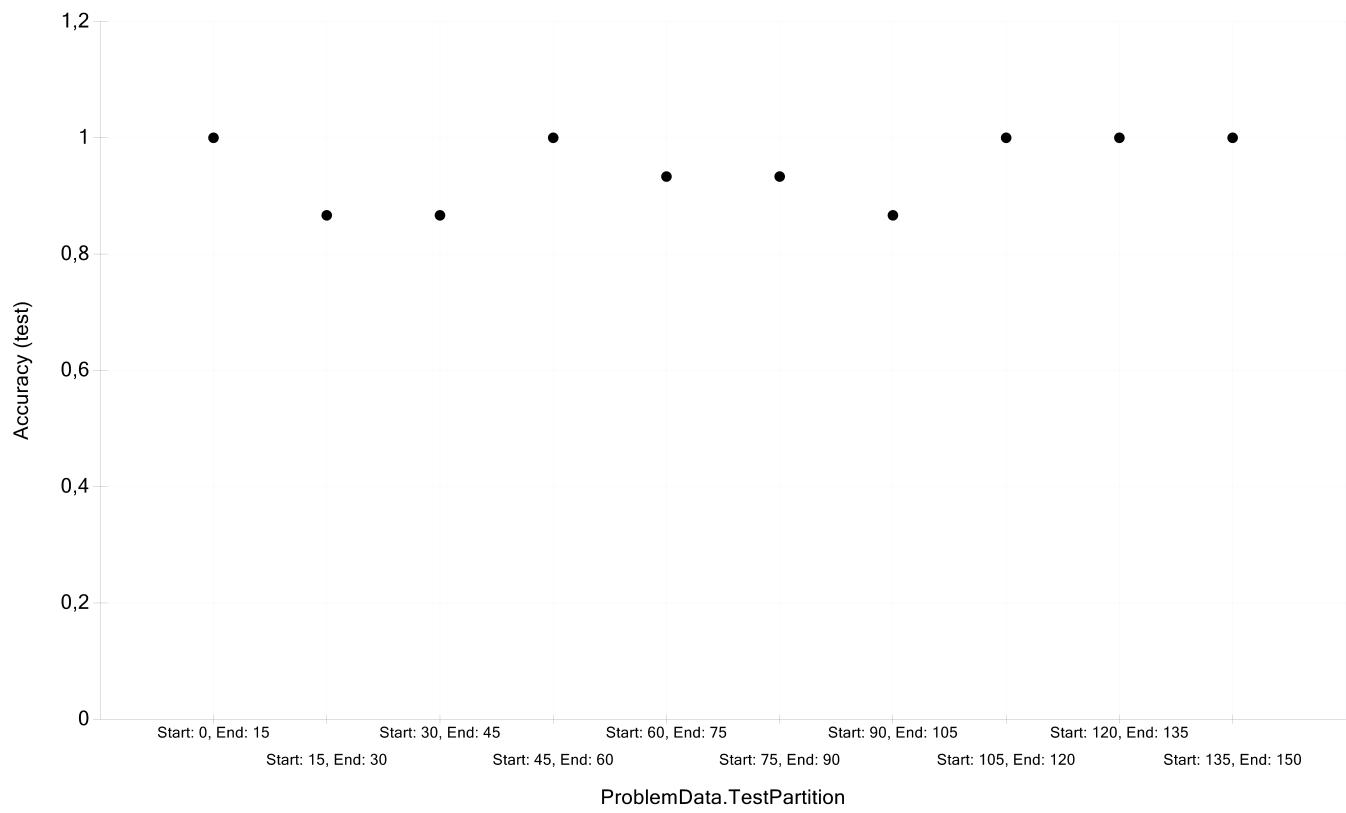
Nachfolgend wird die Klassifikation anhand mehrerer Datensätze gezeigt, dabei werden verschiedene Algorithmen verwendet. Alle Datensätze wurden vom *UCI Repository* herangezogen und beinhalten jeweils die notwendigen Daten wie der Klassenzuordnung. Die Ergebnisse stammen jeweils einen *Cross-Validation (CV)*-Durchlauf. Ausgewertet wurden die jeweiligen Genauigkeiten [14]

#### 4.2.2 Iris Datensatz

Der Iris-Datensatz ist der bekannteste Datensatz auf dem Gebiet des Data Mining und der Klassifikation. Dieser wird zum Testen von Erkennungsmerkmalen von zusammengehörigen Formen verwendet. Daher ist dieser Datensatz auch zum Testen für die Klassifikation geeignet. [14]

**Gaussian Process Least-Squares Classification**

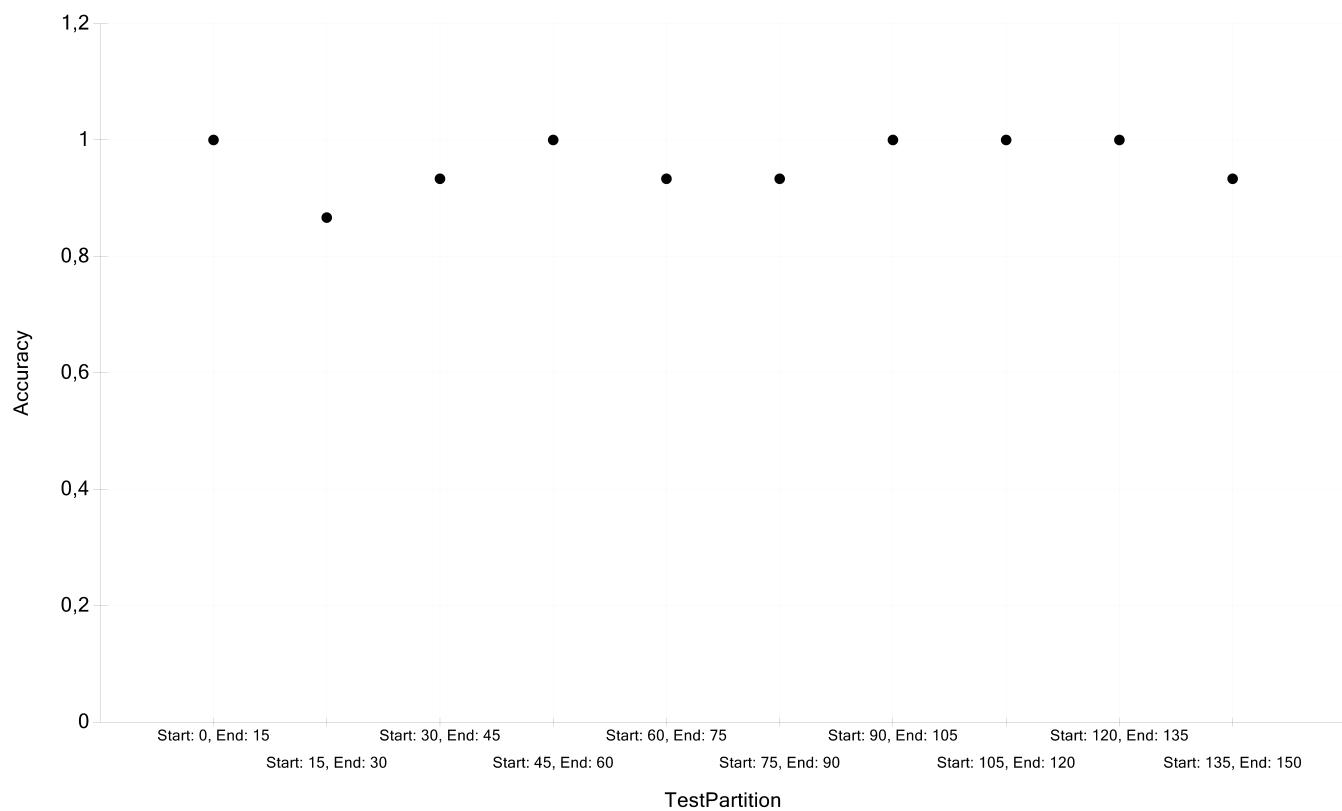
- Eingabedaten
  - Klassen: 3
  - Datengröße: 150
- Parameter
  - Folds: 10
  - Iterations: 20
  - MeanFunction: Constant
  - Seed: 0
  - SetSeedRandomly: True
- Ergebnis

**Abbildung 4.1:** Genauigkeit

- Mittelwert: 94,7%
- Standardabweichung: 6%

### Nearest Neighbour Classification

- Eingabedaten
  - Klassen: 3
  - Datengröße: 150
- Parameter
  - Folds: 10
  - K: 3
- Ergebnis

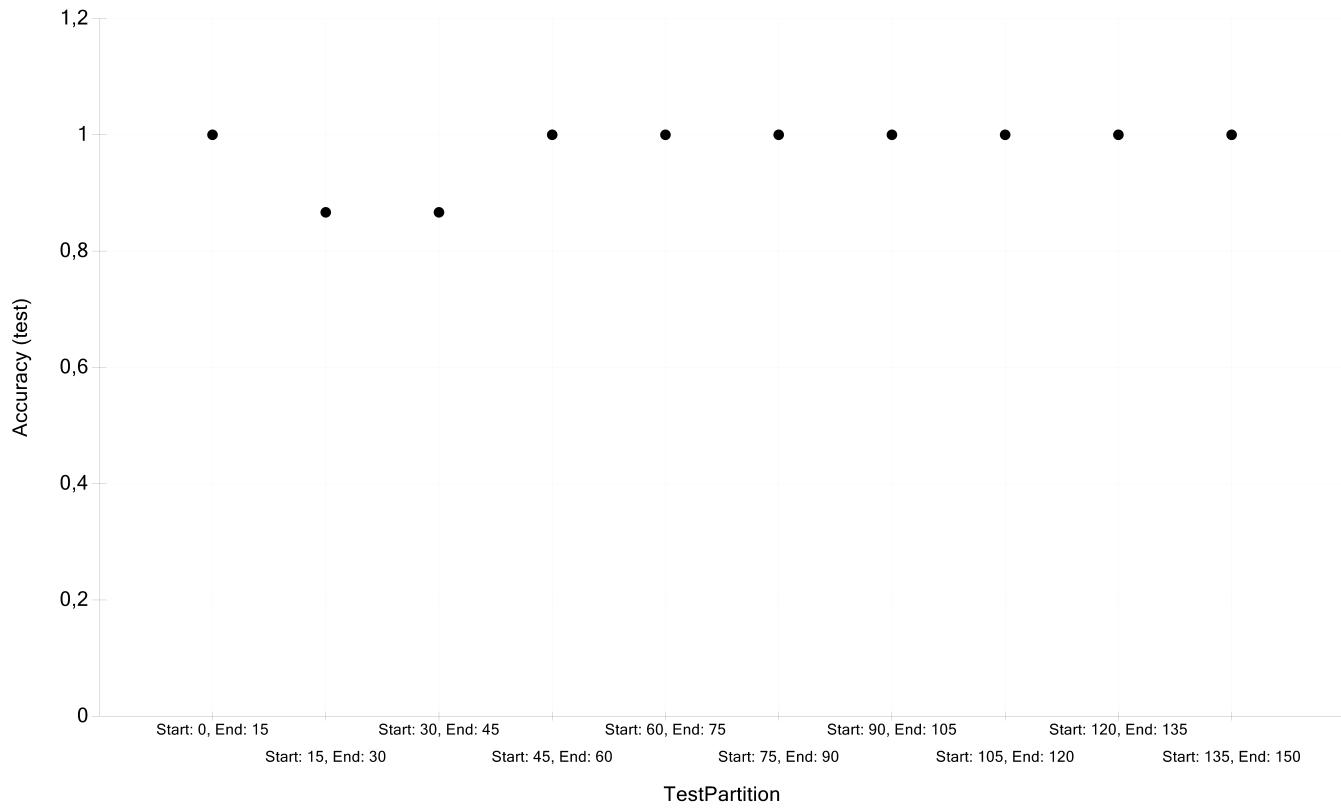


**Abbildung 4.2:** Genauigkeit

- Mittelwert: 96%
- Standardabweichung: 4,66%

### Multinomial Logit Classification

- Eingabedaten
  - Klassen: 3
  - Datengröße: 150
- Parameter
  - Folds: 10
- Ergebnis

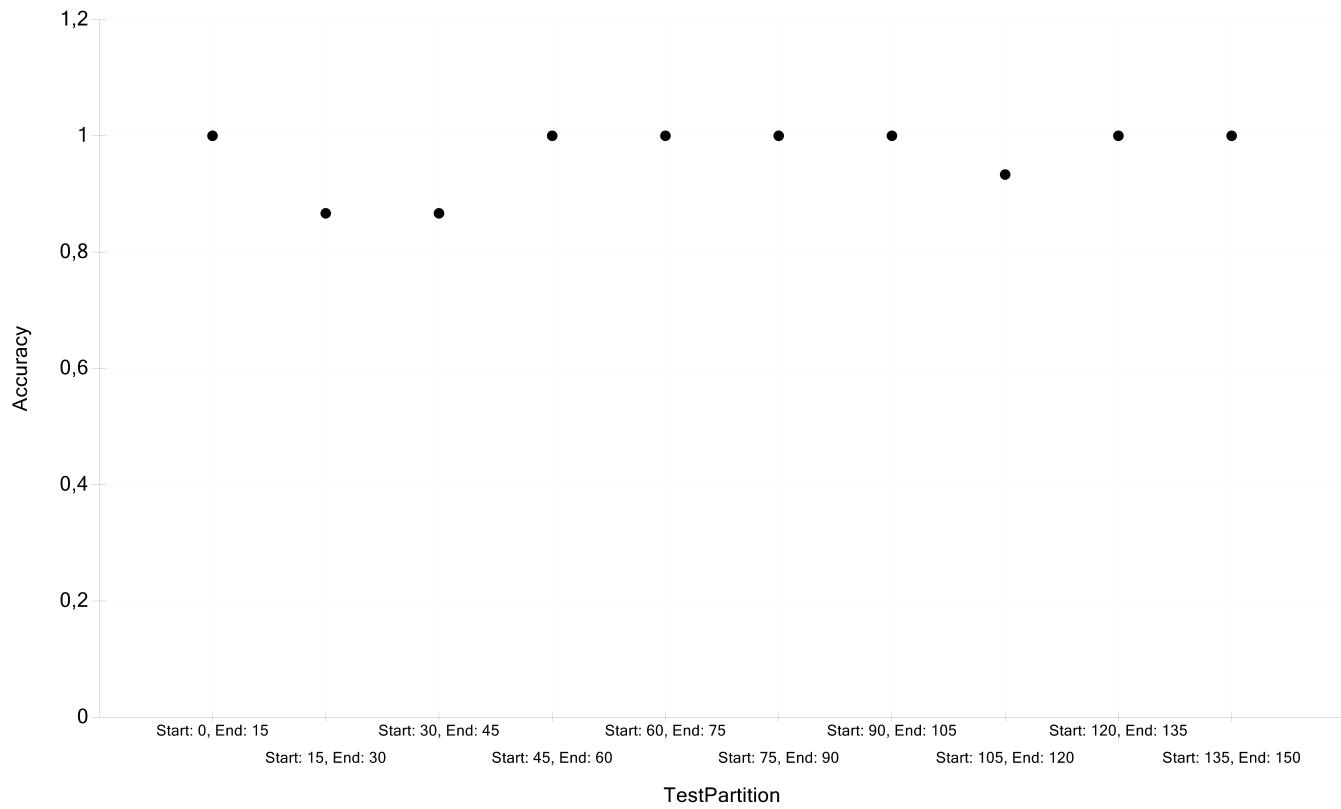


**Abbildung 4.3:** Genauigkeit

- Mittelwert: 97,4%
- Standardabweichung: 5,62%

### Neural Network Classification

- Eingabedaten
  - Klassen: 3
  - Datengröße: 150
- Parameter
  - Folds: 10
  - Decay: 1
  - HiddenLayers: 1
  - NodesInFirstHiddenLayer: 10
- Ergebnis

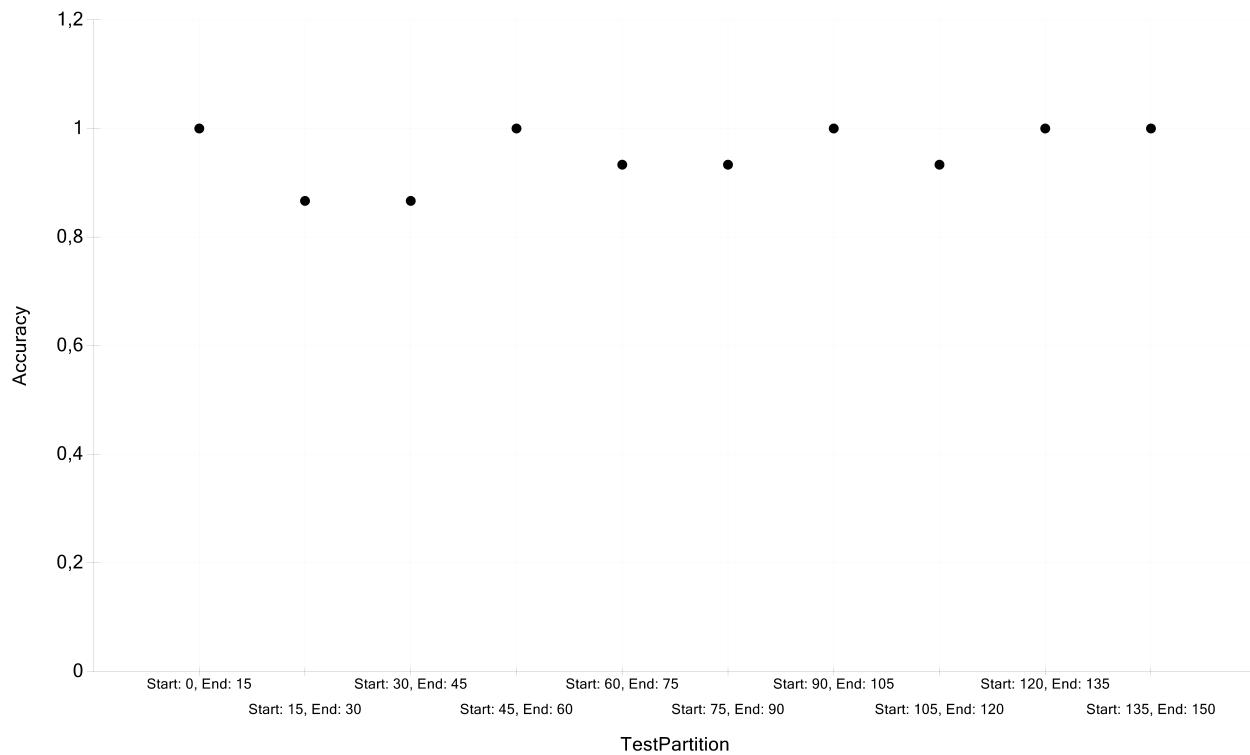


**Abbildung 4.4:** Genauigkeit

- Mittelwert: 96,7%
- Standardabweichung: 5,67%

### Random Forest Classification

- Eingabedaten
  - Klassen: 3
  - Datengröße: 150
- Parameter
  - Folds: 10
  - M: 0,5
  - Number of trees: 50
  - R: 0,3
  - Seed: 0
  - SetSeedRandomly: True
- Ergebnis

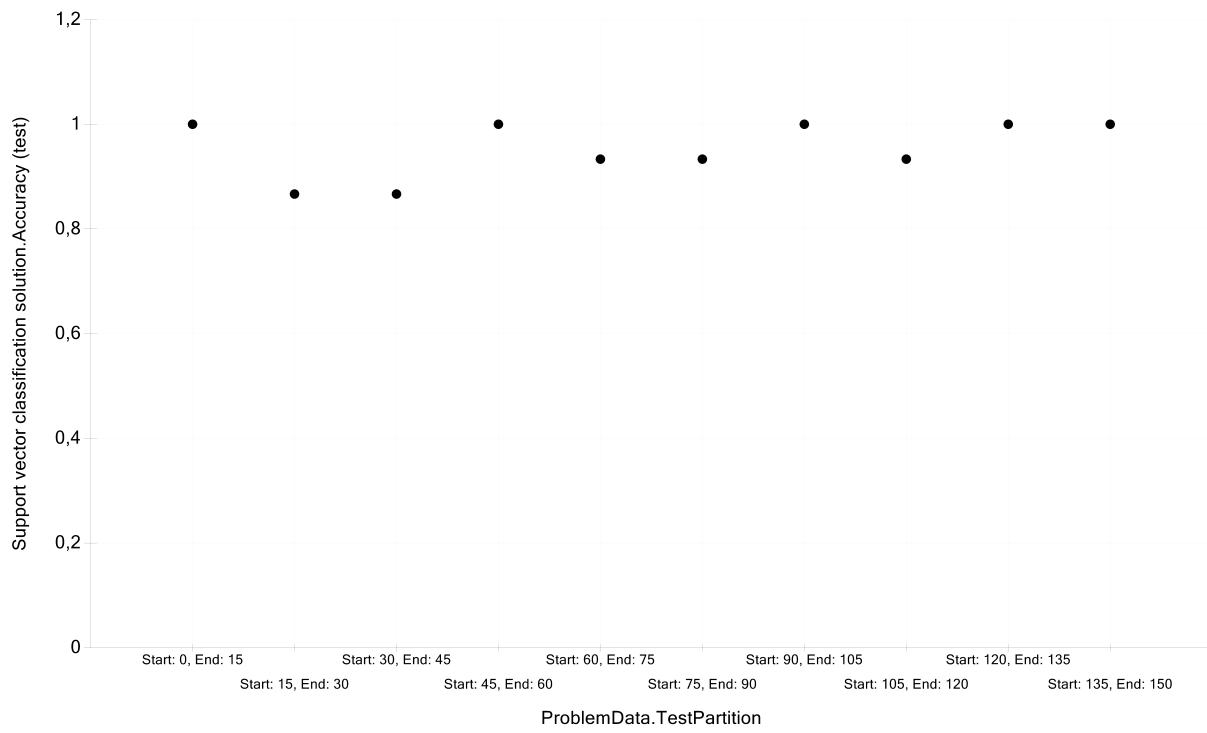


**Abbildung 4.5:** Genauigkeit

- Mittelwert: 95,3%
- Standardabweichung: 5,49%

### Support Vector Classification

- Eingabedaten
  - Klassen: 3
  - Datengröße: 150
- Parameter
  - Folds: 10
  - SvmType: NU\_SVC
  - KernelType: RBF
  - Nu: 0,5
  - Cost: 1
  - Gamma: 1
  - Degree: 3
- Ergebnis



**Abbildung 4.6:** Genauigkeit

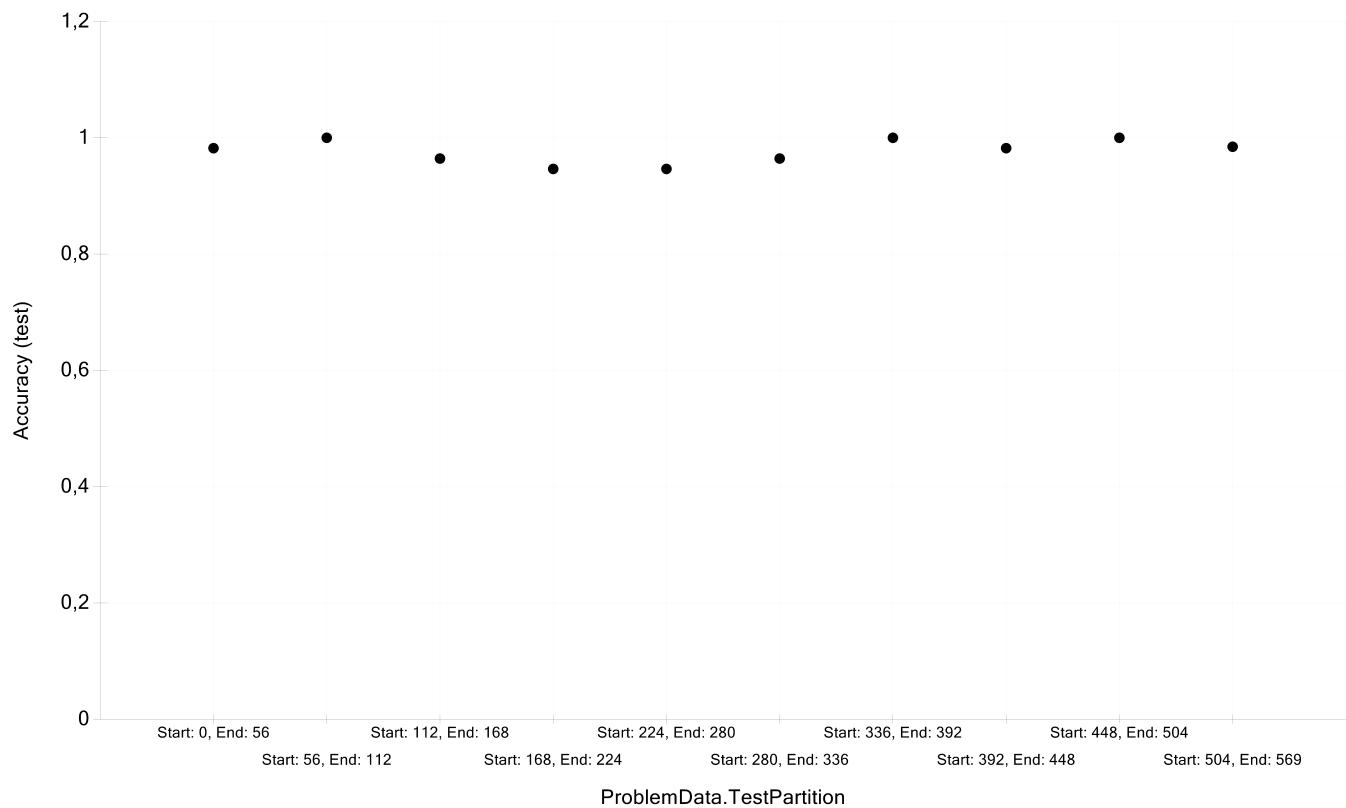
- Mittelwert: 95,3%
- Standardabweichung: 5,48%

#### 4.2.3 Wisconsin Diagnostic Breast Cancer Datensatz

Die Werte stammen aus einem digitalen Bild und beschreiben die Beschaffenheit der einzelnen Zellen im Gewebe, welche auf dem Bild zu sehen sind.  
[14]

**Gaussian Process Least-Squares Classification**

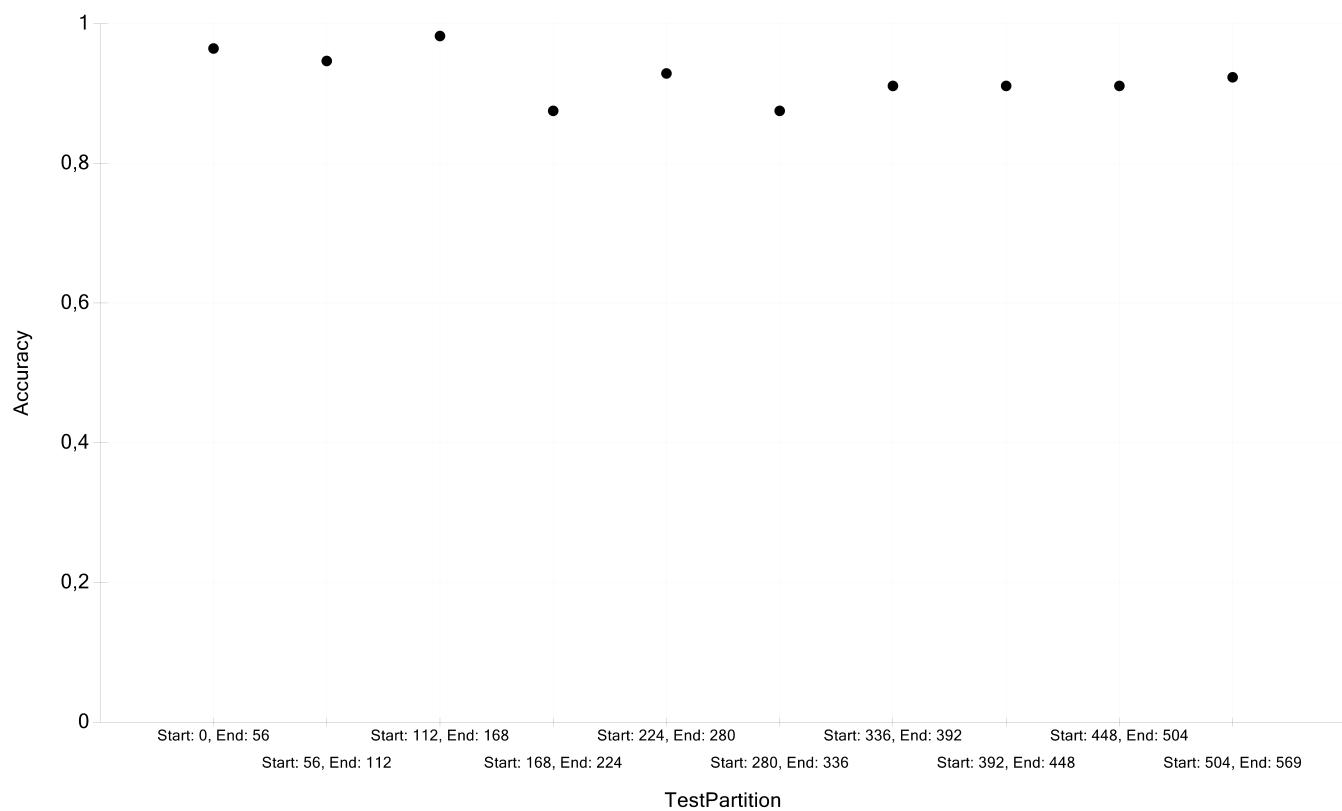
- Eingabedaten
  - Klassen: 2
  - Datengröße: 596
- Parameter
  - Folds: 10
  - Iterations: 20
  - MeanFunction: Constant
  - Seed: 0
  - SetSeedRandomly: True
- Ergebnis

**Abbildung 4.7:** Genauigkeit

- Mittelwert: 97,7%
- Standardabweichung: 2%

### Nearest Neighbour Classification

- Eingabedaten
  - Klassen: 2
  - Datengröße: 596
- Parameter
  - Folds: 10
  - K: 3
- Ergebnis

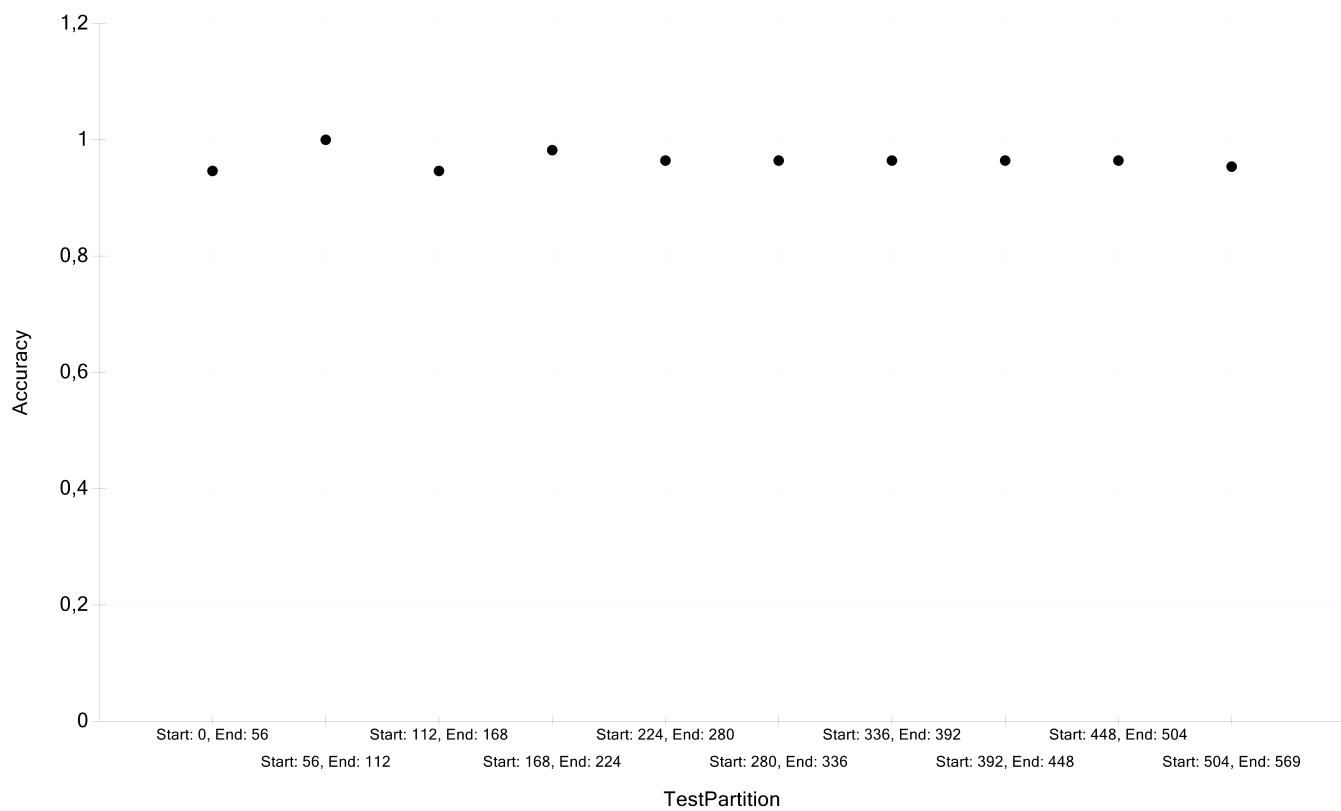


**Abbildung 4.8:** Genauigkeit

- Mittelwert: 92,27%
- Standardabweichung: 3,47%

### Multinomial Logit Classification

- Eingabedaten
  - Klassen: 2
  - Datengröße: 596
- Parameter
  - Folds: 10
- Ergebnis

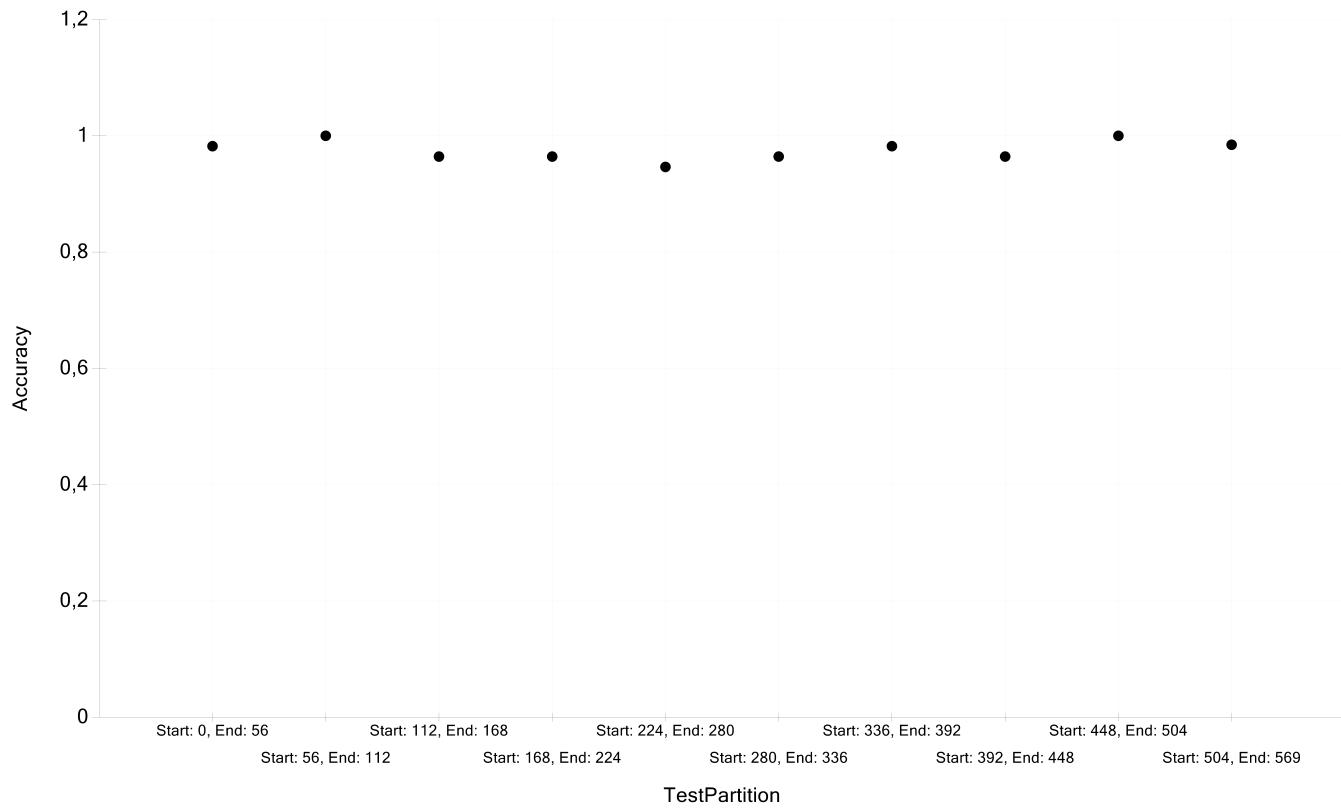


**Abbildung 4.9:** Genauigkeit

- Mittelwert: 96,5%
- Standardabweichung: 1,6%

### Neural Network Classification

- Eingabedaten
  - Klassen: 2
  - Datengröße: 596
- Parameter
  - Folds: 10
  - Decay: 1
  - HiddenLayers: 1
  - NodesInFirstHiddenLayer: 10
- Ergebnis

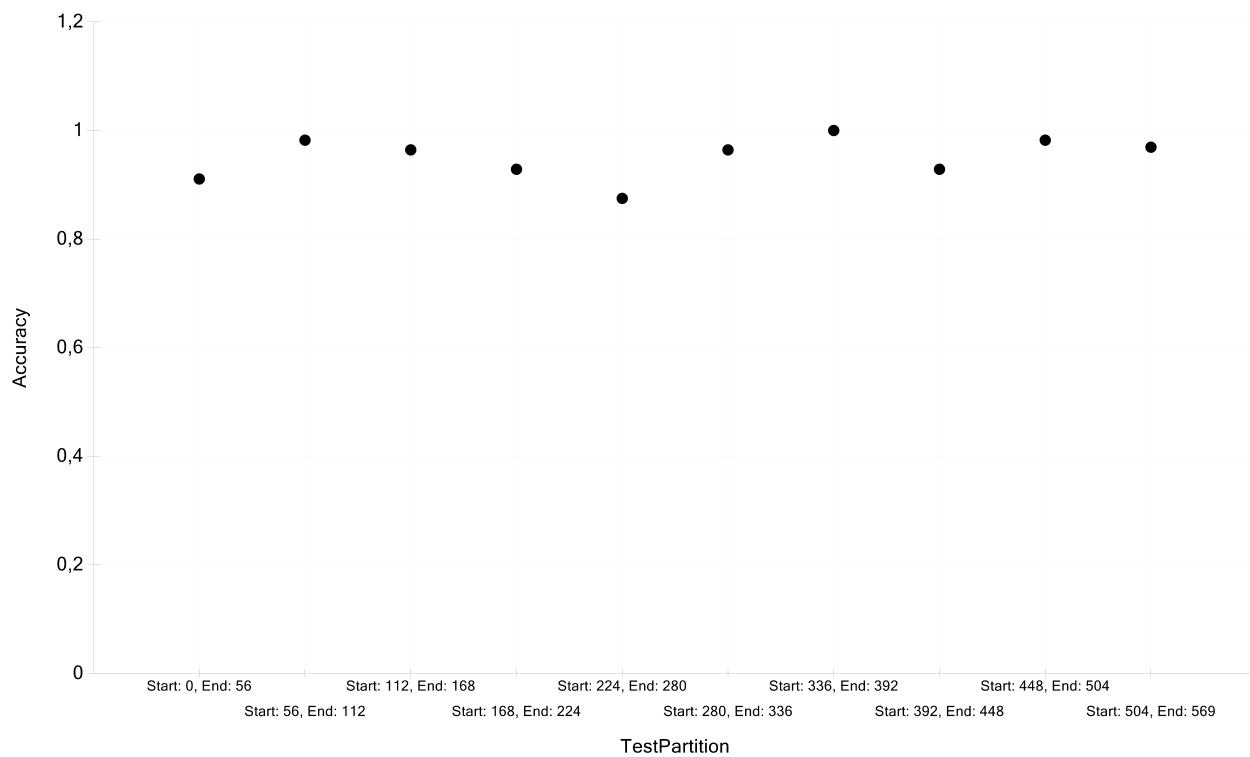


**Abbildung 4.10:** Genauigkeit

- Mittelwert: 96,5%
- Standardabweichung: 1,74%

### Random Forest Classification

- Eingabedaten
  - Klassen: 2
  - Datengröße: 596
- Parameter
  - Folds: 10
  - M: 0,5
  - Number of trees: 50
  - R: 0,3
  - Seed: 0
  - SetSeedRandomly: True
- Ergebnis

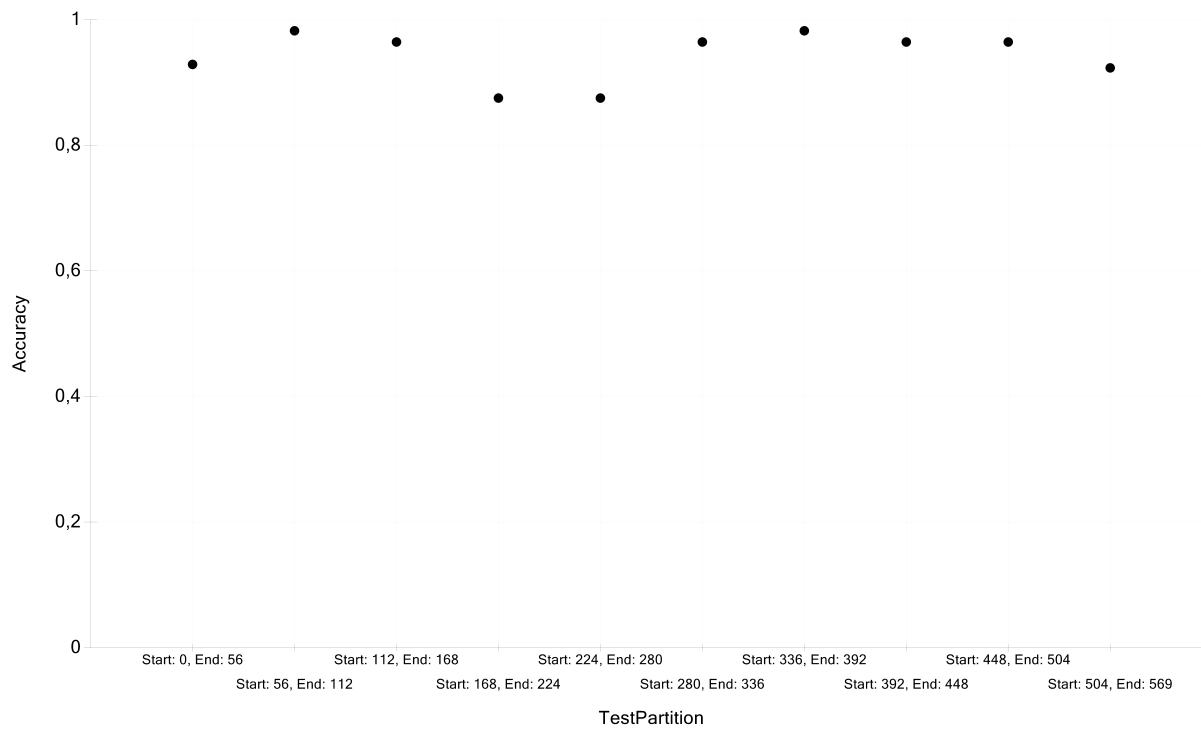


**Abbildung 4.11:** Genauigkeit

- Mittelwert: 95%
- Standardabweichung: 3,86%

### Support Vector Classification

- Eingabedaten
  - Klassen: 2
  - Datengröße: 596
- Parameter
  - Folds: 10
  - SvmType: NU\_SVC
  - KernelType: RBF
  - nu: 0,5
  - Cost: 1
  - Gamma: 1
  - Degree: 3
- Ergebnis



**Abbildung 4.12:** Genauigkeit

- Mittelwert: 94,2%
- Standardabweichung: 4,04%

#### 4.2.4 Fazit

Aus den Ergebnissen kann ableitet werden, dass es bei der Klassifikation wichtig ist, das richtige Verhältnis zwischen Trainings- und Testdaten zu finden. Dies ist stark ausschlaggebend für die Qualität der Ergebnisse. Weiters ist auch zu beachten, dass das Verhältnis zwischen Datenreihen und Klassen entspricht, da es sonst zu Misklassifikationen und dadurch zu einer Verschlechterung der Güte kommen kann. Derzeitiger Stand ist, dass es keine wirkliche Regel für das Verhältnis zwischen Klassen und Anzahl der Daten gibt. Trotzdem sollte von einem ausgeglichenen Verhältnis ausgegangen werden.

### 4.3 Clustering

#### 4.3.1 Einführung

Das Clustering wird anhand des *k-means* Algorithmus aufgezeigt. Da ein Vergleich zwischen Clustering und Klassifikation zu erfolgen hat, wurden hier dieselben Datensätze verwendet. Der Unterschied zur Klassifikation ist, dass hier die Klassendefinition weggelassen wird und stellt damit eine unüberwachte Klassifikation dar. Bei der Auswertung werden die Summen zu den einzelnen Cluster-Zentren dargestellt. [14]

### 4.3.2 Iris Datensatz

- Datengröße: 150

#### Parameter

- K: 1-10
- Wiederholungen: 0

#### Ergebnis

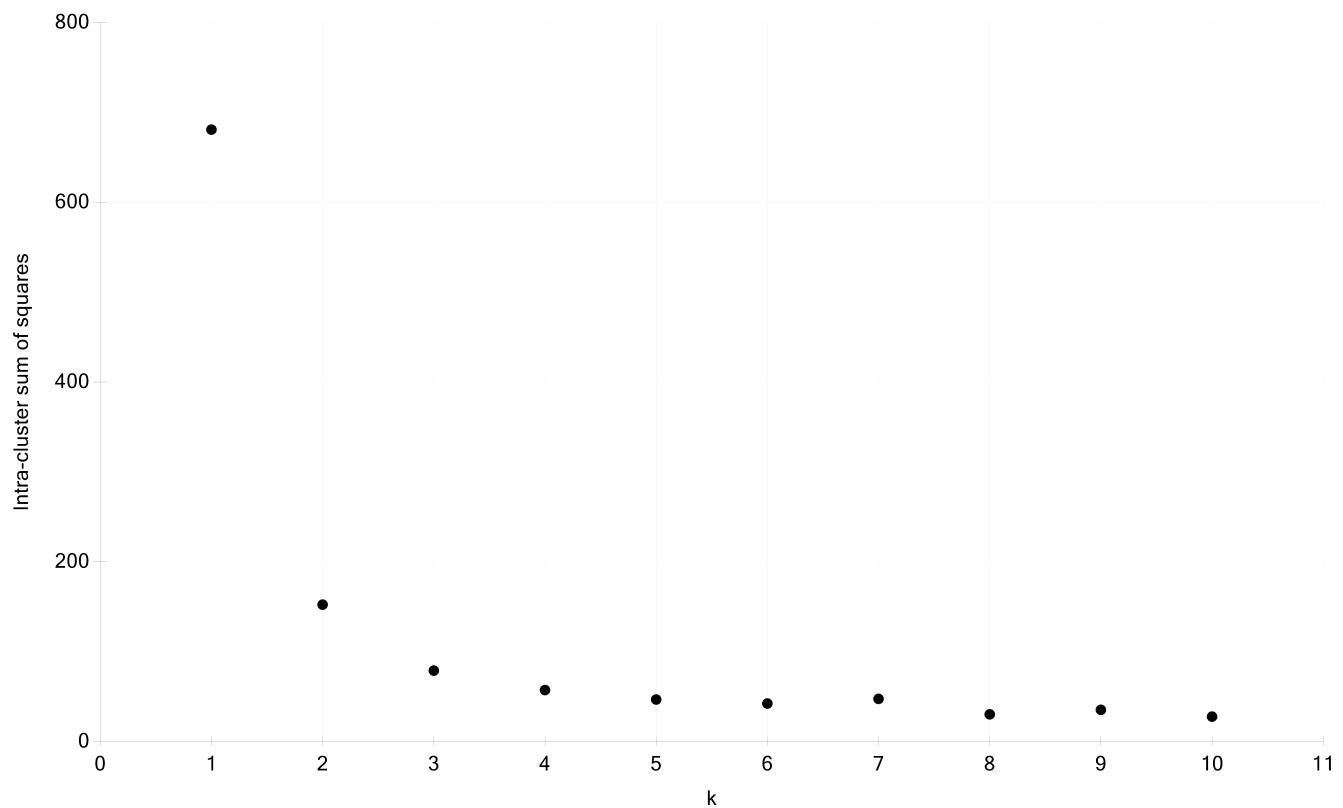


Abbildung 4.13: Abstand

- Mittelwert: 119,9475
- Standardabweichung: 200,432

#### 4.3.3 Wisconsin Diagnostic Breast Cancer Datensatz

- Datengröße: 569

##### Parameter

- K: 1-10

##### Ergebnis

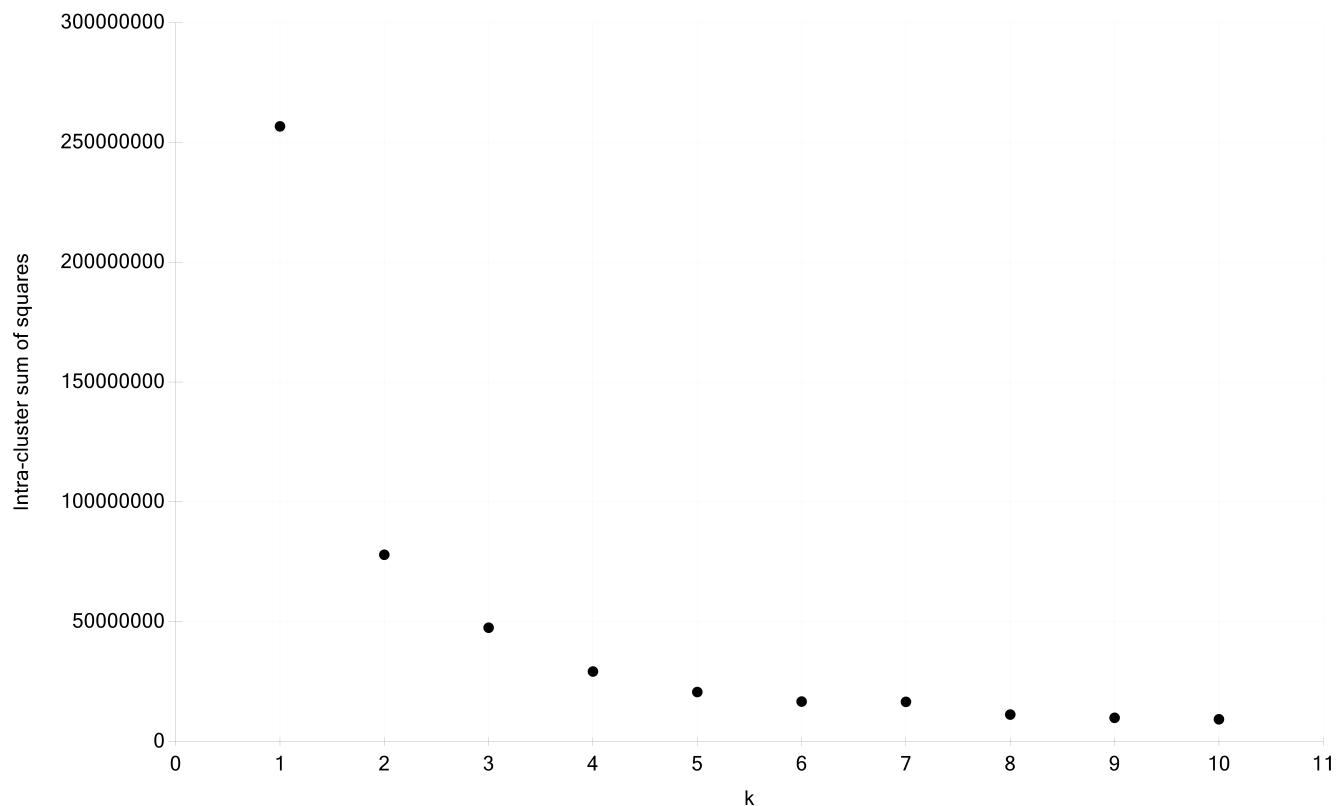


Abbildung 4.14: Abstand

- Mittelwert: 49578211,2465
- Standardabweichung: 75851854,2659

[14]

#### 4.3.4 Fazit

Beim Clustering ist zu erkennen dass es Gemeinsamkeiten mit der Klassifikation gibt da die meisten Clustereinteilungen die wahren Klassen voraussagen. Daher ist das Clustering ebenso bedeutend wie die Klassifikation. Beim Clustering kommt es auf die vorgegebene Clusteranzahl an um gute Ergebnisse zu liefern.

# Kapitel 5

## Schluss

### 5.1 Zusammenfassung

In dieser Arbeit wurden die Themen Clustering und Klassifikation behandelt. Ziel war es die verschiedenen Algorithmen aufzuzeigen und zu vergleichen. Grundsätzlich wird von einer überwachten und unüberwachten Klassifikation ausgegangen. Bei der unüberwachten Klassifikation kann auch von Clustering gesprochen werden, da es hier keine eindeutige Klasseneinteilung gibt. Im Gegensatz dazu ist bei der klassischen Klassifikation die Einteilung bzw. die Zuordnung bekannt. Doch arbeiten beide Verfahren nach demselben Prinzip. Umfangreiche Daten werden in verwendbare Teile eingeteilt. Die beiden Methoden spielen in der heutigen Welt eine bedeutende Rolle und werden häufig in der Datenanalyse sowie des Data Mining verwendet. Zur Lösung des Klassifikations- und Clusteringsproblem werden daher unterschiedliche Alogorithmen implementiert. Dabei kommt es auf die verwendeten Daten an, welcher Algorithmus die besten Ergebnisse liefert. Abschließend kann zusammengefasst werden, dass in dieser Arbeit nicht alle Algorithmen beschrieben worden sind. In dieser Arbeit wurden nur die bedeutsamsten und relevanten Algorithmen aufgelistet. Die Algorithmen werden nicht nur für Clustering und Klassifikation verwendet, sondern auch in Bereichen wo Datenanalysen und Heuristiken eine wesentliche Rolle spielen.

# Quellenverzeichnis

## Literatur

- [1] Charles J. Alpert und So-Zen Yao. „Spectral Partitioning: The More Eigenvectors, the Better“. In: *Proceedings of the 32Nd Annual ACM/IEEE Design Automation Conference*. DAC '95. ACM, 1995, S. 195–200.
- [2] Mihael Ankerst u. a. „OPTICS: Ordering Points To Identify the Clustering Structure“. In: *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*. ACM Press, 1999, S. 49–60.
- [3] Leo Breiman u. a. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, 1984.
- [4] Daniel Fasulo. *An Analysis of Recent Work on Clustering Algorithms*. 1999.
- [5] Usama Fayyad, Cory Reina und P. S. Bradley. „Initialization of Iterative Refinement Clustering Algorithms“. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. KDD'98. AAAI Press, 1998, S. 194–198.
- [6] Haitao Gan u. a. „Using clustering analysis to improve semi-supervised classification“. *Neurocomputing* 101 (2013), S. 290–298.
- [7] A. K. Jain, M. N. Murty und P. J. Flynn. „Data Clustering: A Review“. *ACM Comput. Surv.* 31.3 (Sep. 1999), S. 264–323.
- [8] Sina Khanmohammadi, Naiier Adibeig und Samaneh Shanehbandy. „An improved overlapping k-means clustering method for medical applications“. *Expert Systems with Applications* 67 (2017), S. 12–18.
- [9] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., 1997.
- [10] Muhammad Umer Munir, Muhammad Younus Javed und Shoab Ahmad Khan. „A hierarchical k-means clustering based fingerprint quality classification“. *Neurocomputing* 85 (2012), S. 62–67.

- [11] Andrew Y. Ng, Michael I. Jordan und Yair Weiss. „On Spectral Clustering: Analysis and an algorithm.“ In: *NIPS*. MIT Press, 2001, S. 849–856.
- [12] Dan Pelleg und Andrew Moore. „Accelerating Exact K-means Algorithms with Geometric Reasoning“. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '99. ACM, 1999, S. 277–281.
- [13] P. Tamayo u. a. „Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.“ *Proceedings of the National Academy of Sciences of the United States of America* (1999).
- [14] *UCI Repository*. die Datensätze stammen von deren Webseite.
- [15] Sholom M. Weiss und Casimir A. Kulikowski. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann Publishers Inc., 1991.



Fachhochschul-Bachelorstudiengang  
**MEDIZIN- UND BIOINFORMATIK**  
A-4232 Hagenberg, Austria

# **Outlook/Exchange Adapter für CGM G3 Clinical Information System MRP**

Bachelorarbeit  
Teil 2

zur Erlangung des akademischen Grades  
Bachelor of Science in Engineering

Eingereicht von

**Philipp Krainer**

Betreuer: Georg Hörlsberger, CGM Clinical, Linz  
Begutachter: Oliver Krauss MSc

Hagenberg, Dezember 2016

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>57</b>
<b>Abstract</b>	<b>58</b>
<b>1 Einleitung</b>	<b>59</b>
1.1 Zielsetzung . . . . .	59
<b>2 State of the Art</b>	<b>61</b>
2.1 EWS API . . . . .	61
2.1.1 Funktionen . . . . .	62
2.2 Multidimensionales Ressourcen Planning (MRP) . . . . .	63
2.2.1 Funktionsweise . . . . .	63
2.2.2 Masterdata . . . . .	64
2.2.3 Hospital Information System . . . . .	65
<b>3 Konzept und Design</b>	<b>66</b>
<b>4 Implementierung</b>	<b>67</b>
4.1 Microsoft Outlook . . . . .	67
4.1.1 Exchangeservice . . . . .	70
4.1.2 Model . . . . .	70
4.1.3 Kalender . . . . .	70
4.1.4 Appointments . . . . .	71
4.1.5 Availability . . . . .	71
4.1.6 Kontakt . . . . .	71
4.1.7 Email . . . . .	72
4.1.8 Filter . . . . .	72
4.2 Masterdata-Feature . . . . .	72
4.2.1 Filter . . . . .	74
4.2.2 Filterkomponenten . . . . .	74
4.2.3 Weboberfläche . . . . .	74
<b>5 Evaluierung</b>	<b>75</b>
5.1 Benutzerinteraktives Testen . . . . .	75

Inhaltsverzeichnis	56
<b>6 Zusammenfassung</b>	<b>77</b>
6.1 Resultate . . . . .	77
6.2 Diskussion . . . . .	77
<b>Quellenverzeichnis</b>	<b>78</b>
Literatur . . . . .	78

# Kurzfassung

Diese Arbeit setzt sich mit der Implementierung eines Moduls für MRP mit der EWS API auseinander. Das MRP-System generiert mit Hilfe von Resourcen und Regeln Termine, welche in das Microsoft Outlook übertragen werden. Ziel ist es die Verbindung von MRP und Microsoft Outlook herzustellen. Das Modul ist in JAVA programmiert. Mit Hilfe der EWS API wird die Verbindung zum Server geschaffen.

Bei der Implementierung wird geachtet, dass die Richtlinien von der Zielsetzung eingehalten werden. Es wird ein Konzept zur Ansteuerung mittels MRP vorgestellt. In der Implementierung werden die Funktionalitäten sowie Methoden beschrieben, die verwendet worden sind. Weiters werden in der Implementierung die EWS Funktionen angewendet und an das MRP angepasst. Die einzelnen Funktionen werden mittels der Methode des benutzerinteraktiven Testens überprüft. Die Tests wurden mehrfach ausgeführt und erfolgreich abgeschlossen.

# Abstract

This thesis deals with the implementation of a module for the MRP System using the EWS API. The MRP System generates appointments using resources and rules. These appointments get exported to Microsoft Outlook. The connection of MRP and Microsoft Outlook presents the goal of this thesis. The programming language JAVA is used to implement the functions and methods of this software. The EWS API handles the requests from the MRP System and communicates with the server.

The implementation follows the rules of the given objective of the MRP System. There is a concept for using MRP and EWS. The implementation shows used methods and functions for the implementation with EWS. The implementation contains the EWS related functions as well as the methods, which are adapted for the MRP System. The functions as described in the implementation are tested with the method called userinteractive testing. The tests were executed several times and successfully completed.

# Kapitel 1

## Einleitung

Diese Arbeit stellt den praktischen Teil der Bachelorarbeit dar. Diese wurde im Rahmen des Semesterpraktikums vom 01.03 bis 30.06.2016 erstellt. Das Praktikum im Umfang von 16 Wochen wurde bei der Firma CGM Clinical Austria am Standort Linz absolviert. Die CompuGroup Medical (CGM) ist ein österreichischer Softwarehersteller der umfassende IT-Lösungen zur Optimierung des Gesundheitswesens produziert. Diese Software-Lösungen unterstützen die Prozesse von niedergelassenen Ärzten und deren Personal sowie von medizinischen, pflegerischen und administrativen Krankenhauspersonal. Die CGM beschäftigt in Österreich ca. 250 Mitarbeiter (<https://www.cgm.com/at/index.de>).

### 1.1 Zielsetzung

Das Modul Multidimensionale Ressourcen Planning (MRP) ermittelt Termine für Operationen in Krankenhäusern unter der Berücksichtigung der beteiligten Ressourcen (Personen, Geräte, Material) und vor und nachgelagerten Untersuchungen inklusive Betten und Zimmern. Ziel dieses Projektes ist es für die ressourcenbezogenen Daten (z.B. Dienste von Ärzten , Operationstermine etc.) eine Möglichkeit zu schaffen, Termine in den persönlichen Kalendern der beteiligten Personen (Teams) anzuzeigen. Damit sollen die Anwender in ihrer gewohnten Umgebung die Ergebnisse des komplexen Planungsprozesses des MRP-Systems vermittelt bekommen.

Es wird davon ausgegangen, dass Termine im Microsoft Outlook eingetragen werden. Dies soll automatisch nach der Generierung von vorgegebenen Terminvorschlägen geschehen. Diese Terminvorschläge werden als Teil des MRP-Systems generiert. Die Aufgabe war es mit Hilfe der Java Exchange Web Services (EWS) Application Programming Interface (API) die Daten an das Microsoft Outlook zu senden bzw. an den Server zu übermitteln. Die EWS API erleichtert das Arbeiten mit Microsoft Outlook.

Bei flexibler API, kann sowohl Java, C# als auch Extensible Markup Language (XML) verwendet werden. Die Termine werden dabei in Microsoft Outlook-Kalendern dargestellt. Änderungen von Terminen im Microsoft Outlook haben keine Auswirkung auf das MRP System. Weiters können sogenannte „Appointments“ mit Hilfe von vordefinierten Filtern gesucht und exportiert werden. Dies ermöglicht eine einfachere Handhabung von Abfragen. Die Appointments werden dann in Outlook-Termine umgewandelt und können so exportiert werden.

# Kapitel 2

## State of the Art

Im folgenden Kapitel werden die Features des MRP-Planungssystems sowie der EWS Schnittstelle dargestellt.

### 2.1 EWS API

Die Java Exchange Web Services (EWS) Application Programming Interface (API) ist eine von Microsoft bereitgestellte Schnittstelle, welche die Kommunikation mit einem Microsoft Exchange-Server zulässt. Die API kann mit Java und C# angesteuert werden und beinhaltet einen umfangreichen Funktionskatalog um die verschiedenen Funktionen in Microsoft Outlook anzusteuern. Diese Funktionen dienen zur Verwaltung von Emails, Kontakten, Aufgaben, Terminen, Kalendern sowie Benutzerrechten. Die API wurde entwickelt um programmgesteuerte Abfragen und Aufgaben mit Hilfe eines Microsoft Outlookservers zu realisieren. Sie unterstützt synchrone sowie asynchrone Abfragen und Anfragen an den Server die zur Abfrage von Verfügbarkeiten und Benutzerrechten sowie von Microsoft Outlookelementen und Ordnern dienen. Weiters kann mit Hilfe der API gefiltert und gesucht werden. Weiters lässt die EWS API einen Mehrbenutzerbetrieb zu.[6]

#### **HyperText Transfer Protocol und HyperText Transfer Protocol Secure**

Das Internetprotokoll HyperText Transfer Protocol (HTTP) beschreibt die zustandslose Übertragung von Internetdaten. Die Übertragung läuft über die Webmethoden GET, PUT und POST. HTTP verwendet die Standardports, welche in der Transmission Control Protocol (TCP)-Richtlinie definiert worden sind.

HyperText Transfer Protocol Secure (HTTPS) stellt die sichere Verbindung von HTTP dar wobei die Verschlüsselung über Secure Socket Layer (SSL) erfolgt. Webseiten werden mittels HyperText Markup Language (HTML),

JavaScript und Personal Home Page (PHP) angesteuert, wobei die Daten mit Extensible Markup Language (XML) und JavaScript Object Notation (JSON) übertragen werden.[6, 11]

### 2.1.1 Funktionen

#### Exchange Service

Der Exchange Service ist das zentrale Service, welcher Aufgaben verarbeitet und in Verbindung mit dem HTTP(S) Protokoll den Exchange-Server ansteuert. Dieses Service stellt die Verbindung mit dem Server mittels Benutzername, Passwort und Serveradresse her. Außerdem sind zusätzliche Informationen zur Serverversion und Übertragungsart gespeichert. Um die Kommunikation mit dem Server zu ermöglichen, muss die EWS Funktion freigeschaltet sein.[6]

#### Items und Folder

Die Funktionen der EWS API funktionieren nach dem Prinzip der Eindeutigkeit. Dies bedeutet, dass jeder Ordner, jedes Mail und jeder Termin seine eigene ID besitzt, welche vom Server festgelegt wird. Dabei gibt es keine Einschränkung, da jedes Element über die gleiche Priorität verfügt. Weiters kann mit Hilfe dieser eindeutigen ID jedes gesonderte Element abgerufen und beliebig verändert werden. Die Ordner verfügen über eine eindeutige ID mit der auf den ganzen Ordner zugegriffen werden kann. Auch besitzen spezielle Ordner wie z.B. der Posteingang sogenannte *WellKnownFolderNames*, welche wie Aliase bzw. Pseudonyme für IDs fungieren.[6]

#### Email

Emails können mit Hilfe der API gesendet und empfangen werden. Dabei besitzt jedes Email einen zugeordneten Ordner. Die Emails werden automatisch versendet sobald sie am Server eingetroffen sind. Auch können Emails von beliebigen Konten abgefragt und verändert werden. Vordefinierte Kategorien ermöglichen eine organisierte Verwaltung. So können Dateien aus dem lokalen Dateisystem an Emails angehängt werden. Die Emails können per Filter gesucht und mittels Posteingangsregeln erstellt und abgerufen werden. Emails können bei Bedarf in einen Unterordner verschoben werden. [7]

#### Termine / Appointments

Termine können erstellt, abgerufen und gelöscht werden. Sie repräsentieren eine Hauptfunktionalität im Outlook. Jeder Termin beinhaltet Datum, Zeit, Ort sowie Teilnehmer. Die Termine werden in den persönlichen Kalendern hinterlegt. Weiters können den Terminen optionale Teilnehmer hinzugefügt

werden. Periodisch wiederholende Termine werden als Serientermine hinterlegt. Neben dem Hauptkalender können zusätzliche Kalender erstellt werden. Eine weitere Funktionalität stellt die Abfrage von Verfügbarkeiten dar. Diese liefern in weiterer Folge Terminvorschläge. Dabei wird auch die terminliche sowie räumliche Verfügbarkeit mitgeliefert. [7]

### Tasks

Ein weitere Funktionalität der API stellen Tasks dar, welche erstellt und abgerufen werden können. Tasks enthalten eine Liste von Aufgaben, welche Personen zugeordnet werden und von diesen abgearbeitet werden. Einzelne Aufgaben können priorisiert und kategorisiert werden.[6]

### Kontakte

Kontakte stellen neben den Terminen eine wichtige Rolle in der Microsoft Outlookumgebung dar. Die Kontakte werden im Adressbuch verwaltet und können angelegt, abgerufen sowie gelöscht werden. Kontakte können aus vordefinierten Dateien importiert und exportiert werden. Das Adressbuch enthält Kontakte sowie Kontaktgruppen. In den Kontakten sind Namen, Telefonnummern, Emailadressen und persönliche Daten hinterlegt. In dem Adressbuch kann gesucht und gefiltert werden.[7]

## 2.2 Multidimensionales Ressourcen Planning (MRP)

Das MRP-System ist zur Planung von Krankenhausterminen entwickelt worden. Die Hauptaufgabe von MRP ist es aus gegebenen Ressourcen (Personen, Räume, Geräte, Betten, Materialien) in Verbindung mit Regeln Planungskalender für einzelne Räume inklusive Operationsplanungen zu erstellen. Ein Krankenhausinformationssystem (KIS) besteht in erster Linie aus den Zentralkomponenten Organisation, Bettmanagement, Aufnahme, Entlassung sowie Operationsplanungen. Das MRP System deckt nicht die zentralen Komponenten ab, da sich dieses System nur mit der Planung von Räumen und Terminen mithilfe von Ressourcen beschäftigt. Es übernimmt nicht die Verwaltung des Krankenhauses. [8, 5, 9, 2]

### 2.2.1 Funktionsweise

Das MRP System generiert aus vorgegebenen Daten geeignete Termine und Pläne für Räume des Krankenhauses. Es werden die benötigten Ressourcen gesammelt und in einer zentralen Datenbank verwaltet. Aus den Datensätzen werden mittels Ressourcenmanager die Ressourcen zusammengetragen

und Appointments generiert. Diese werden dann mit Hilfe des Kapazitätsmanagementtools den verfügbaren Kapazitäten der einzelnen Ressourcen zugeordnet und in einer Kalenderansicht dargestellt.<sup>1</sup>

### 2.2.2 Masterdata

Jedes Feature im MRP-System folgt der selben Struktur. Jedes Feature besteht aus den Teilen API, Component, Datenbank und Service. Dabei wird die Drei-Schichten-Architektur realisiert. Dies bedeutet, dass es drei getrennte Schichten in der Implementierung gibt, welche als eigenständig fungieren. Die unterste Schicht stellt die Datenbank-Schicht dar und ist zuständig für die Verwaltung von persistenten Daten. Die Mittlere Schicht ist die Business-Logik bzw. Verarbeitungsschicht. Diese ruft die Daten von der untersten Schicht ab und leitet sie gegebenenfalls an die Präsentationsschicht weiter. Es besteht die Möglichkeit für Berechnungen und Änderungen der Daten. Die Präsentationsschicht ist zuständig für die visuelle Darstellung der Daten. Die erweiterte 3-Schichten-Architektur hat den Vorteil, dass die Aufgaben der Schichten klar von einander getrennt sind. Weiters gibt es die Component-Service-Implementation, die aus den Komponenten, Datenbank, Service, und API besteht[1].<sup>2</sup>

### Struktur

Jedes Feature folgt der Component-Service-Implementation, welche aus folgenden Teilen besteht(siehe Abb. 2.1)[4]:

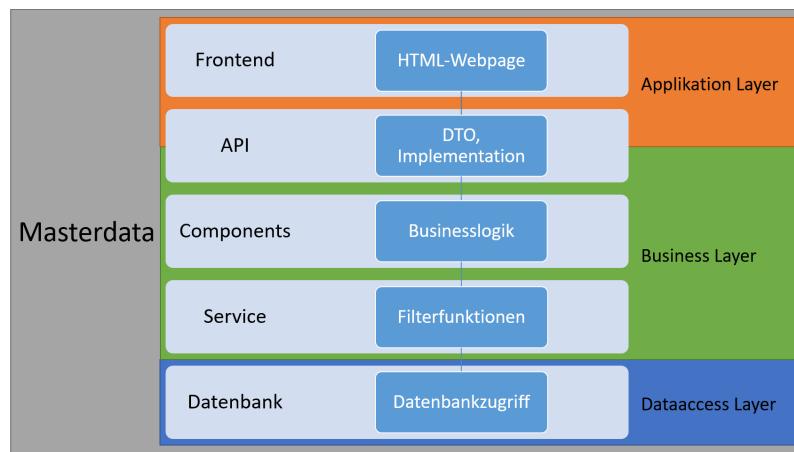


Abbildung 2.1: Architekturdiagramm

<sup>1</sup>MRP Technical Documentation

<sup>2</sup>G3 HIS Reference Documentation

- *Datenbank:* Das Datenbankfeature verwaltet die Daten in einer Datenbanktechnologie, welche variabel auswählbar ist. Das Speichern und Abrufen von Daten wird über den jeweiligen Datenbanktreiber betrieben, welcher mit einer standardisierten Datenbankansteuerung kommuniziert. Dieser befindet sich innerhalb eines Data Access Object (DAO). Das DAO beinhaltet die relevanten Datenbankzugriffsfunktionen und liefert ein datenbankspezifisches Data Transfer Object (DTO).
- *Service:* Das Service erhält Daten aus der Datenbank und sendet diese an die Components weiter. Die Services dienen den Components für eine erleichterte Abfragemöglichkeit von Datenobjekten. Weiters benötigen die Services vordefinierte Filterfunktionen. Die Funktionalität der Services gehört der Business-Logik-Schicht an.
- *Components:* Die Components stellen im Feature die Hauptbestandteile dar und sind ein Teil der Businesslogik. Die Components repräsentieren die eigentliche Logik, da in diesen Funktionen die Verarbeitung der Daten erfolgt. Die Components erhalten die Daten aus der Datenbank und wandeln diese in *Data Transfer Objects* um. Die Objekte werden an die API weitergeleitet.
- *API:* Die API ist ein Teil der Businessschicht und stellt die Verbindung zum Frontend dar. Die API bekommt die Daten von den Components und wandelt diese in webkonforme Übertragungsobjekte um, welche an das Frontend weitergeleitet werden. Außerdem ist die API zuständig für das Empfangen der Daten aus dem Frontend.
- *Frontend:* Das Frontend repräsentiert die Präsentationsschicht, welche für den jeweiligen Benutzer sichtbar ist. Im Frontend werden die Benutzerinteraktionen verwaltet, Daten zu Transferobjekte umgewandelt und an das Backend gesendet.

### 2.2.3 Hospital Information System

Das Hospital Information System (HIS) von CGM wird unter der Produktbezeichnung G3 HIS geführt und besteht aus mehreren Modulen. Die Zentralkomponenten eines KIS sind in diesem System integriert und bilden eine einheitliche Weboberfläche. Diese ist betriebssystemunabhängig und wird in einem vom G3 HIS unterstützten Browser verwendet. Das G3 HIS Projekt umfasst mehrere Projektgruppen, die in die unterschiedlichen Aufgaben (Organisation, Bettenmanagement, Aufnahme, Entlassung, Operationsplannungen) eines Krankenhauses unterteilt sind. Sie bieten eine umfangreiche Konfiguration sowie Verwaltung der einzelnen Funktionsgruppen bzw. Module an. Die Software stellt die 3. Generation dar und wird ständig weiterentwickelt.

## Kapitel 3

# Konzept und Design

In diesem Kapitel wird das Konzept zur Übertragung von Daten aus dem MRP ins Microsoft Outlook beschrieben.

MRP ist eine von CGM entwickelte Software, die Operationstermine in Krankenhäusern unter Berücksichtigung der beteiligten Ressourcen optimiert und berechnet. Der Ablauf ist wie folgt.<sup>1</sup>:

1. Zu Beginn werden die verfügbaren Ressourcen (Personen, Räume, Geräte, Betten, Materialien) angelegt und zugeordnet.
2. Zuordnung der ressourcenbezogenen Daten: Jeder Ressource werden Informationen zugeteilt.
3. Aus Ressourcen werden mithilfe von Kapazitäten (terminliche Verfügbarkeiten) Organisationseinheiten (Zeitbereiche mit verknüpften Ressourcen) gebildet.
4. Aus diesen Einheiten werden Vorschläge gebildet. Diese Vorschläge beinhalten lediglich einen Zeitbereich. Um einen Termin festlegen zu können, müssen Ressourcen definiert werden. Dies entsteht aus den bereits hinterlegten Personendaten sowie aus den Verfügbarkeitsabfragen der EWS Schnittstelle. Mit diesen erweiterten Funktionen können konkrete Termine generiert u. nach deren Bestätigung ins Outlook exportiert werden. Die exportierten Termine können im betreffenden Kalender eingesehen werden. Das Anlegen von Ressourcen und das Selektieren von exportierten Terminen erfolgt manuell.

Die Erstellung eines MRP Moduls erfolgt nach den Programmierrichtlinien von CGM. Dieses Modul dient zum Export von generierten Terminen ins Outlook. Weiters werden Filter- und Suchfunktionen bereitgestellt. Diese können MRP- als auch Outlook-Appointments filtern und suchen.

---

<sup>1</sup>G3 HIS Reference Documentation

# Kapitel 4

# Implementierung

In Abstimmung mit CGM wurde das Konzept aus Kapitel 3 für die Umsetzung gemäß Zielsetzung herangezogen. Dabei wurden die Funktionen der EWS-API verwendet und in den Methoden der nachfolgenden Module aufgerufen.

## 4.1 Microsoft Outlook

Die nachfolgenden aufgelisteten Packages beziehen sich auf die Klassen: *Packagename + Service*, *I + Packagename + Service* und *Packagename + Dto* (siehe Klassendiagramm Abb. 4.1 und Abb. 4.2)

Die jeweilige Klasse des Packages implementiert eine vordefinierte Schnittstelle (Interface), welche einen vordefinierten Benennungsschema folgt. Die Funktionalitäten sind unabhängig vom MRP-System, können aber von diesem verwendet werden. Die für die Arbeit entwickelten Datentypen sind im Package *Model* definiert. Diese Klassen können gleichnamige Klassen im MRP haben, haben aber keine Beziehung zu diesen Klassen.

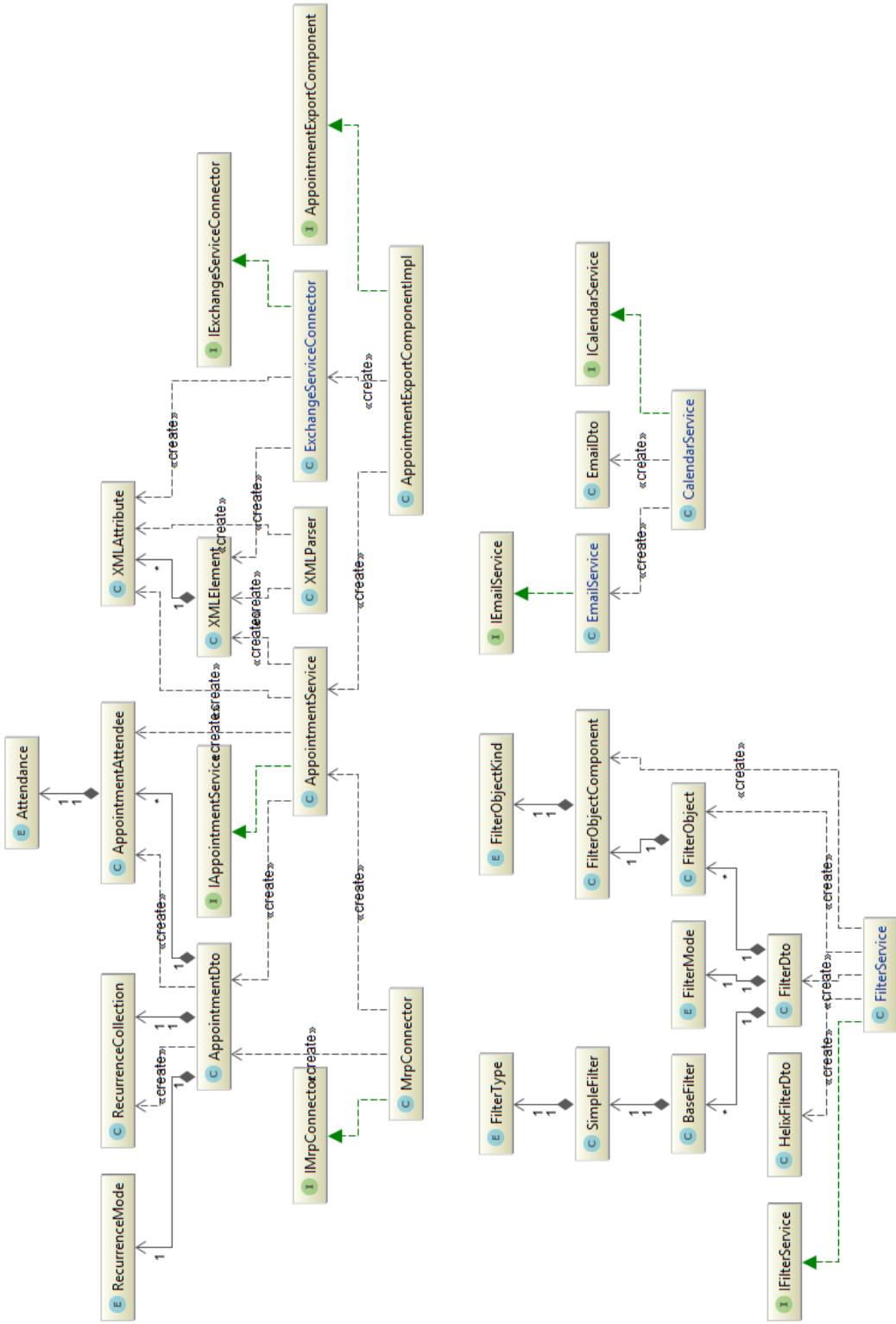


Abbildung 4.1: Klassendiagramm

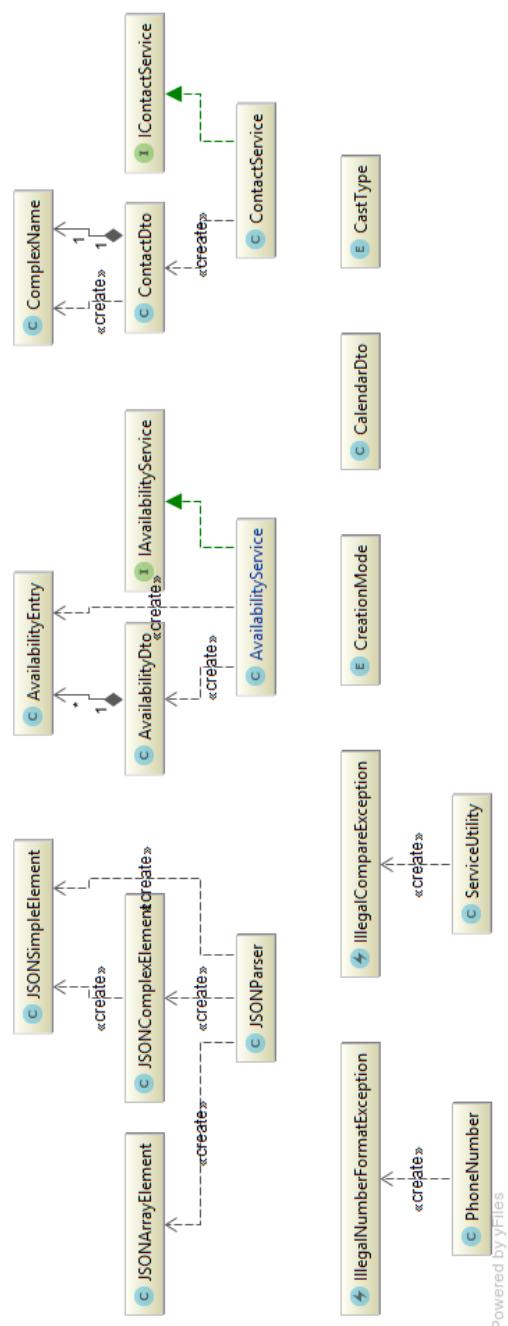


Abbildung 4.2: Klassendiagramm

#### 4.1.1 Exchangeservice

Der Exchangeservice dient als zentrale Verbindung zum Microsoft Outlook. Die Features bzw. Methoden, welche mit dem Exchange-Server kommunizieren, verwenden ExchangeService als Parameter.

Der *ExchangeServiceConnector* ist zuständig für die Verbindung zum Server sowie für die Anfragen der Features. Dieser Service bezieht sich ausschließlich auf das eigene Outlookkonto bzw. auf das Konto, welches mit Passwort und Benutzername angegeben wird. Weiters besteht die Möglichkeit die Exchangeserverversion festzulegen. Die Klasse *ExchangeServiceConnector* implementiert das zugehörige Interface *IExchangeServiceConnector*.

Die Serververbindung wird mit Hilfe der Methode *createServerConnection* initialisiert, wobei der Benutzername und das Passwort als Parameter übergeben werden. Die gleichnamige Methode verwendet zusätzlich den Parameter Server-URL. Die Methode *createServerConnection* baut mit Hilfe der Daten eines XML-Dokuments eine Serververbindung auf. Die genannten Methoden returnieren eine Instanz des Exchangeservice. Dieser Service existiert nur einmal zur Laufzeit. Damit wird sichergestellt, dass immer der gleiche Server angesteuert wird. Mit der Methode *saveSettingsToXML* können die Servereinstellungen in einem XML-Dokument gespeichert werden.

#### 4.1.2 Model

Die verwendeten Objekte bzw. Datentypen in den einzelnen Features bzw. Packages werden im Model repräsentiert. Die einzelnen Klassen repräsentieren die jeweiligen Datentypen, welche einer vordefinierten Struktur folgen. Diese folgt der Bean-Definition. Das bedeutet, dass die Membervariablen *private* sind und daher auf die Variablen mit Getter und Setter zugegriffen wird. Die einzelnen definierten Datentypen besitzen einen öffentlichen Default-Konstruktor ohne Parameter [3].

#### 4.1.3 Kalender

Mittels Kalenderservice können Kalender angelegt und abgerufen werden. Der *CalendarService* implementiert das zugehörige Interface *ICalendarService* und verwendet den Datentyp *CalendarDto*.

Die Methode *getCalendars* ruft die verfügbaren Kalender einer Emailadresse ab. Die Emailadresse repräsentiert Personen oder Räume. Zusätzliche Kalender werden durch die Methode *createNewCalendar* angelegt. Auch gibt es die Methode *sendWarningToUser*, mit dieser kann ein Email an den betreffenden Benutzer versendet werden. Die Emailnachricht enthält die betreffende Warnung.

#### 4.1.4 Appointments

Mit Hilfe dem Appointmentservice können Microsoft Outlook-konforme Appointments angelegt, verwaltet und abgerufen werden. Weiters ist möglich Appointments in einem XML-Dokument zu speichern. Das XML Dokument beinhaltet die Datenfelder aus dem *Model* in einer validen XML-Struktur. Diese XML-Dokumente können zur Erstellung von Appointments verwendet werden. Es besteht die Möglichkeit Konflikte zu betreffenden Appointments vom Server abzufragen.

Der *AppointmentService* implementiert das zugehörige Interface *IAppointmentService* und verwendet den Datentyp *AppointmentDto*. Mit der Methode *createAppointment* kann ein Appointment angelegt werden. Der *ExchangeService* wird als Parameter übergeben. Die Methode *buildAppointment* erstellt den Datentyp *Appointment*. Die Methode *createAppointments* ermöglicht das gleichzeitige Speichern und Anlegen von Appointments. Eine weitere Funktionalität ist durch die Methode *hasConflictedAppointments* gegeben. Diese Funktion überprüft mit Hilfe des Exchangeservers Konflikte bei betreffenden Appointments. Die Funktionen *appointmentListToXML*, *xmlToDto* und *xmlToAppointmentList* ermöglichen die Verwaltung von outlookkonformen Appointments mithilfe von XML.

#### 4.1.5 Availability

Durch den Availabilityservice können Verfügbarkeiten von Personen und Räumen abgefragt werden. Der *AvailabilityService* implementiert das zugehörige Interface *IAvailabilityService* und verwendet den Datentyp *AvailabilityDto*.

Die Hauptfunktion *checkAvailability* ruft mit Hilfe dem *ExchangeService* Verfügbarkeiten über die EWS-Schnittstelle ab. Die Methode *checkAvailabilityForPerson* überprüft die Verfügbarkeit von Personen anhand deren Email-Adressen.

Mit Hilfe der Methode *checkAvailabilityForRooms* können im Gegensatz zur obengenannten Methode anstelle von Personen Räume abgefragt werden. Die beiden Funktionen *getRoom* und *getAllRooms* liefern Räume zurück. Personen und Räume werden durch die jeweiligen Emails repräsentiert.

#### 4.1.6 Kontakt

Der Kontaktservice ermöglicht das Anlegen, Abrufen und Verwalten von Outlookkontakten in den jeweiligen persönlichen Adressbüchern. Der Kontaktservice arbeitet mit dem Benutzer des Exchangeservices. Der *ContactService* implementiert das zugehörige Interface *IContactService* und verwendet den Datentyp *ContactDto*.

Die Methode *getContacts* ruft persönliche Kontakte des angemeldeten Benutzers aus dem persönlichen globalen Adressbuch ab. Die Kontakte sind un-

sortiert und können mittels Filter zusätzlich gefiltert werden. Neue Kontakte werden über die Funktion *buildContact* erstellt. Die Funktion *getContactDtos* liefert mithilfe der Funktion *buildContact* eine Liste vom vordefinierten Datentyp *ContactDto*.

#### 4.1.7 Email

Der Emailservice ermöglicht das Senden von bestehenden Emails. Diese beinhalten die relevanten Felder der Emails. Weiters können die Emails aus dem jeweiligen Postfach des Benutzers aufgerufen werden. Der *EmailService* implementiert das zugehörige Interface *IEmailService* und verwendet den Datentyp *EmailDto*.

Durch die Methode *sendEmail* können Emails versendet werden. Diese werden in der Methode *buildEmail* erstellt und enthalten alle relevanten Informationen zu Sender und Empfänger sowie Betreff und Emailtext. Die Methode *getEmails* ruft eine vordefinierte Anzahl Emails aus dem Posteingang ab.

#### 4.1.8 Filter

Die Filter bestehen aus Filterkomponenten, welche vordefiniert sind. Diese Filterkomponenten folgen dem Prinzip der überpersistente Speicherung. Dies bedeutet, dass bereits gefilterte Daten trotzdem erhalten bleiben. Grundsätzlich ist jede Filterkomponente aus zwei Teilen aufgebaut: Der erste Teil beinhaltet die Informationen zu den Filtern. Der zweite Teil stellt eine Datensammlung dar. Diese Daten bleiben solange erhalten solange die Filterkomponenten aktiv sind. Die Filterkomponenten werden gleichzeitig bzw. hintereinander angewendet, um die logischen Funktionen AND und OR zu repräsentieren. Eine weitere Funktionalität ist der Vergleich von Methoden mit den Variablen. Die Methode *apply* beschreibt die Hauptfunktionalität des Filters und beinhaltet die Funktionalität, die zur Anwendung eines bestimmten Filters erforderlich ist. Das *FilterDto* beinhaltet sowohl die Filterkomponenten als auch die zu filternden Daten. Mit der Methode *applySimpleFilter* wird ein Filter für eine Sammlung von zu filternden Objekten angewandt. Die Funktion *applyFilter* wird verwendet, wenn die zu filternden Objekte den Typ *AppointmentComponent* darstellen. Die zwei Funktionen *applyHelixFilter* werden verwendet, wenn die Filter vom MRP-System stammen. Die Funktionen *getFiltered* und *getUnfiltered* dienen zum Abrufen von gefilterten und ungefilterten Daten eines *FilterDtos*.

## 4.2 Masterdata-Feature

Als Masterdata werden Features bezeichnet, welche zentrale Daten im MRP verwalten und in einer Datenbank speichern (Klassendiagramm siehe Abb. 4.3).

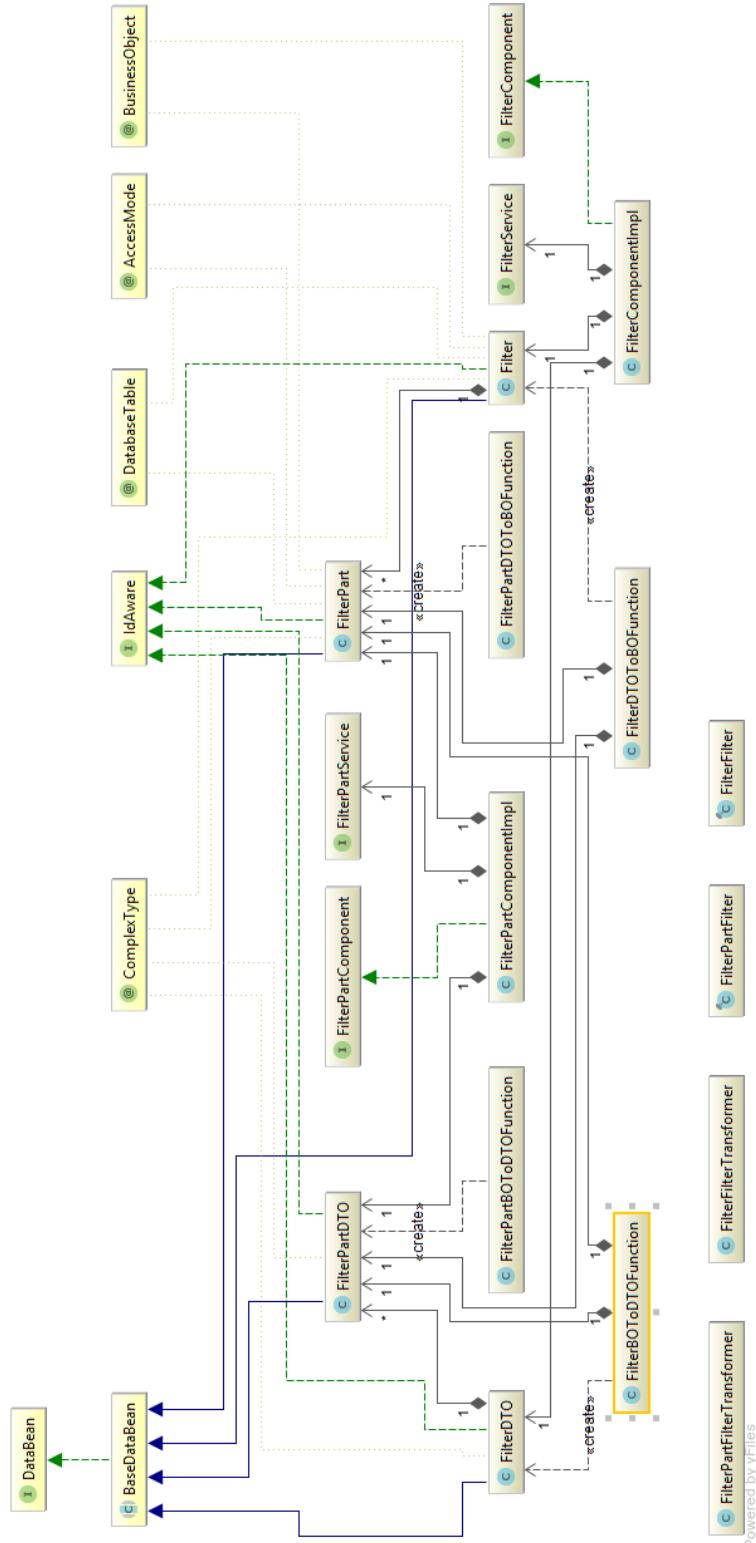


Abbildung 4.3: Klassendiagramm

#### 4.2.1 Filter

In dem MRP-Feature Masterdata wurde ein Feature hinzugefügt, welches zur Verwaltung von Filtern dient. Die Filter werden in der zentralen Datenbank von MRP gespeichert und über die Weboberfläche von MRP verwaltet, da die Filter zu den Masterdata-Features zählen. Die Filter dienen zur Filterung von MRP-Appointments. Jeder Filter beinhaltet die benötigten Filterkomponenten, welche für das Filtern der Daten verantwortlich sind.

#### 4.2.2 Filterkomponenten

Jeder Filter hat untergeordnete Filterkomponenten, welche die Informationen zu Filterart, Filtertyp und Filterwert enthalten. Diese Komponenten werden in einer Relation zu den Filtern in der Datenbank abgelegt. Die Filterkomponenten sind immer für einen bestimmten Filter zuständig und können nicht einzeln verwendet werden. Die Filterkomponenten dienen zum Filtern von Appointments.

#### 4.2.3 Weboberfläche

In der Weboberfläche von dem MRP-System werden die Filter und Filterkomponenten verwaltet. Die Oberfläche wird von G3 HIS bereitgestellt und bietet die Funktionalität von MRP. Mit Hilfe der Filter werden Appointments gesucht und angezeigt. Zukünftig sollen selektierte Appointments ins Microsoft Outlook exportiert werden können.

# Kapitel 5

## Evaluierung

Das Outlookmodul ist im MRP-System teilintegriert. Die Testumgebung wird mit Hilfe eines Jenkins-Maven-Testserver realisiert, welcher selbständiges automatisiertes Testen zulässt. Bei einen Jenkins-Maven-Testserver handelt es sich um einen Linux-Server, welcher das gesamte Projekt in regelmäßigen Abständen compiliert und die vorgesehenen Tests ausführt. Die Ergebnisse sind in einer Weboberfläche einsehbar. In dem Repository gibt es die Branches *master* und *dev*. Diese werden in die automatisierte Testumgebung integriert und in fixierten Zeitabständen getestet. Da das zu bearbeitende Modul kein Teil der zwei Branches ist, kann es nicht auf diese Art getestet werden. Durch den Umstand des Testprozesses muss auf andere Testmittel umgestiegen werden.

### 5.1 Benutzerinteraktives Testen

Auf Grund des Testprozesses wird auf die Methodik benutzerinteraktives Testen zurückgegriffen. Die gewählte Methodik ist auch bekannt als *Whitebox-Test*[10]. Diese Methodik beruht auf der visuellen Kontrolle des Programmierers. Dabei überprüft der Programmierer die Ergebnisse in der grafischen Oberfläche bzw. Konsole. Die Testmethodik dient rein zur visuellen Kontrolle ob die geforderten Datenfelder richtig in der Konsole angezeigt werden. Trotzdem kann mit Hilfe dieser Methodik die Funktion des Programms getestet werden. Dies ist auch möglich wenn kein geeignetes Testframework vorhanden ist. Da das Modul mit einem Server kommuniziert, bei dem die Verarbeitung der Anfragen einen Engpass darstellt, stellt sich automatisiertes Testen mittels Frameworks zu aufwändig für den Projektrahmen dar.

Im Rahmen der Testung werden folgende Funktionen überprüft:

Getestete Funktionen	Tests erfolgreich
Appointments anlegen	16
Appointments abrufen	24
Availablilities abfragen	20
Kalender anlegen	21
Kalender abfragen	21
Kontakte abrufen	14
Adressbuch abfragen	30
Emails senden	40
Emails abrufen	45
Serververbindung initialisieren	5
Serververbindung speichern	5
Serververbindung abfragen	10
Filter verwalten	38
Filterkomponenten verwalten	64
Filterkomponenten zuordnen	23
MRP-Filter verwalten	12
MRP-Filterkomponenten verwalten	17
MRP-Appointments exportieren	0 (keine Tests vorhanden)

**Tabelle 5.1:** Tests

# Kapitel 6

## Zusammenfassung

In diesem Kapitel werden die Ergebnisse zusammengefasst und diskutiert.

### 6.1 Resultate

Bei dieser Arbeit wurde ein Programm implementiert, welches als Modul für das MRP-System der Firma CGM fungiert. Die Implementierung wurde mittels JAVA durchgeführt. Es wurde die Java EWS API verwendet. Die EWS API dient zur Ansteuerung eines Microsoft Outlookservers mit Hilfe von Programmmethoden. Das Modul verwendete diese API um bereitgestellte Termine aus dem MRP-System an das Microsoft Outlook zu exportieren. Die Daten wurden von weiteren Modulen des MRP-Systems generiert und bereitgestellt. Allerdings fehlte die Integration des Moduls in das gesamtheitliche MRP-System, da dieses zum damaligen Zeitpunkt noch nicht fertiggestellt war. Das Testen erfolgte durch benutzerinteraktives Testen. Dieses beruhte auf einer rein visuellen Betrachtung des Benutzers. Die diesbezüglichen Tests verliefen positiv.

### 6.2 Diskussion

Das in dieser Arbeit erstellte Programmmodul ist universell einsetzbar, wurde aber geringfügig an das MRP-System angepasst. Die EWS API mit ihren komplexen Datentypen wurden durch das vereinfachte DTO ersetzt. Die universelle Einsetzbarkeit wurde damit ermöglicht. Die Implementierung der verwendeten Filter wurde generisch gehalten, um eine universelle Einsetzbarkeit zu ermöglichen. Seitens CGM ist es geplant das im Rahmen dieser Arbeit erstellte Modul in einem zukünftigen Release in das MRP-System zu integrieren.

# Quellenverzeichnis

## Literatur

- [1] Len Bass, Paul Clements und Rick Kazman. *Software Architecture in Practice*. 3rd. Addison-Wesley Professional, 2012.
- [2] Robert H. Dolin u. a. „The HL7 Clinical Document Architecture“. *Journal of the American Medical Informatics Association* 8.6 (2001), S. 552–569.
- [3] David Flanagan. *Java In A Nutshell, 5th Edition*. O'Reilly Media, Inc., 2005.
- [4] John Grundy. „Multi-Perspective Specification, Design and Implementation of Software Components using Aspects“. *International Journal of Software Engineering and Knowledge Engineering* 10.06 (2000), S. 713–734.
- [5] Peter Haas. *Medizinische Informationssysteme und elektronische Krankenakten*. Springer Verlag, 2004.
- [6] *Java EWS API Getting Started Guide*. [zuletzt besucht am 13.12.2016]. <https://github.com/OfficeDev/ews-javascript-api/wiki/Getting-Started-Guide>, 2016.
- [7] T. Joos. *Microsoft Outlook 2013 - Das Handbuch*. O’Rielly Verlag GmbH, 2013.
- [8] Seung-Chul Kim u. a. „Analysis of capacity management of the intensive care unit in a hospital“. *European Journal of Operational Research* 115.1 (1999), S. 36–46.
- [9] Klaus Kuhn u. a. „A Conceptual Approach to an Open Hospital Information System“. In: *Proc. 12th Int'l Congress on Medical Informatics (MIE '94)*. Mai 1994, S. 374–378.
- [10] Andreas Spillner und Tilo Linz. *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB-Standard*. dpunkt, 2005.
- [11] w3c. *HTTP - Hypertext Transfer Protocol*. [zuletzt besucht am 13.12.2016]. <https://www.w3.org/Protocols/>, 2014.