In Kooperation mit:

AIT Austrian Institute of Technology

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

09.09.2015
Datum

Sandra Neubauer
Unterschrift

Danksagung

Ich möchte mich bei allen von ganzem Herzen bedanken, die mich auf diesem Wege begleitet und sowohl direkt als auch indirekt unterstützt haben.

Kurzfassung

Im Gesundheitsbereich werden enorme Datenmengen gesammelt. Die Herausforderung besteht zukünftig darin, mit modernen Methoden aus den Bereichen Data Mining, Knowledge Discovery und Human-Computer-Interaction das relevante Wissen aus diesem Datenmeer zu extrahieren, um die darin verborgenen Potentiale für Patient und Gesundheitssystem nutzen zu können. Ziele dieser Arbeit sind das Aufbereiten von Lerndaten aus codierten Abrechnungsdaten für das Predictive Modelling mit maschinellen Lernverfahren, sowie die Entwicklung eines Visual Analytics Frameworks zur einfacheren Gestaltung des Modellierungsvorganges und zur verbesserten Interaktion zwischen Mensch und Maschine.
Umgesetzt wurden diese Aufgaben unter Verwendung von MATLAB (Mathworks Inc., Natick, USA) in der Version 2013b. Die Validierung erfolgte am Beispiel der Vorhersage zukünftiger Krankenhausaufenthalte mit Daten von Klienten eines australischen Krankenversicherungsunternehmens, welche nach ICD-10-AM und AR-DRG codiert sind.
Die Extraktion von Features aus Hauptdiagnosen erfolgte durch Entschlüsselung der Basisklassifikation der hierarchisch gegliederten ICD-10 Systematik der WHO. Aus den Nebendiagnosen wurde der Charlson Komorbiditäts-Score berechnet. Für das Ableiten von Features aus codierten Krankenhausleistungen wurde das standardisierte Schema des australischen Patientenklassifikationssystems (AR-DRG) herangezogen. Das Framework setzt sich aus vier Tools zusammen: Daten-, Statistik-, Modellierungs- und Zeitreihen-(Evaluierungs)-Tool.
Erste Erkenntnisse zeigen, dass das Aufbereiten von Trainingsdaten für den maschinellen Lernalgorithmus ein nicht-trivialer Prozess ist, der von zahlreichen Einflussfaktoren abhängt. Visual Analytics Tools können beim gezielten Aufbereiten von Lerndaten unterstützen und bessere Einsicht in die Stärken und Schwächen maschinell generierter Vorhersagemodelle gewähren.

Information Science, Big Data, Visual Analytics, Interactive Data Mining, Machine Learning

Abstract

As the volume of data from the health sector is growing at a staggering rate, a combination of technologies from the areas Data Mining, Knowledge Discovery and Human-Computer-Interaction is becoming increasingly important. The purpose of this work was to implement a concept for automatically extracting features from clinical codes and to develop a Visual Analytics Framework helping to combine the strengths of man and machine in the context of data-driven Predictive Modelling.

For implementation, MATLAB (Mathworks Inc., Natick, USA) in the version 2013b was used. Validation was performed by predicting future days in hospital on the basis of health insurance claims data, which contained medical claims encoded in ICD-10-AM and AR-DRG format.

For main diagnoses, feature extraction was done using the hierarchically structured core classification of the World Health Organizations ICD-10 Coding System. From secondary diagnoses the Charlson Comorbidity-Score was calculated. The standardized coding scheme of the AR-DRGs formed the basis for data transformation of encoded hospital services. The framework comprises four tools: Data-, Statistic-, Modeling- and Time-Series-(Evaluation)-Tool.

The predictive performance of data-driven models is highly affected by the way in which the data is represented. Thus, developing proper methods for feature generation and representation remains a pivotal area in machine learning. Visual Analytics can help in providing training data in a more targeted manner and with gaining insights into the strengths and the weaknesses of non-transparent machine-generated models.

Information Science, Big Data, Visual Analytics, Interactive Data Mining, Machine Learning

Inhaltsverzeichnis

1 Einleitung 1
1.1 Motivation 1
1.2 Wissensgenerierung 2
1.3 Ausgangssituation 5
1.4 Aufgabenstellung 6
1.5 Untersuchungsbereich 6
2 Methoden 7
2.1 Unternehmensinterne Strukturen und Prozesse 7
2.1.1 Datenstruktur 7
2.1.2 Routinen der prä-existenten Predictive Modelling Pipeline 9
2.2 Feature-Extraktion 10
2.2.1 ICD-10 Codes: Hauptdiagnose 10
2.2.2 ICD-10 Codes: Nebendiagnosen 11
2.2.3 DRG Codes 12
2.3 Visual Analytics 12
2.3.1 Allgemeine Grundlagen zur GUI-Programmierung mit MATLAB 12
2.3.2 Framework (siehe 3.2.1) 13
2.3.3 Data-Tool (siehe 3.2.2) 13
2.3.4 Zeitreihen-Tool (siehe 3.2.3) 16
2.3.5 Statistik-Tool (siehe 3.2.4) 20
2.3.6 Modellierungs-Tool (siehe 3.2.5) 21
3 Ergebnisse 21
3.1 Feature-Extraktion 21
3.1.1 ICD-10 Codes: Kurzbeschreibung 21
3.1.2 ICD-10 Codes: Hauptdiagnose 22
3.1.3 ICD-10 Codes: Nebendiagnosen 23
3.1.4 AR-DRG Codes 24
3.2 Visual Analytics 27
3.2.1 Framework 27
3.2.2 Data-Tool 27
3.2.3 Zeitreihen-(Eval)-Tool 31

3.2.4 Statistik-Tool 36
3.2.5 Modellierungs-Tool 40
4 Diskussion 42
4.1 Feature-Extraktion 42
4.1.1 ICD-10 Hauptdiagnose-Codes und AR-DRGs 42
4.1.2 ICD-10 Codes: Nebendiagnosen 43
4.2 Visual Analytics 44
4.2.1 Zeitreihen-Tool 44
4.2.2 Statistik-Tool 48
4.2.3 Modellierungs-Tool 48
5 Zusammenfassung und Ausblick 49
Abbildungsverzeichnis 50
Tabellenverzeichnis 53
Literaturverzeichnis 54

Allgemeine Orientierungshinweise

Um eine bessere Lesbarkeit zu erzielen sind Begriffe wie Patient oder Klient sowohl in weiblicher als auch in männlicher Form zu verstehen.

Kursiv geschriebener Text entspricht einem MATLAB-Befehl. `MATLAB-Funktionen` sind gesondert formatiert.

Im Kapitel Resultate werden das Konzept zur Feature-Extraktion, sowie die wichtigsten für eine Anwendung erforderlichen Funktionalitäten des Visual Analytics Frameworks beschrieben. Um den Interessen eines Entwicklers nachzukommen, werden im Kapitel Methoden die dahinterstehenden Strukturen, Prozessabläufe und deren Umsetzung näher beleuchtet. Wird beispielsweise im Kapitel Resultate beschrieben, dass Daten automatisch geladen werden – WAS – , so wird im Kapitel Methoden deutlich gemacht, WIE das automatische Laden zustande kommt.

Unter Rohdatensätze sind in dieser Arbeit bereits codierte und in Datasets konvertierte Quelldatensätze zu verstehen. Der Begriff Rohdatensatz wird verwendet, um den Unterschied zwischen MATLAB-Datasets, die ausschließlich Daten von einer Quelle beinhalten, und jenem MATLAB-Dataset (Featureset) hervorzuheben, in dem Daten aus verschiedenen Quellen zusammengeführt sind.

Die Tools des Frameworks wurden auf Basis von Versicherungsdaten des australischen HCF-Projektes entwickelt und anschließend mit Daten von zwei weiteren Projekten (Patient Blood Management, Gesundheitsdialog Diabetes) validiert. Sofern auf den Screenshots Daten aus den letzteren beiden Projekten zu sehen sind, wird explizit darauf hingewiesen.

Abkürzungsverzeichnis

AR-DRG	Australian Refined Diagnosis Related Groups
AUC	Area Under The Receiver Operating Characteristic Curve
Carc.	Carcinoma (Karzinom)
EHR	Electronic Health Record
GUI	Graphical User Interface
GUIDE	Graphical User Interface Developement Environment
HCF	Health Care Fund
HCI	Human Computer Interaction
ICD-10-AM	International Classification of Diseases and Related Health Problems, 10th edition, Australian Modified
KDD	Knowledge Discovery in Databases
USA	United States of America
WHO	World Health Organization

1 Einleitung

In diesem Abschnitt wird das Potential von Big Data im Gesundheitsbereich aufgezeigt, wie diese großen Datenmengen in Wissen transformiert werden können und wo dabei die Schwierigkeiten und Herausforderungen liegen.

1.1 Motivation

Der fortschreitende Trend zur Digitalisierung im Gesundheitswesen und der elektronischen Archivierung von Patientendaten, moderne Technologien wie Telemonitoring oder mobile Apps und Innovationen im Bereich des Self-Tracking führen zu einem rasanten Anstieg des Datenvolumens. Der Healthcare-Sektor zählt mit einer jährlichen Wachstumsrate von knapp 50 % zu den schnellst wachsenden Märkten des digitalen Universums (Gantz and Reinsel, 2012, IDC, 2014).

Dieses Phänomen eröffnet dem Gesundheitssystem völlig neue Perspektiven. Durch eine intelligente Nutzung dieser Daten können beispielsweise potentiell bevorstehende Erkrankungen frühzeitig erkannt werden (Wu et al., 2010), Behandlungen auf die individuellen Bedürfnisse des Patienten abgestimmt (Donsa et al., 2015, Panahiazar et al., 2014), oder eine drohende Influenzaepidemie rechtzeitig prognostiziert werden (Mayer-Schönberger and Cukier, 2013). Administratoren des Gesundheitssystems können diese Daten als Planungsinstrument zur effizienteren Allokation von Ressourcen, Vermeidung von unnötigen Kosten (Donzé et al., 2013, Hasan et al., 2010, Xie et al., 2014), oder als Grundlage für die Qualitätskontrolle (Halfon et al., 2006) einsetzen.

Das Informationszeitalter bietet somit die Möglichkeit, sich von einer reaktiven Medizin hin zu einer prädiktiven und präventiven Medizin zu bewegen, den Weg in Richtung individualisierter Medizin einzuschlagen und damit gleichzeitig Rationalisierungen vorzunehmen. Davon profitiert potentiell nicht nur der Patient durch einen Gewinn an zusätzlichen Jahren in besserer Gesundheit, sondern auch die Öffentliche Verwaltung und die Krankenkassen.

Big Data stellt die Wissenschaft jedoch vor eine wesentliche Herausforderung. Solange der Datenberg nicht zu verwertbarem Wissen verarbeitet wird, ist er als Basis zur Entscheidungsunterstützung nutzlos.

John Naisbitt hat in seinem Buch „Megatrends“ bereits 1982 diese Problematik aufgezeigt (Naisbitt, 1982).

> *„We are drowning in information, but starved for knowledge“.*
> *John Naisbitt*

Heutzutage ist es nicht ausschließlich das Überleiten von Informationen in Wissen, das Schwierigkeiten bereitet. Eric D. Brown bringt mit seiner ergänzenden Eigeninterpretation dieses Zitats die gegenwärtige Situation auf den Punkt:

> *"Today, we are drowning in data and starved for information".*
> *Eric D. Brown.*

Daten sind Rohinformationen, wie beispielsweise Sensordaten, codierte Diagnosen oder das Datum für die Einweisung in ein Krankenhaus. Ohne weitere Vorverarbeitung kann auf Basis von Daten keine Aussage getätigt werden. Informationen sind strukturierte Daten, die in einem Kontext zueinander stehen. So kann beispielsweise eine Subtraktion des Austritts- und des Eintrittsdatums die Information liefern, wie viele Tage der Patient im Krankenhaus verbracht hat. Wissen entsteht durch eine logische Verknüpfung von Daten und Informationen unter Einbeziehung von Expertenmeinungen, Fähigkeiten und Erfahrungen, sodass beispielsweise Wissen über die Anzahl der Tage geschaffen werden kann, die ein Patient im nächsten Jahr im Krankenhaus verbringen wird. Weisheit erlangt, wer versteht dieses Wissen in die Praxis umzusetzen. Weisheit könnte somit darin bestehen, die vermeidbaren vorhergesagten Tage im Krankenhaus durch vorzeitiges Setzen von Maßnahmen zu verhindern (Vgl. Landauer, 1998).

Die Herausforderung besteht im Wissensgenerierungsprozess somit darin, diese riesigen Datenvolumina zu neuen Informationen zu verdichten, anschließend daraus Wissen zu generieren, um letztendlich Weisheit erlangen zu können. Dieser mehrstufige Weg bis zur Nutzung von Big Data ist am Beispiel der Wissenspyramide in Abbildung 1 noch einmal illustrativ dargestellt.

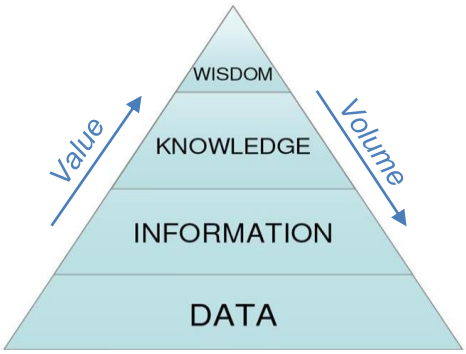

Abbildung 1: The Knowledge Pyramid (Vgl. Landauer, 1998)

1.2 Wissensgenerierung

Ein Schlüssel zur intelligenten Nutzung von Big Data ist das Predictive Modelling, dessen zugrundeliegender Ansatz das fallbasierte Schließen ist. Darunter ist das Lernen aus den

Daten möglichst vieler bekannter Fälle zu verstehen, um anhand der daraus gewonnenen Erkenntnisse für ähnliche Personen Wissen in Form einer Vorhersage zu erhalten.

Die Schwierigkeit in der Erstellung von Vorhersagemodellen auf Basis von Big Data besteht in der Fülle an Einflussfaktoren, die in der Regel äußerst komplex zusammenwirken. Daraus jene auszuwählen, die für eine Prognose relevant sind und diese in geeigneter Weise miteinander in Beziehung zu setzen, ist für den Menschen unter Verwendung von herkömmlichen statistischen Analyseverfahren in einer vernünftigen Zeit nicht mehr realisierbar. Abhilfe kann hier das maschinelle Lernen bieten. Es handelt sich dabei um Lernalgorithmen, die dazu in der Lage sind, auf Basis von Beispieldaten selbstständig zu lernen, zu verallgemeinern und daraus ein Vorhersagemodell zu entwickeln (Abbildung 2). Neben deren Schnelligkeit in der Modellgenerierung besteht eine weitere Besonderheit von maschinellen Lernalgorithmen darin, dass sehr große Datensätze mit vielen Beobachtungen sie nicht überfordern, sondern das Ergebnis noch genauer machen (Kotsiantis et al., 2007).

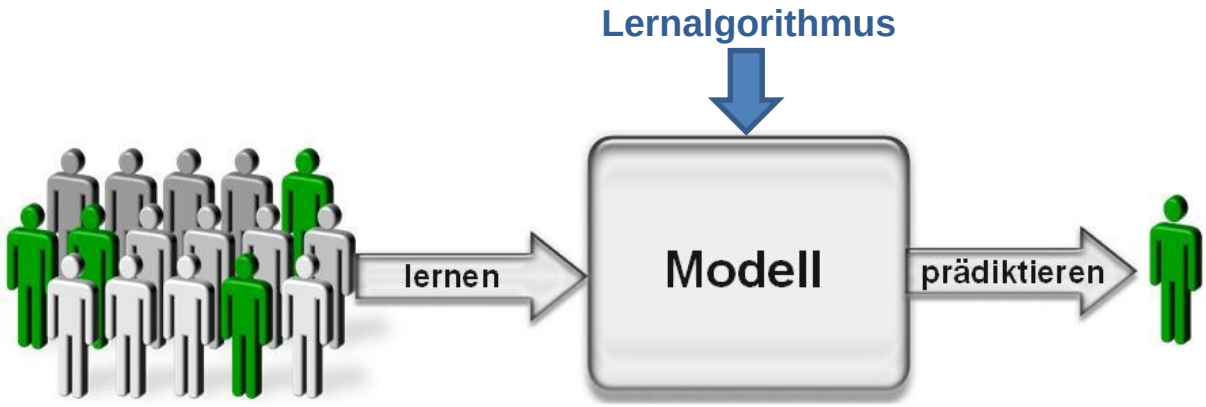

Abbildung 2: Predictive Modelling mit maschinellen Lernverfahren

Wie in der Wissenspyramide dargestellt (Abbildung 3) übernimmt der maschinelle Lernalgorithmus im Prozess der Wissensgenerierung die Aufgabe der Transformation von Informationen in Wissen und löst somit das von John Naisbitt genannte Problem.

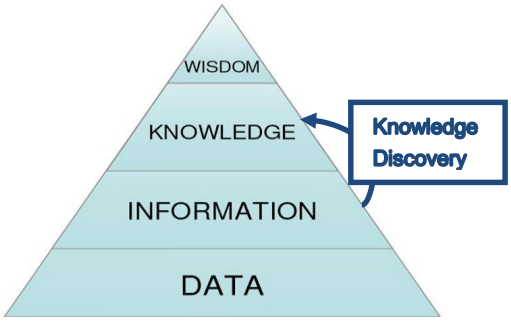

Abbildung 3: The Knowledge Pyramid (Landauer, 1998):
Knowledge Discovery/Predictive Modelling

Diese Informationen müssen dem Algorithmus in Form eines Trainingsdatensatzes zur Verfügungen gestellt werden (Abbildung 4)

Abbildung 4: Maschineller Lernprozess

Die Güte von maschinell generierten Vorhersagemodellen wird wesentlich von der Repräsentation der Trainingsdaten beeinflusst (Pechenizkiy, 2005). Die Aufbereitung dieser Lerndaten stellt jedoch eine besondere Herausforderung dar, weil die optimale Repräsentation von Features von zahlreichen Faktoren abhängt, wie dem verwendeten Lernalgorithmus, der Fragestellung oder dem Stichprobenumfang. Einen Goldstandard gibt es dafür nicht (Guyon, 2006, Hall, 1999, Langley, 1994, Song et al., 2004).

Ein in der Literatur häufig genanntes Problem beim maschinellen Lernen ist der sogenannte „Fluch der Dimensionalität“. Hiermit ist gemeint, dass die Dimensionalität der Daten meist groß im Verhältnis zur Anzahl der Beispiele im Trainingsdatensatz ist. Diese kann dabei in zwei Richtungen gehen – in die Breite, wonach eine große Anzahl an Features ein Problem darstellen kann und in die Tiefe, wonach der Informationsgehalt von Daten in einem Feature mit zu vielen Kategorien unterschätzt werden kann. Besitzt ein Feature sehr viele Ausprägungen und stehen nur wenige Beispiele zur Verfügung, so bleibt der Großteil der Kategorien unbesetzt und somit ohne Information. Algorithmen, die Features beispielsweise auf Basis des maximalen Informationsgewinnes auswählen, stufen in diesem Fall den Informationsgehalt des gesamten Features als gering ein, wodurch indirekt vorhandene, potentiell wertvolle Informationen im Modell unterrepräsentiert sein können. Um deren Einfluss zu erhöhen, ist eine Datentransformation erforderlich, indem aus einem Feature mit vielen Kategorien beispielsweise neue Features mit weniger Kategorien erzeugt werden können. (Pechenizkiy, 2005, Hastie et al., 2009).

In diesem Zusammenhang kommt das Data Mining ins Spiel. Es geht dabei darum, den riesigen unstrukturierten Datenberg von unterschiedlichen Perspektiven aus zu beleuchten und zu nützlichen Informationen abzubauen (Tan et al., 2006) (Abbildung 5).

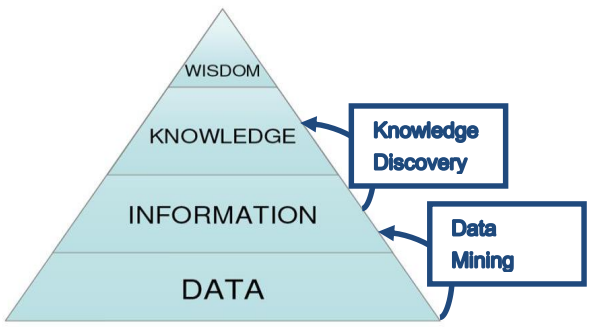

Abbildung 5: The Knowledge Pyramid (Landauer, 1998):
Data Mining zur Unterstützung des Predictive Modellings

Diese Dimensionsreduktion kann ebenso wie das Predictive Modelling maschinell erfolgen. Im Gesundheitsbereich existieren jedoch oftmals komplexe Zusammenhänge innerhalb der Daten, die nur mit Hilfe von zusätzlichen Kompetenzen, wie Erfahrung oder Domänenwissen erkennbar sind und welche nur der Mensch durch ein lebenslanges Lernen mitbringt. Nachdem sich der Mensch zudem durch seine Fähigkeit schnell Muster visuell erkennen zu können auszeichnet, eignen sich Visual Analytics Tools optimal als Schnittstelle zwischen Mensch und Maschine. Sie bieten die Möglichkeit einer interaktiven und explorativen Analyse von umfassenden Datenbeständen. Die daraus erlangten Erkenntnisse können anschließend in Form eines Features dem Algorithmus zur Verfügung gestellt werden, wodurch die menschlichen Stärken in den maschinellen Lernprozess einfließen („Human-In-The-Loop“) und somit verbesserte Modelle erzielt werden können (Abbildung 6).

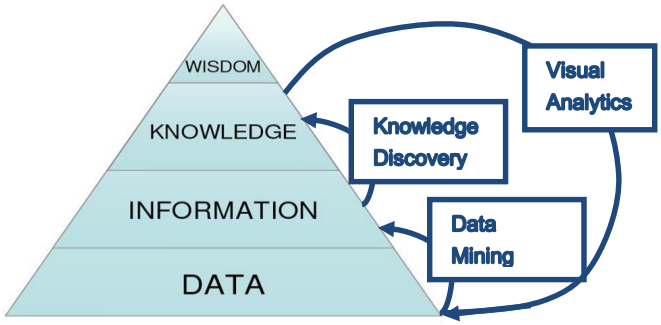

Abbildung 6: The Knowledge Pyramid (Landauer, 1998): Data Mining, Predictive Modelling und Visual Analytics

1.3 Ausgangssituation

Das Austrian Institute of Technology (AIT) ist ein außeruniversitäres Forschungsunternehmen, das sich, unter anderem, im Bereich der Akquisition, Sammlung und Aggregation von Daten im Gesundheitsbereich etabliert hat. Ein relativ neues Aufgabengebiet ist das Predictive Modelling mit maschinellen Lernverfahren.

Ein Ziel das sich das AIT in diesem Sinne gesetzt hat, ist der Aufbau einer unternehmensinternen Predictive Modelling Pipeline, um eine möglichst effiziente Generierung von Vorhersagemodellen für komplexe Fragestellungen aus dem Healthcare-Bereich zu ermöglichen. Eine besondere Anforderung an diese Pipeline ist, dass sie auch von Personen ohne Programmierkenntnisse angewendet werden können soll.

Der erste Teil der Pipeline umfasst die Integration von Daten aus verschiedenen Quellen und wurde bereits erfolgreich umgesetzt. Der zweite Teil ist der analytische Teil. Dieser soll aus einer Modellierungs-Routine bestehen, die sich bereits im Aufbau befindet und weiteren Analysemöglichkeiten, wie der interaktiven Analyse von Variablen mit grundlegenden Methoden der Statistik oder von Daten mit Zeitreihencharakter (Abbildung 7).

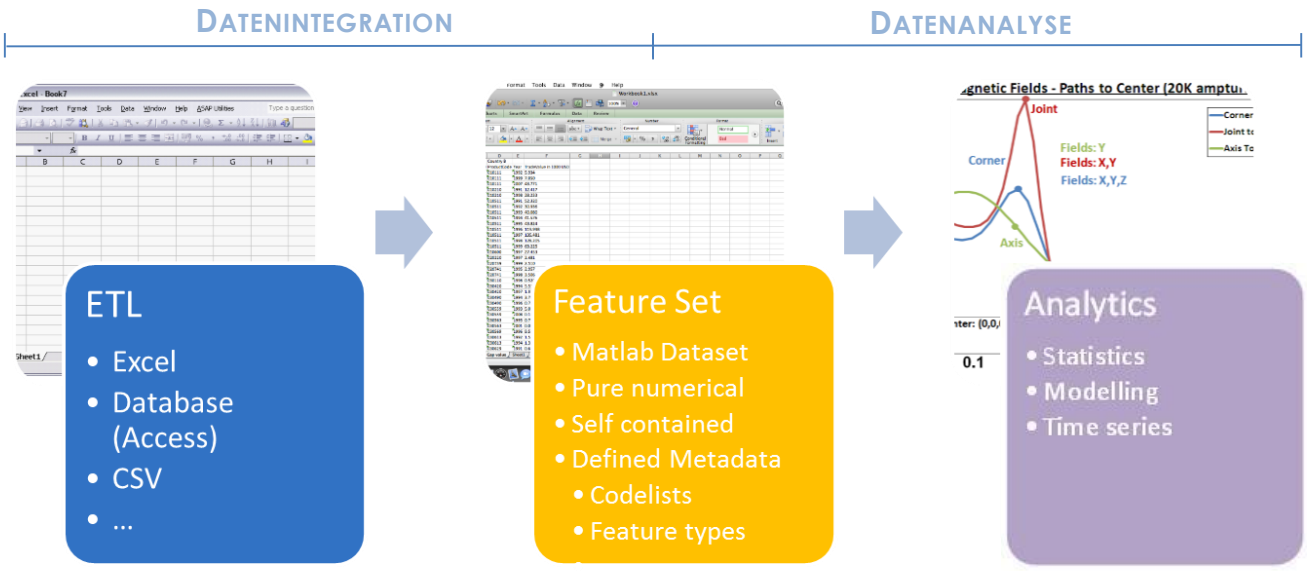

Abbildung 7: Predictive Modelling Pipeline

1.4 Aufgabenstellung

Im Zusammenhang mit dieser Arbeit gab es zwei Teilaufgaben zu bearbeiten.

Feature-Extraktion

Es war ein Konzept zur Aufbereitung von Lerndaten für das Predictive Modelling mit maschinellen Lernverfahren aus codierten Abrechnungsdaten (ICD-10 und DRG Codes) zu entwickeln und zu implementieren.

Visual Analytics

Es galt ein Visual Analytics Framework aufzubauen, in das verschiedene Tools eingebettet werden können. Dieses sollte ein Statistik-Tool beinhalten, mit dem die Variablen des Featuresets mit einfachen Methoden der Statistik schnell analysiert werden können und ein Visual Analytics Tool, welches eine interaktive und explorative Analyse von Daten mit Zeitreihencharakter ermöglicht. Der Schwerpunkt sollte dabei auf das Zeitreihen-Tool gelegt werden. Zudem war für den bereits bestehenden Prozess der Featureset-Generierung ein GUI zu entwickelt und das initiale Layout des Modellierungs-Tools zu gestalten.

1.5 Untersuchungsbereich

Umzusetzen waren diese Aufgaben am Beispiel eines laufenden Projektes, bei dem es darum ging, die Anzahl der Tage vorherzusagen, die Klienten eines australischen Krankenversicherungsunternehmens im nächsten Jahr im Krankenhaus verbringen werden.

Zur Verfügung gestellt wurden Daten von über einer Million Klienten für den Zeitraum 2010 bis 2012. Das Ziel war, auf Basis der Daten von 2010 und 2011 die Anzahl der Tage zu prädiktieren, die jeder Klient im Jahr 2012 im Krankenhaus verbringen wird (Abbildung 8).

Abbildung 8: Prädiktionsvorhaben

Das Datenkontingent setzt sich wie folgt zusammen:

- Claims-Table (Demographie und Versicherungsdaten)
- Procedure-Table (erhaltene Leistungen während einer Hospitalisierung)
- Admission-Table (Hauptdiagnose, Nebendiagnosen, …)

Zur Validierung der Tools wurden pseudonymisierte Daten der beiden österreichischen Benchmark-Studien zur Optimierung des Einsatzes von Blutkomponenten und aus dem Telemonitoring-Programm „Gesundheitsdialog Diabetes“ herangezogen.

2 Methoden

Sowohl für die Feature-Extraktion als auch für die Entwicklung des Visual Analytics Frameworks und der darin eingebetteten Tools wurde entsprechend der Vorgabe MATLAB (Mathworks Inc., Natick, USA) in der Version 2013b verwendet.

2.1 Unternehmensinterne Strukturen und Prozesse

2.1.1 Datenstruktur

Die Daten wurden pseudonymisiert und in Form von MATLAB-Datasets vom Betreuer zur Verfügung gestellt. Hierbei ist zwischen Rohdatensätzen und dem übergeordneten Featureset zu unterscheiden (Abbildung 9). Letzteres beinhaltet die aus den Rohdaten weiterverarbeiteten Lerndaten, welche zum Trainieren des maschinellen Lernalgorithmus verwendet werden und Features[1] in aggregierter Form beinhalten. Die Aggregation unterliegt hierbei einem Bin-Konzept. Wird für das Experiment beispielsweise der Zeitraum von 01. Jänner

[1] Ein Feature beschreibt die Eigenschaften eines Klienten, wie beispielsweise Alter, Geschlecht oder Anzahl der Krankenhaustage im Vorjahr. Synonyme Begriffe sind Merkmal, Variable oder Attribut.

2010 bis 31. Dezember 2013 betrachtet und die ersten drei Jahre zum Lernen verwendet, um für das vierte Jahr eine Prädiktion durchzuführen, so gibt es insgesamt vier Bins, wobei ein Bin jeweils den Zeitraum zwischen Anfang und Ende eines Jahres charakterisiert. Wird eine Aggregation durchgeführt, so werden die Werte einer Variable zu einem Bin akkumuliert. War ein Klient beispielsweise im Jahr 2011 dreimal im Krankenhaus für jeweils zwei Tage, so enthält das aggregierte Feature „Days in Hospital“ in der Zeile mit dem Bin für das Jahr 2011 den Eintrag 6.
Im Gegensatz zum Featureset kann es beliebig viele Rohdatensätze geben und jeder Klient kann zwischen 0 und N Zeilen pro Rohdatensatz in Anspruch nehmen. Diese Daten werden ausschließlich für die Zeitreihendarstellung im Zeitreihen-Tool benötigt.

Abbildung 9 zeigt die Hierarchie der dieser Arbeit zugrundeliegenden Datenstruktur.

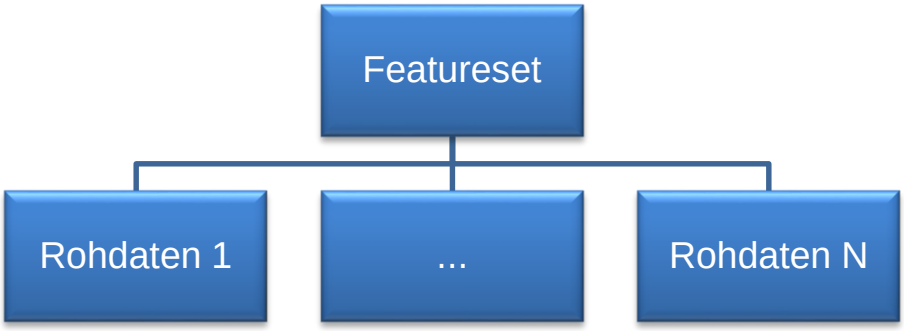

Abbildung 9: Hierarchie der Datenstruktur

Allen Datasets gemein ist deren Aufbau. Es handelt sich dabei um MATLAB-Datasets, deren Zeilen Beobachtungen (Klienten, Patienten) und deren Spalten Merkmale (z.B. Alter, Hauptdiagnose) repräsentieren. Alle Variablen sind von rein numerischer Natur. Die Kategorien sind durch ganzzahlige Ziffern ersetzt und deren zugehörige Bezeichnung wird in den Metadaten-UserData des Datasets mitgeführt. Die Metadaten sind über die Eingabe des Befehls *F.Properties* in das MATLAB Command Window abrufbar, wobei F hier beispielhaft für den Variablennamen des Featuresets steht (Abbildung 10, Abbildung 11).

Abbildung 10 stellt einen Ausschnitt eines MATLAB-Datasets dar.

F <31122x978 dataset>

	1 CustomerID	2 Bin	3 ADMISSION_PREV	4 AMT_CHARGED_rndlog10	5 AMT_CHARGED_ACTUAL_rndlog10	6 AMT_CHARGED_iqr_rndlog10
1	8	1	0	4	4	1
2	8	2	0	3	3	1
3	8	3	0	3	3	2
4	8	4	19371046	3	3	2
5	10	1	0	3	3	2
6	10	3	0	3	3	2
7	10	4	0	4	4	2
8	15	4	0	4	4	2
9	28	3	0	3	3	3
10	31	2	0	3	3	2
11	35	2	0	3	3	2
12	35	3	0	2	2	0
13	42	1	0	4	4	3

Abbildung 10: MATLAB-Dataset

Abbildung 11 zeigt am Beispiel des Featuresets, wie die Metadaten eines MATLAB-Datasets abgerufen werden können.

```
>> F.Properties

ans =

       Description: [1x1 struct]
    VarDescription: {1x978 cell}
             Units: {1x978 cell}
          DimNames: {'Observations'  'Variables'}
          UserData: [1x1 struct]
          ObsNames: {}
          VarNames: {1x978 cell}
```

```
>> F.Properties.UserData

ans =

            Age10: {11x1 cell}
            Age60: {3x1 cell}
    CLIENT_STATUS: {5x1 cell}
          LOG2DIH: {12x1 cell}
      MEMBER_TYPE: {6x1 cell}
        PRODUCT_2: {223x1 cell}
        PRODUCT_3: {64x1 cell}
    PRODUCT_GROUP: {6x1 cell}
```

Abbildung 11: Metadaten des Featuresets: UserData mit Kategorie-Bezeichnungen

2.1.2 Routinen der prä-existenten Predictive Modelling Pipeline

Die bestehende Modellierungs-Pipeline setzt sich aus drei Routinen zusammen (Abbildung 12). Die erste umfasst das Erzeugen eines Trainingsdatensatzes für den maschinellen Lernalgorithmus. In der zweiten geht es um die Generierung eines Vorhersagemodells und in der dritten um die Bewertung dessen Güte.

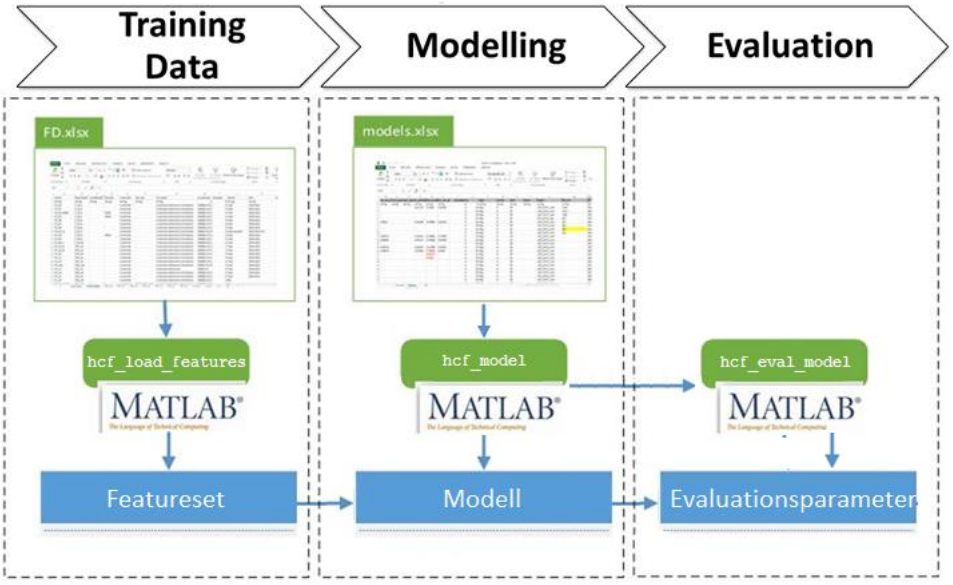

Abbildung 12: Bestehende Routinen der unternehmensinternen Predictive Modelling Pipeline

Die Featureset-Generierungs-Routine besteht aus einer „Feature-Definition-Table“ (Excel-File) und der MATLAB-Funktion `hcf_load_features`. Soll ein neuer Trainingsdatensatz erzeugt werden, so muss in der Feature-Definition-Table ein neuer Eintrag mit dem Namen und den Parametern des Featuresets vorgenommen werden. Anschließend wird in MATLAB die Funktion `hcf_load_features` aufgerufen, um den Prozess zur Generierung des Fea-

turesets zu starten. Das fertige Featureset wird automatisch gespeichert und eine standardisierte Ordnerstruktur (Abbildung 13) mit dem Namen des Featuresets im gleichen Pfad angelegt. In diesem Ordner befinden sich die Rohdatensätze, aus denen sich das Featureset zusammensetzt und ein weiterer - zu diesem Zeitpunkt leerer - Ordner „models“.

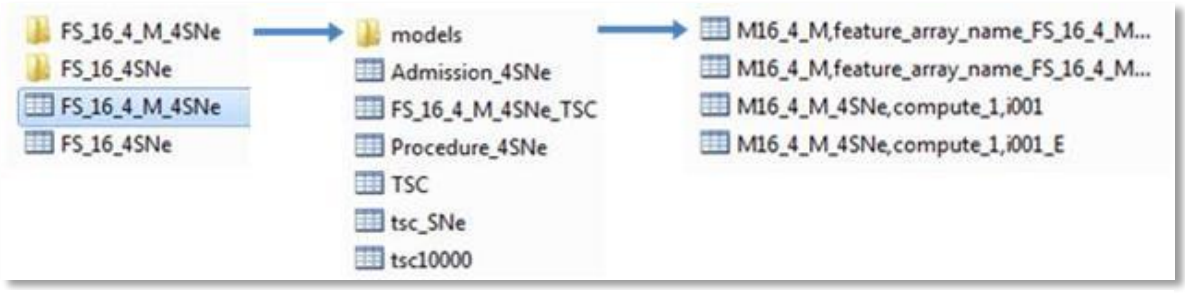

Abbildung 13: Standardisierte Ordnerstruktur der Predictive Modelling Pipeline

Auch die Modellierungs-Routine setzt sich aus einer Konfigurationsdatei und einer MATLAB-Funktion `hcf_model` zusammen. In der Konfigurationsdatei können der Lernalgorithmus ausgewählt und dessen Parameter eingestellt werden. Durch Aufrufen der Funktion `hcf_model` wird der Modellierungsvorgang gestartet und anschließend automatisch die Evaluierungs-Routine durch Aufrufen der Funktion `hcf_eval_model` ausgelöst. Die Ausgabedateien werden anschließend im „model“-Ordner des für die Modellierung verwendeten Featuresets gespeichert. Es werden dabei immer zwei Dateien erstellt. Eine enthält das Objekt „Lernalgorithmus“ (z.B. aus der Verwendung der TreeBagger-Funktion) und das andere File beinhaltet die Evaluationsergebnisse und eine Struktur „models“, welche die Modellergebnisse und Informationen über den Trainings- und den Testprozess beinhaltet.

2.2 Feature-Extraktion

2.2.1 ICD-10 Codes: Hauptdiagnose

Für das Ableiten von Features aus den ICD-10 Codes wurde die von der WHO herausgegebene ICD-10 Systematik aus dem Jahr 2010 in einem Excel-Dokument nachgebildet (WHO-apps, 2010). Es wurde dabei eine Einschränkung auf die ersten drei Stellen des Codes – der Basisklassifikation - vorgenommen.

Für Features von Level 1 und Level 2 (siehe 3.1.2) gilt, dass jedes Excel-Blatt ein neues Feature repräsentiert, wobei der Name des Blattes zugleich den Namen des Features kennzeichnet. Dieses beinhaltet die von 1 bis N durchnummerierten Codegruppen, die dazugehörige Kategorie-Bezeichnung, sowie das Textmuster, welches von der MATLAB-Funktion zur Kategorisierung benötigt wird.

Level-3-Features wurden aufgrund ihrer großen Anzahl nach deren Zugehörigkeit zu einem Level-2-Feature in einem Excel-Sheet zusammengefasst, dessen Name sich aus dem Level-2-Feature-Namen und der Erweiterung „_L3“ zusammensetzt (z.B. ICD10_INFECTION_L3). Jede Spalte dieses Excel-Sheets repräsentiert ein neues Feature und wird nach dem Muster „Level-2-Feature-Name_L3_1 bis N“ benannt, wobei N für die Anzahl der Spalten bzw. Features im jeweiligen Excel-Blatt steht.

Es wurde eine MATLAB-Funktion geschrieben, die das gesamte Excel-Dokument einliest und die Excel-Sheets mit Hilfe von MATLABs Built-In-Funktion `regexprep` in Features transformiert. Dabei werden die Codes aus den Rohdaten mit dem Textmuster der jeweiligen Kategorie abgeglichen und im Falle einer Übereinstimmung mit der entsprechenden Nummer versehen, die der Kategorie angehört. Sollte der Code keiner Codegruppe zuordenbar sein, so wird er für dieses Feature als fehlender Eintrag gewertet und durch „NaN“ ersetzt. Die Bezeichnung der Kategorien wird den Metadaten des Featuresets hinzugefügt.

Abbildung 14 illustriert das Prinzip der Verwendung von MATLABs „regular expressions“ am Beispiel des Überleitens des ursprünglichen Features für ICD-10 Hauptdiagnosecodes in ein neues Feature mit weitaus weniger Kategorien.

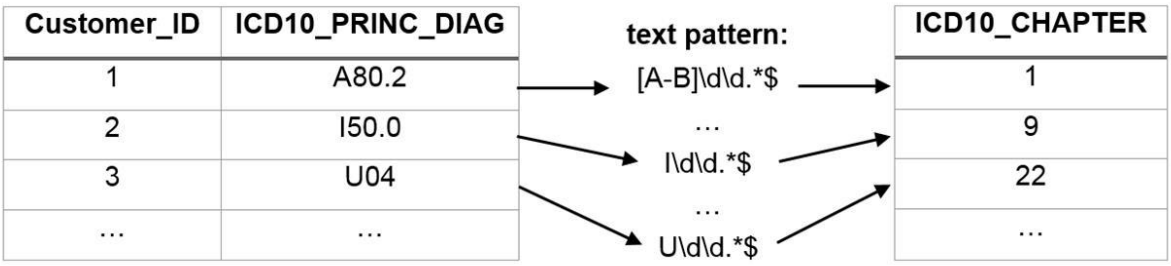

Abbildung 14: ICD-10 Feature-Extraktion

Jedem Klienten (Customer_ID) ist eine Hauptdiagnose in Form eines ICD-10 Codes (ICD10_PRINC_DIAG) zugeordnet. Dieses Feature wird in ein neues Feature namens ICD10_CHAPTER überführt. Dessen erste Kategorie steht für Infektionserkrankungen und schließt alle Codes ein, deren ersten drei Stellen dem Muster A00-B99 folgen. Dem Klienten 1 mit der Hauptdiagnose A80.2 wird somit die Zahl 1 zugewiesen.

2.2.2 ICD-10 Codes: Nebendiagnosen

Für die Berechnung des Charlson Komorbiditäts-Scores wurde ebenfalls ein Excel-Blatt angelegt. Dieses enthält die vordefinierten Erkrankungen, sowohl das dazugehörige ursprüngliche als auch das zeitgerechtere Charlson-Gewicht, sowie die mit den Erkrankungen assoziierten ICD-10 Codes und die dazugehörigen Textmuster (Sundararajan et al., 2004, Quan et al., 2011).

Die Umwandlung des Excel-Sheets in ein Feature unter Verwendung von MATLAB erfolgte durch eine entsprechende Funktionalität der vorhandenen Software.

2.2.3 DRG Codes

Die Transformation des theoretischen DRG-Feature-Konzeptes in praktische DRG-Features mit Hilfe von MATLAB wurde in der Anfangsphase von anderen Teammitgliedern durchgeführt. Dazu zählt das DRG-Level-1-Feature (Major Diagnostic Categories), sowie jene, die keiner hierarchischen Struktur unterliegen.

Für die Feature-Extraktion von der zweiten und dritten Hierarchieebene der DRG Codes in der zweiten Phase wurde je ein Excel-Blatt angelegt, wobei das Muster jenem entspricht, das für ICD-10-Features der ersten beiden Levels erarbeitet wurde. Auf diese Weise konnte die MATLAB-Funktion zum Generieren von ICD-10-Features bis auf wenige Anpassungen auch hier zur Anwendung gebracht werden.

2.3 Visual Analytics

2.3.1 Allgemeine Grundlagen zur GUI-Programmierung mit MATLAB

Die Erstellung der graphischen Benutzeroberflächen (GUIs) erfolgte primär mit Hilfe von MATLABs „Graphical Userinterface Developement Environement (GUIDE)“, welche eine interaktive Gestaltung von GUIs ermöglicht. Es werden ein Fig-File und ein M-File ausgegeben, wobei ersteres das Design des GUIs enthält und letzteres die Funktionen beinhaltet, die den einzelnen Objekten zugehören und beliebig programmierbar sind. Eine entscheidende Funktion ist die sogenannte Callback-Funktion. In ihr wird festgelegt, was das Objekt ausführen soll, wenn es durch Anklicken aktiviert wird.
Jedem Objekt (Pushbutton, Listbox, …) wird eine eindeutige Zahl (= handle) zugewiesen mit der es identifiziert werden kann. Handles werden in einer handles-Struktur gespeichert, welche den Funktionen jedes Objektes automatisch als Inputparameter übergeben wird.

Eine übersichtliche Einführung in die Welt der GUI-Programmierung mit MATLAB bietet sowohl MATLABs mitgelieferte „Documentation“ als auch das Buch „Learning to Program with MATLAB: Building GUI Tools“ (Lent, 2013).[2]

[2] Sollten die dokumentierten Funktionalitäten der GUI-Programmierung nicht ausreichen, so könnte das Buch „Undocumented Secrets of MATLAB – Java Programming“ ALTMAN, Y. M. 2011. *Undocumented secrets of MATLAB-Java programming*, CRC Press. Abhilfe leisten.

2.3.2 Framework (siehe 3.2.1)

Der Austausch von Daten zwischen den Modulen wird über die Callback-Funktionen der Tabs gesteuert (Abbildung 29). Wird ein neues Modul durch Betätigen des entsprechenden Tabs geöffnet, so wird die einheitliche Datenstruktur aus der handles-Struktur des aktuell verwendeten Tools ausgelesen und mit Hilfe von MATLABs „application data“ in die handles-Struktur des neuen Tools übertragen. Das Einbinden eines neuen Moduls kann auf einfache Weise durch Hinzufügen eines weiteren Tabs erfolgen.

2.3.3 Data-Tool (siehe 3.2.2)

2.3.3.1 Grundlegende Prinzipien

Beim Öffnen des GUIhome wird in dessen Opening-Funktion[3] eine Struktur angelegt, in der das Featureset (F), die Rohdatensätze (R) und - im Falle der Verwendung des Zeitreihen-Tools – auch die Files mit den Modellergebnissen (M), sowie eine Zeitreihenkollektion (time series collection - tsc) gespeichert werden können (Abbildung 15).

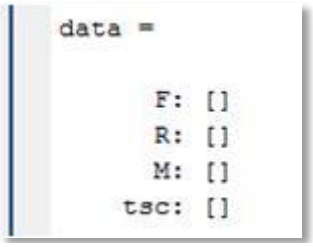

Abbildung 15: Struktur zur Verwaltung von Daten des Frameworks

Diese Struktur wird dem Fig-File des GUIhome mit Hilfe von MATLABs „application data“ angehängt. Das handle des GUIs wird in der Root gespeichert, um anderen Tools einen Zugriff auf diese Datenstruktur zu gewähren.

Abbildung 16 veranschaulicht die Funktionsweise von MATLABs „application data“. Der erste Teil umfasst das Prinzip des Speicherns und der zweite Teil stellt das Prinzip des Abrufens dar.

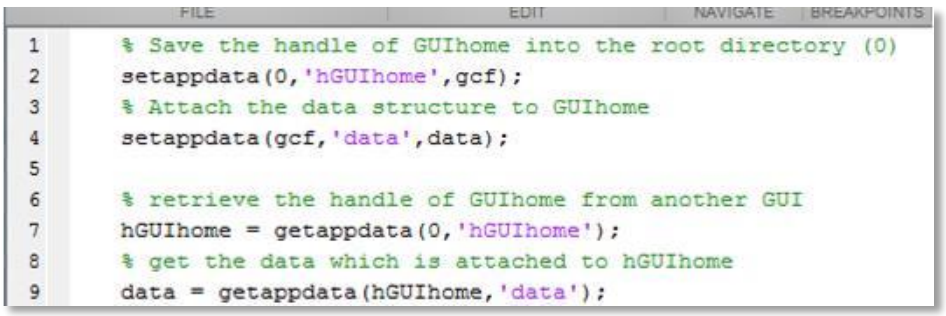

Abbildung 16: Matlabs „application data“: Speichern und Abrufen

[3] Die Opening-Funktion wird bei der Verwendung von GUIDE automatisch generiert. Sie wird ausgeführt, bevor der Benutzer das GUI zu Gesicht bekommt.

2.3.3.2 Lade-Kaskade

Das automatische Laden von Dateien, die einem Featureset zugehören, wird in der „`Update-GUIhomeData`"-Funktion geregelt. Mit Hilfe dessen Metadaten-Levels (Abbildung 17), die angeben aus welchen Rohdatensätzen sich das Featureset zusammensetzt, können im Ordner des Featuresets (Abbildung 13) die Rohdatensätze von den anderen Dateien unterschieden werden. Zeitreihenkollektionen werden als solche erkannt, wenn der Filename „TSC" oder „tsc" beinhaltet. Aus dem Ordner „models" finden nur Files mit der „models"-Struktur Verwendung. Die anderen Dateien werden beim Laden ignoriert. Nachdem sich ein Modell aus mehreren Modellen zusammensetzen kann, werden alle Teilmodelle als eigenständige Modelle geladen, um auch diese getrennt voneinander analysieren zu können. Für alle Listboxen wurde eine Keypress-Function programmiert, die es ermöglicht, eine irrtümlicherweise geladene Datei wieder aus der Datenstruktur zu entfernen.

Abbildung 17: Metadaten des Featuresets: Rohdatensätze

2.3.3.3 Erzeugen eines Featuresets (GUIcreateFeatureSet)

Zum Erstellen eines neuen Featuresets wurde für die in 2.1.2 vorgestellte, bereits bestehende Featureset-Generierungs-Routine ein eigenständiges GUI (Abbildung 18) entwickelt. Im GUI können die Inputparameter für die Funktion `hcf_load_features` festgelegt werden. In der Callback-Funktion des „Create Featureset"-Pushbuttons findet die Übergabe der Inputparameter statt, woraufhin die Funktion `hcf_load_features` aufgerufen und der Prozess zur Featureset-Generierung gestartet wird.

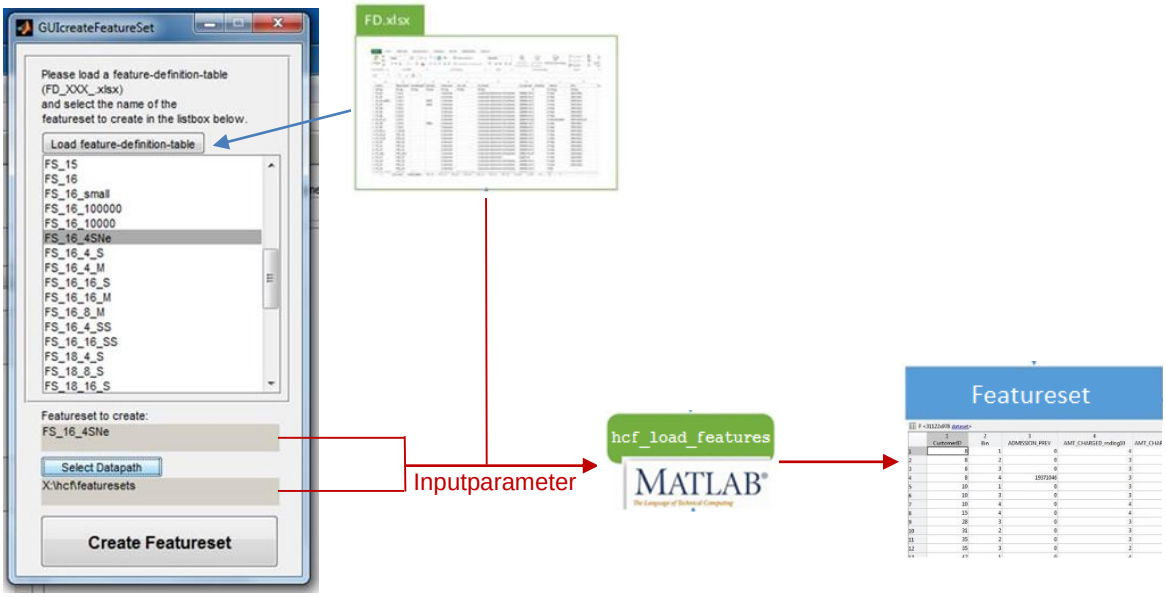

Abbildung 18: GUIcreateFeatureSet für die prä-existente Featureset-Generierungs-Routine

2.3.3.4 Erzeugen einer Zeitreihenkollektion (GUIcreateTSCobj)

Im Zeitreihen-Tool steht der Patient im Mittelpunkt, weshalb die verschiedenen Datenquellen nach Einträgen zum aktuell gewählten Patienten durchsucht werden müssen. Nachdem es sich dabei um einen relativ zeitintensiven Prozess handelt, wird dieser Prozessschritt vorgelagert, indem für jeden Patienten ein Zeitreihenobjekt angelegt wird. Dieses enthält Informationen darüber, welche Zeilen der Patient im Featureset und in den dazugehörigen Rohdatensätzen in Anspruch nimmt.

Im Popupmenu des GUIs zur interaktiven Generierung von Zeitreihenobjekten (*GUIcreateTSCobj,* Abbildung 34) werden nur jene Variablen zur Auswahl zur Verfügung gestellt, die vom Typ „ID – Identifier“ oder „Key – Schlüsselvariable“ sind (hier die Variablen „Klienten“ oder „Patienten“). Demzufolge erfolgt eine Vorselektion der Variablen mit Hilfe der Metadaten-Units („I“) und der Metadaten-Keys des Featuresets. Bei Klicken des „Create TSC“-Pushbuttons wird in den Datensätzen nach Einträgen zum jeweiligen Klienten gesucht und die Zeilenindizes dem entsprechenden Zeitreihenobjekt hinzugefügt. Jedem Zeitreihenobjekt wird ein Name zugeordnet, der sich aus „$xID_{i=(1:n)}$“ ergibt, wobei i für das i-te und n für die maximale Anzahl an zu generierenden Objekten steht (z.B. x329, wenn der Patient die ID „329“ besitzt).

Abbildung 19 veranschaulicht die Struktur einer Zeitreihenkollektion (tsc) und der darin enthaltenen Zeitreihenobjekte am Beispiel der Daten des Gesundheitsdialog-Diabetes-Projektes.

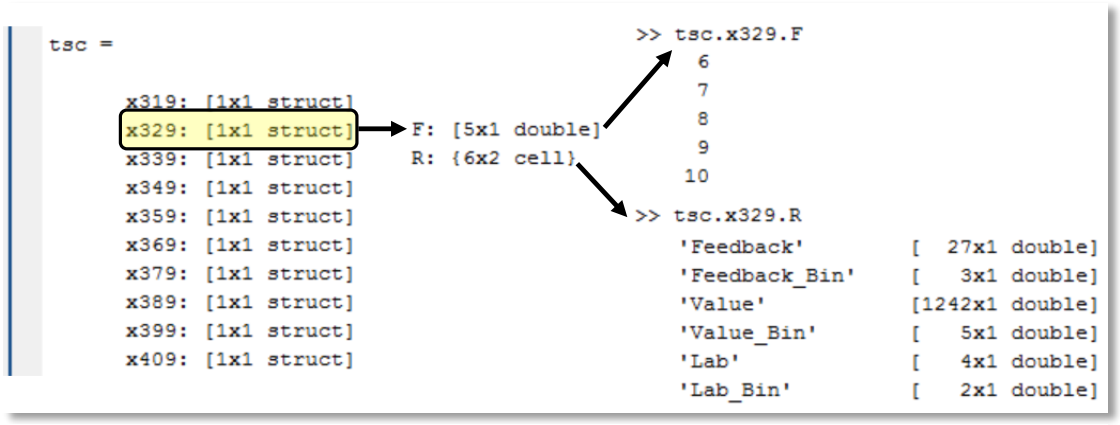

Abbildung 19: Zeitreihenkollektion (tsc) mit den Zeitreihenobjekten

2.3.4 Zeitreihen-Tool (siehe 3.2.3)

Die Entwicklung des Zeitreihen-Tools erfolgte in zwei Phasen. In der ersten wurde ein Prototyp (Abbildung 20) entwickelt, um die Machbarkeit festzustellen und zu bewältigende Probleme zu identifizierten. Dieses Tool wurde auf die „Admission-Table“ aufgebaut, welche Hintergrundinformationen über den jeweiligen Krankenhausaufenthalt eines Klienten beinhaltet. In einer zweiten Phase wurde das in dieser Arbeit vorgestellte Zeitreihen-Tool entwickelt. Dieses baut auf eine allgemeine Datenstruktur auf und wird primär von den Metadaten des Featuresets bestimmt.

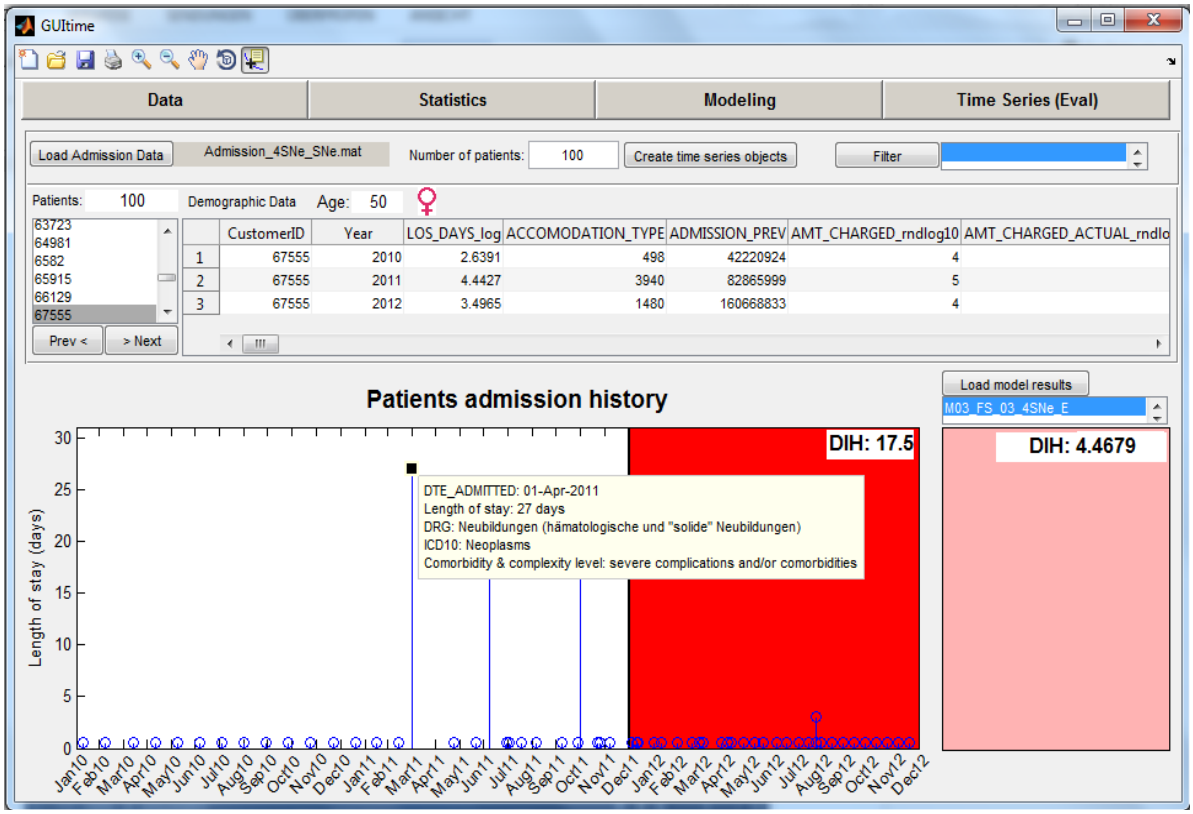

Abbildung 20: Zeitreihen-Tool: Prototyp

2.3.4.1 Grundlegende Prinzipien

Bevor das Zeitreihen-Tool für den Anwender sichtbar ist, werden mit Hilfe der Funktion `CreateAxesObjects` Achsenobjekte generiert und deren handles in der handles-Struktur des GUItime gespeichert. Es wird dabei zwischen Zeitreihen- und Modellachsenobjekten unterschieden. Wird die Variable „maxNumberOfAxes“ in der `CreateAxesObjects`-Funktion nicht verändert, so können bis zu zehn Zeitreihenebenen, welche sich aus je einem Zeitreihen- und einem Modellachsenobjekt zusammensetzen, gleichzeitig visualisiert werden. Wenn der User weniger als zehn Ebenen betrachten möchte, so werden die anderen Objekte nicht zerstört, sondern lediglich unsichtbar gemacht. Die sichtbaren Achsen werden automatisch optimal im jeweiligen Achsenpanel positioniert.

Jedem Achsenobjekt werden mit Hilfe von MATLABs „application data“ Metadaten in Form einer Struktur angehängt, welche Informationen darüber enthalten, was in der jeweiligen Achse visualisiert werden soll (Abbildung 21).

```
>> getappdata(handles.TimeSeriesAxesHandles(1),...
'TimeSeriesAxisMetaData')
        RawDataSetName: []
                     R: []
                     x: []
                     y: []
    DataCursorVariables: []
                  XLim: [734139 735600]
                 XTick: [734139 734504 734869 735235 735600]
             XTickLabel: [5x11 char]
                  YLim: []
                 YTick: []
             YTickLabel: []
```

```
>> getappdata(handles.ModelResultsAxesHandles(1),...
'ModelResultsAxisMetaData')
    ModelResultsFileName: []
                       S: []
                  target: []
                      TP: [0x0 dataset]
```

Abbildung 21: Metadaten der Zeitreihenachsen (links) und der Modellachsen (rechts)

Die Metadaten einer Zeitreihenachse umfassen den Namen des Datensatzes, aus dem die Daten zur Visualisierung stammen, sowie die Namen der Variablen, die auf der x- und der y-Achse dargestellt werden sollen. „DataCursorVariables“ ist eine Liste von Variablen aus diesem Datensatz, die angezeigt werden, wenn der User auf ein Event in der Zeitreihenachse klickt. „XLim“, „XTick“ und „XTickLabel“ werden für die Skalierung der x-Achse benötigt. Diese Informationen werden von den Metadaten des Featuresets automatisch ausgelesen und in die Metadaten der Achsenobjekte übertragen (Abbildung 22). „YLim“, „YTick“ und „YTickLabel“ finden in dieser Version des Zeitreihentools keine Verwendung. Sie sind in der Metadaten-Struktur enthalten, um zukünftig ein GUI bereitstellen zu können, welches eine benutzerdefinierte Skalierung der y-Achse des jeweiligen Achsenobjektes zulässt.

```
>> F.Properties.Description.Bin

ans =

     pivot: 735600
       num: 4
      unit: 'Y'
       int: 1
    length: 1
        id: [1 2 3 4]
       val: [1 2 3 4]
     start: [734139 734504 734869 735235]
       end: [734504 734869 735235 735600]
      name: {'2010'  '2011'  '2012'  '2013'}
```

```
>> getappdata(handles.TimeSeriesAxesHandles(1),'TimeSeriesAxisMetaData')

ans =

        RawDataSetName: []
                     R: []
                     x: []
                     y: []
    DataCursorVariables: []
                  XLim: [734139 735600]
                 XTick: [734139 734504 734869 735235 735600]
             XTickLabel: [5x11 char]
                  YLim: []
                 YTick: []
             YTickLabel: []
```

Abbildung 22: Informationstransfer zwischen Featureset und Metadaten der Zeitreihenachsenobjekte

In den Metadaten eines Modellachsenobjektes wird der Name der Datei des zu darstellenden Modells festgehalten, sowie eine Struktur „S“, welche Informationen über Trainings- und Testperiode des ausgewählten Modells beinhaltet. Zudem wird unter „target“ der Name der aus dem Featureset stammenden Prädiktionsvariable gespeichert und deren wahren und prädiktierten Werte für jeden Patienten im Datensatz „TP“ zusammengefasst.

Mit der Funktion `VisualizeTimeSeriesMetaData` werden die Metadaten des jeweiligen Zeitreihenobjektes abgerufen und visualisiert. Für Modellachsenobjekte ist die Funktion `VisualizeModelResultsMetaData` das entsprechende Äquivalent. Klickt der User auf ein Event in der Zeitreihenachse, so werden die ausgewählten Data-Cursor-Variablen mit Hilfe der Funktion `DataCursorUpdateFcn` in einem Datatip angezeigt.

2.3.4.2 Visualisieren einer Zeitreihe (GUITimeSeriesAxis)

Durch Doppelklick auf eine der Zeitreihenachsen wird die Funktion `TSAxisMouseClick` ausgeführt, in der das handle der Achse dem *GUITimeSeriesAxis* übergeben und dieses dann geöffnet wird. Die verfügbaren Rohdatensätze werden automatisch in der Opening-Funktion des GUIs geladen und dem Popupmenu zur Anzeige übergeben. Per Default werden die Variablen des ersten Rohdatensatzes der Datenstruktur des Frameworks in den Listboxen zur Auswahl zur Verfügung gestellt. Für die x-Achse werden nur Variablen angezeigt, die laut den Metadaten-Units des Rohdatensatzes vom Typ „T“ (Time) sind. Y-Achsen-Variablen können alle Typen annehmen außer „I“ (Identifier) oder „T“ (Time). Die ausgewählten Parameter werden durch Betätigen des „Apply“-Pushbuttons an die Metadatenstruktur der Zeitreihenachse übergeben (Abbildung 23), welche daraufhin visualisiert wird.

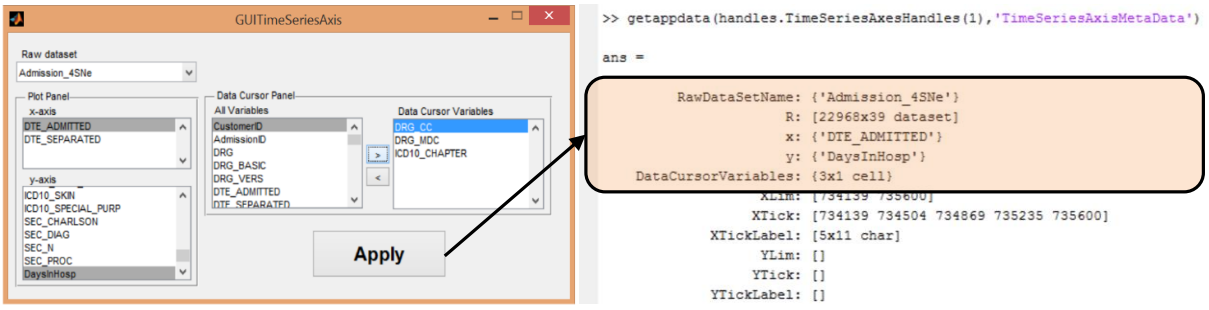

Abbildung 23: Informationstransfer zwischen GUITimeSeriesAxis und Metadaten der Zeitreihenachsenobjekte

2.3.4.3 Visualisieren der Modellergebnisse

Führt der Anwender einen Doppelklick in einer der Modellachse aus, so wird die Funktion `MRAxisMouseClick` aufgerufen. In dieser wird MATLABs Built-in-Funktion `listdlg` zur Anzeige einer Dialogbox verwendet. Die Dateien mit den Modellergebnissen werden aus der Datenstruktur des Zeitreihen-Tools ausgelesen und in der Dialogbox angezeigt. Die Informa-

tionen über das ausgewählte Modell werden schließlich in den Metadaten des Modellachsenobjektes gespeichert (Abbildung 24).

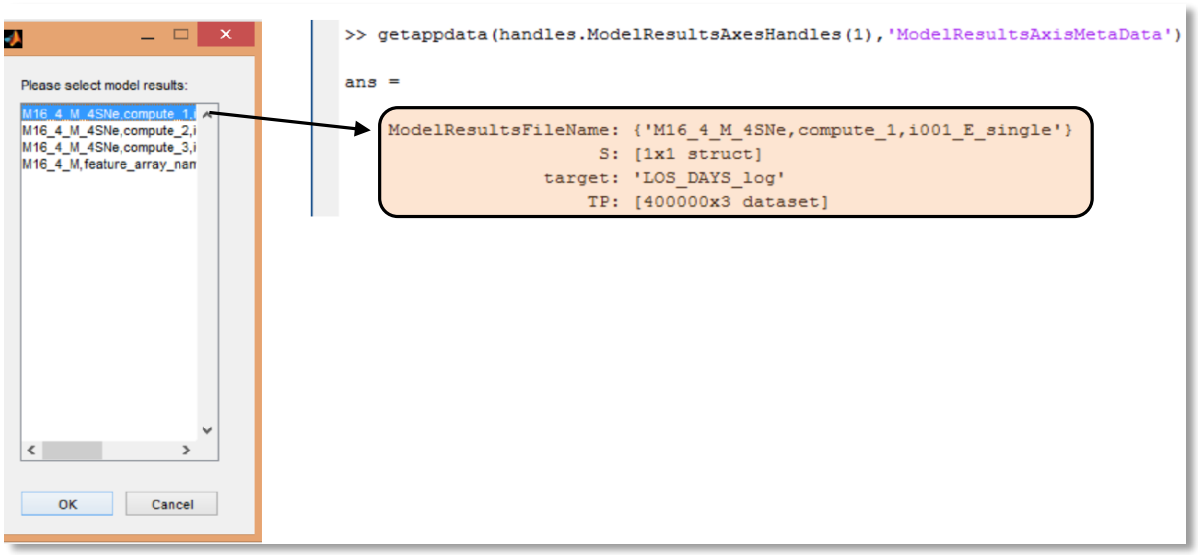

Abbildung 24: Informationstransfer zu den Metadaten der Modellachsenobjekte

2.3.4.4 Filter

Beim Öffnen des *GUITimeSeriesFilter* (Abbildung 41) wird die Datenstruktur des Zeitreihen-Tools geladen und das Featureset und die Rohdatensätze werden daraufhin ins Popupmenu eingespielt. Wird vom Anwender ein Dataset ausgewählt, so werden dessen Metadaten-UserData ausgelesen und die Feldnamen dieser Struktur in der „Categories“-Listbox angezeigt. Wird eine dieser Kategorien in der Listbox markiert, so werden die zugehörigen Subkategorien in die „Sub-Categories“-Listbox eingespielt. Durch Aktivieren des „>“-Pushbuttons werden die markierten Subkategorien in die Filter-Listbox übertragen. Um den zeitintensiven Filtervorgang zu beschleunigen, wird bereits bei der Betätigung des „>“-Buttons eine Teilfilterung durchgeführt, indem im ausgewählten Dataset nach Patienten gesucht wird, die den aktuell gewählten Filterkriterien entsprechen. Die Namen der Patienten (IDs) werden in den UserData der „Filter“-Listbox zwischengespeichert (Abbildung 25). Dabei wird für jede gewählte Kategorie eine neue Zeile angelegt. Wird nach mehreren Subkategorien gefiltert, so wird für jede Subkategorie eine Spalte hinzugefügt und die IDs der Patienten, die das Kriterium der jeweiligen Subkategorie erfüllen, werden in der zugehörigen Spalte eingetragen. Wird ein MATLAB-Code zur Erstellung eines Filterkriteriums verwendet, so wird für dieses Kriterium eine neue Zeile angelegt.

Abbildung 25 veranschaulicht das Cellarray mit den Teilergebnissen des Filterprozesses, welches in den UserData der Filter-Listbox gespeichert wird. In diesem Fall wird nach Tumorpatienten gefiltert, die jünger als 20 Jahre sind und im Prädiktionszeitraum (Bin 3) min-

destens einen halben Tag im Krankenhaus verbracht haben. „<10 (Age10)“ bedeutet in diesem Beispiel 0-9 Jahre und „<20 (Age10)“ steht für die Subkategorie 10 bis 19 Jahre.

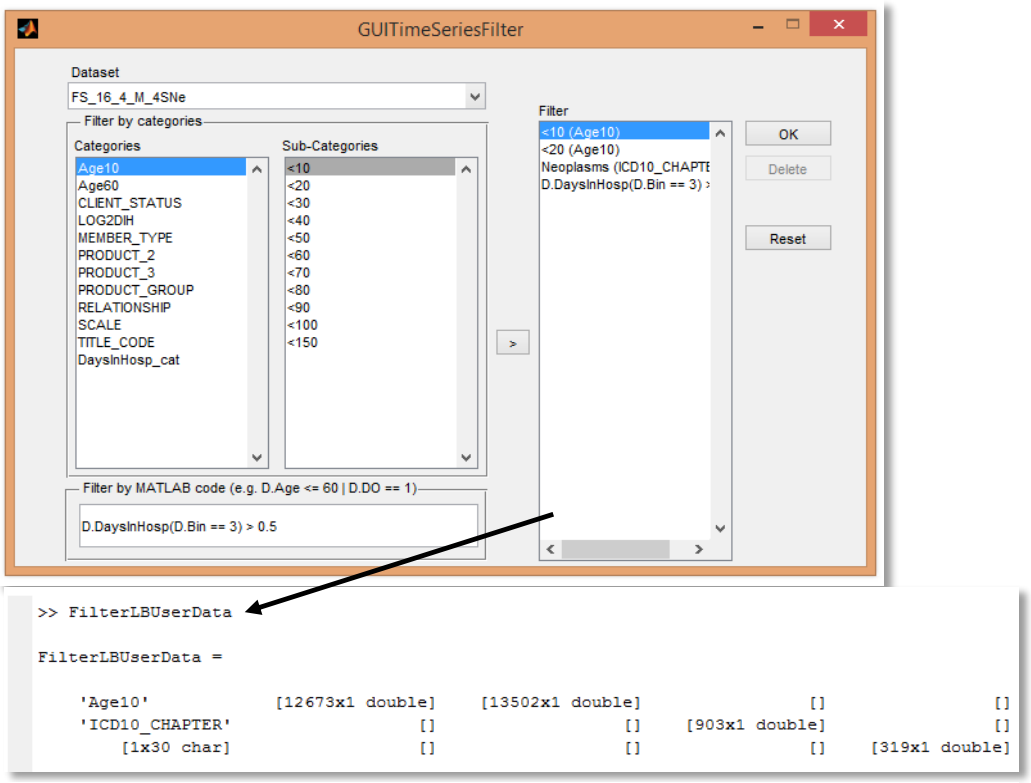

Abbildung 25: Zwischenspeichern der Ergebnisse der Teilfilterungen

Bei Betätigen des „Ok“-Pushbuttons werden die vorgefilterten IDs Zeile für Zeile zu einem einzigen Vektor zusammengeführt. Gibt es pro Zeile mehrere Spalteneinträge, so gilt für diese IDs eine „ODER“-Verknüpfung. Für das zeilenweise Zusammenführen von IDs wird eine „UND“-Verknüpfung herangezogen.

Die Namen der Zeitreihenobjekte enthalten lediglich ein „x“ vor der ID eines Patienten. Somit können sie mit dem ID-Vektor des Filter-GUIs verglichen werden und nur jene Namen in der Patientenebene angezeigt werden, die mit einem der ID-Namen des Filters übereinstimmen. Werden die Filterkriterien gelöscht, so werden wieder alle Zeitreihenobjekte in der Patientenebene zur Auswahl zur Verfügung gestellt.

2.3.5 Statistik-Tool (siehe 3.2.4)

Für die Visualisierung werden Funktionen der MATLABs Statistik-Toolbox verwendet:

1. Histogramme (Abbildung 43): `hist`
2. Boxplots (Abbildung 44): `boxplot`
3. Gruppierte Scatterplots (Abbildung 45): `gscatter`

Die Beschriftung der Achsen erfolgt automatisch anhand der Metadaten-UserData des Datasets. Die Variablenbeschreibungen für das UIContextMenu, das beim Rechtsklick auf eine Variable in einer der Listboxen angezeigt wird, werden aus den Metadaten-VarDescription des Featuresets ausgelesen.

2.3.6 Modellierungs-Tool (siehe 3.2.5)

Für die in 2.1.2 vorgestellte prä-existente Modellierungs-Routine wurde lediglich das initiale Layout gestaltet. Wie beim GUIcreateFeatureSet (2.3.3.3) orientiert auch dieses sich an den Inputparametern der Funktion `hcf_model`, welche die Modellierungs-Routine steuert.
Die Implementierung zum Layout wurde von einem Kollegen im Rahmen der Kooperation von AIT mit der University of New South Wales, Sydney, Australien, durchgeführt.

3 Ergebnisse

3.1 Feature-Extraktion

In diesem Kapitel wird gezeigt, wie aus ICD-10 und DRG Codes mit Hilfe von standardisierten und validierten Methoden Features abgeleitet werden können und dadurch gleichzeitig in eine für den Menschen interpretierbare Form gebracht werden.

3.1.1 ICD-10 Codes: Kurzbeschreibung

International Classification of Diseases and Related Health Problems, 10th Revision (ICD-10) ist ein Diagnoseklassifikationssystem, das von der Weltgesundheitsorganisation (WHO) herausgegeben und in jährlichen Abständen adaptiert wird. Es hat sich mittlerweile zum internationalen Standard zur Verschlüsselung von Diagnosen etabliert. Die ICD-10 Codes unterliegen einer hierarchischen Struktur, wonach sie Krankheiten, Verfahren und Zuständen mit unterschiedlichem Detaillierungsgrad zugeordnet sind (WHO, 2011).

In Australien werden Diagnosen mit ICD-10-AM codiert. Es handelt sich dabei um eine modifizierte Version, wobei die ersten drei Stellen des Codes mit der Basisklassifikation der WHO konform sind. Die letzten beiden Stellen der ICD-10-AM Codes liefern zusätzliche Informationen über den Ort des Auftretens und die Aktivität, bei der die Schädigung eingetreten ist (Health, 2004).

3.1.2 ICD-10 Codes: Hauptdiagnose

Kommt ein Patient ins Krankenhaus, so wird jene Diagnose, die ursächlich für seine Einweisung war (Hauptdiagnose) in Form eines ICD-10 Codes verschlüsselt dokumentiert und den Abrechnungsdaten fürs Versicherungsunternehmen beigelegt.

Ohne weitere Verarbeitung der Daten repräsentiert ein Feature (ICD10_PRINC_DIAG) mit rund 13.000 potentiellen Diagnosecodes die Variable Hauptdiagnose im Featureset (Abbildung 26, links). Um eine Reduktion der Dimensionalität zu erzielen, werden diese Codes mit Hilfe des standardisierten, hierarchischen Codiersystems der WHO gruppiert und somit in Features mit weniger Ausprägungen transformiert (Abbildung 26, Mitte). Die Hierarchie (Abbildung 26, rechts) hat zur Folge, dass Features in Form von Diagnosegruppen mit unterschiedlichem Detaillierungsgrad generiert werden, welcher von Level 1 zu Level 3 hin zunimmt.

Abbildung 26 illustriert die Methode des Ableitens von Features aus verschiedenen Levels der ICD-10 Codehierarchie.

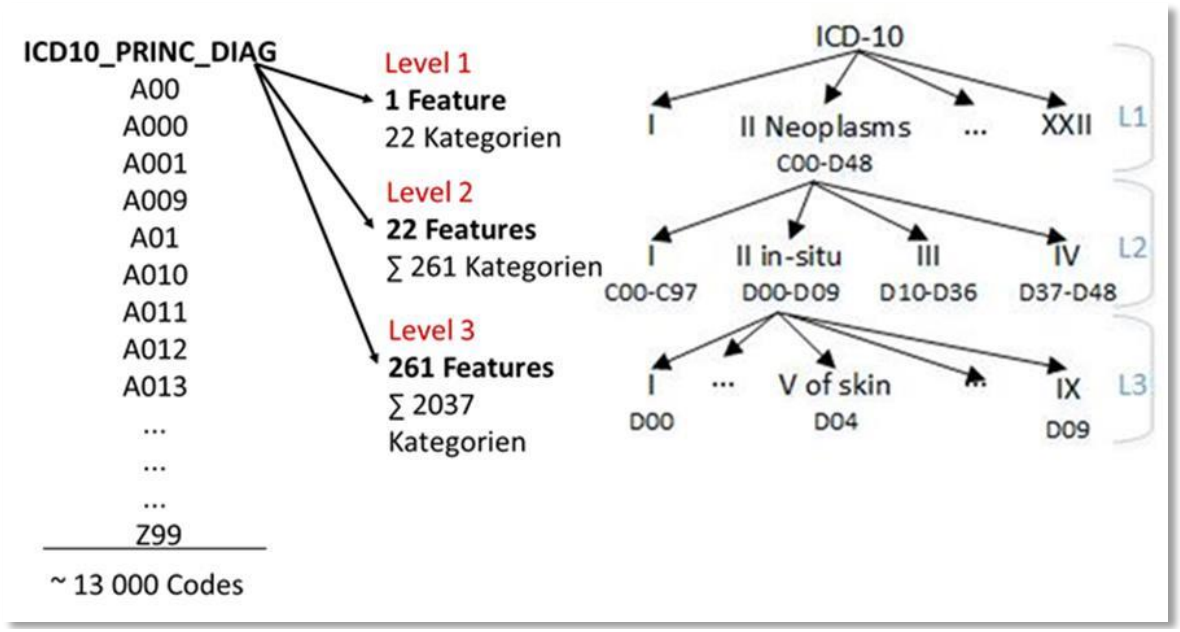

Abbildung 26: Ableiten von Features aus ICD-10 Hauptdiagnose-Codes

Beim ersten Level wird das ursprüngliche Feature in ein neues Feature mit nur 22 Kategorien überführt. Die Kategorien stellen dabei allgemeine Diagnosegruppen in Form von Krankheitskapitel dar (Tabelle 1).

Level-1-Feature: ICD10_CHAPTER

Kategorie 1	Bestimmte infektiöse und parasitäre Krankheiten
Kategorie 2	Neubildungen (beispielsweise Tumore u. Ä.)
…	…
Kategorie 22	Schlüsselnummer für besondere Zwecke

Tabelle 1: ICD-10-Level-1-Feature: Krankheitskapitel

In Ebene zwei wird jede Kategorie des ICD10_CHAPTER-Features in ein neues Feature mit neuen Subkategorien transformiert (Tabelle 2).

Level-2-Feature: Neubildungen

Kategorie 1	Bösartige Neubildungen
Kategorie 2	In-situ-Neubildungen
Kategorie 3	Gutartige Neubildungen
Kategorie 4	Neubildungen unsicheren oder unbekannten Verhaltens

Tabelle 2: ICD-10-Level-2-Feature: Neubildungen

In Level 3 werden Kategorien vom Level 2 wieder zu neuen Features mit eigenen Kategorien (Tabelle 3).

Level-3-Feature: In situ Neubildungen

Kategorie 1	Carc. in situ der Mundhöhle, des Ösophagus und des Magens
Kategorie 2	Carc. In situ sonst. u. nicht näher bezeichneter Verdauungsorgane
	…
Kategorie 9	Carc. In situ sonst. u. nicht näher bezeichneter Lokalisationen

Tabelle 3: ICD-10-Level-3-Features: In situ Neubildungen

3.1.3 ICD-10 Codes: Nebendiagnosen

Im Gegensatz zur Hauptdiagnose können für einen Patienten mehrere Nebendiagnosen vorhanden sein, welche die Komorbiditäten und Komplikationen während des Krankenhausaufenthaltes kennzeichnen. Sie sind ebenfalls nach ICD-10-AM verschlüsselt.

Der Schweregrad der Begleiterkrankungen lässt sich mit Hilfe des Charlson Komorbiditäts-Scores berechnen. Es handelt sich dabei um ein validiertes Verfahren zur Ermittlung des 10-Jahres-Mortalitätsrisikos eines Patienten (Charlson et al., 1987). Es gibt vordefinierte Erkrankungsgruppen, die mit einem erhöhten Mortalitätsrisiko assoziiert werden und denen

dementsprechend ein Score zugewiesen ist. Besitzt ein Patient beispielsweise drei Nebendiagnosen, wobei jede einer der festgelegten Krankheitsgruppen zugeordnet werden kann, so werden die zugehörigen Scores aufsummiert. Umso höher der Wert ist, umso schlechter ist der Allgemeinzustand des Patienten.

Für die Berechnung des Charlson Komorbiditäts-Scores wurde dabei eine upgedatete Version der Scores verwendet (Tabelle 4), um die Änderungen im Mortalitätsrisiko, die sich seit der Entwicklung des Charlson Komorbiditäts-Index 1984 aufgrund des technologischen und medizinischen Fortschritts ergeben haben, zu berücksichtigen.

Gewichte zur Berechnung des Charlson Komorbiditäts-Scores

Vordefinierte Erkrankungsgruppen	Charlson Weight	Updated Charlson Weight
Acute myocardial infarction	1	0
Congestive heart failure	1	2
Peripheral vascular disease	1	0
Cerebral vascular accident	1	0
Dementia	1	2
Pulmonary disease	1	1
Connective tissue disease	1	1
Peptic ulcer disease	1	0
Liver disease	1	2
Diabetes	1	0
Diabetes with complications	2	1
Hemiplegia or paraplegia	2	2
Rena disease	2	1
Cancer	2	2
Metastatic cancer	6	4
Severe liver disease	3	6
HIV disease	6	4

Tabelle 4: Ursprüngliche und upgedatete Charlson-Gewichte (Vgl. Quan et al., 2011, Sundararajan et al., 2004)

3.1.4 AR-DRG Codes

DRG Codes werden zur Abgeltung von Gesundheitsleistungen verwendet. Eine Diagnosis Related Group entspricht einer Gruppe von Patienten mit einem ähnlich hohen Ressourcenaufwand. Die Patientenklassifikation erfolgt anhand der Hauptdiagnose, den erhaltenen Leistungen und demographischen Daten.

Der Australian-Refined-DRG Code ist vierstellig. Die erste Stelle repräsentiert die Major Diagnostic Categories (Hauptdiagnosegruppen), welche die Codes nach dem Organsystem oder der Ursache der Erkrankung gliedern. Die Stellen 2 und 3 geben Auskunft darüber, ob ein chirurgischer, medizinischer oder sonstiger Eingriff stattgefunden hat. Die letzte Stelle beinhaltet Informationen über die Ressourcenintensiät, welche mit dem Komplexitätsgrad und dem Schweregrad der Begleiterkrankungen während des betroffenen Krankenhausaufenthaltes korreliert (Abbildung 27) (Fischer, 2001).

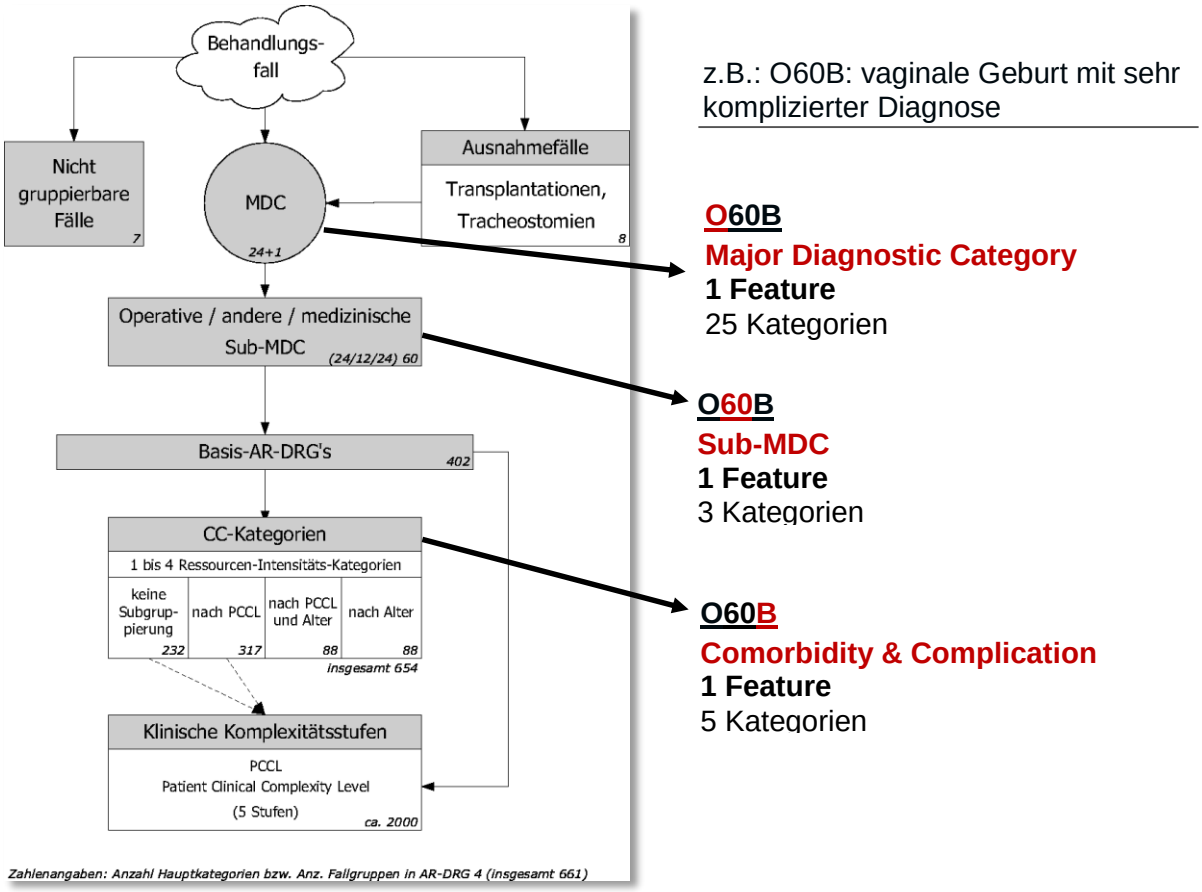

Abbildung 27: Ableiten von Features aus der AR-DRG Hierarchie (Fischer, 2001)

Tabelle 5 enthält Beispiele der Major Diagnostic Categories.

Feature: DRG_MDC

Kategorie 1	9 = Nicht klassifizierbare Fälle
Kategorie 2	A = Ausnahmefälle
Kategorie 3	B = Nervensystem
Kategorie 4	C = Augen
…	…
Kategorie 25	Faktoren, die den Gesundheitszustand beeinflussen

Tabelle 5: DRG-Feature: Major Diagnostic Categories (DRG_MDC)

Tabelle 6 veranschaulicht die Kategorisierung der mittleren zwei Stellen des Codes nach Eingriffen.

Feature: DRG_SUB_MDC

Kategorie 1	01-39 = Operative
Kategorie 2	40-59 = Andere
Kategorie 3	60-99 = Medizinische

Tabelle 6: DRG-Feature: Sub-Major-Diagnostic-Categories (DRG_SUB_MDC)

In Tabelle 7 ist die Information über die Ressourcenintensität aus der letzten Codestelle dargestellt.

Feature: DRG_CC

Kategorie 1	A = schwerste CC-Kategorie
Kategorie 2	B
Kategorie 3	C
Kategorie 4	D = leichteste CC
Kategorie 5	Z = keine CC-Unterteilung

Tabelle 7: DRG-Feature: Comorbidity or Complication (DRG_CC)

Diese Features können miteinander in Beziehung gesetzt werden, um somit binäre Features aus der semi-hierarchischen Gliederung zu erhalten (Abbildung 28).

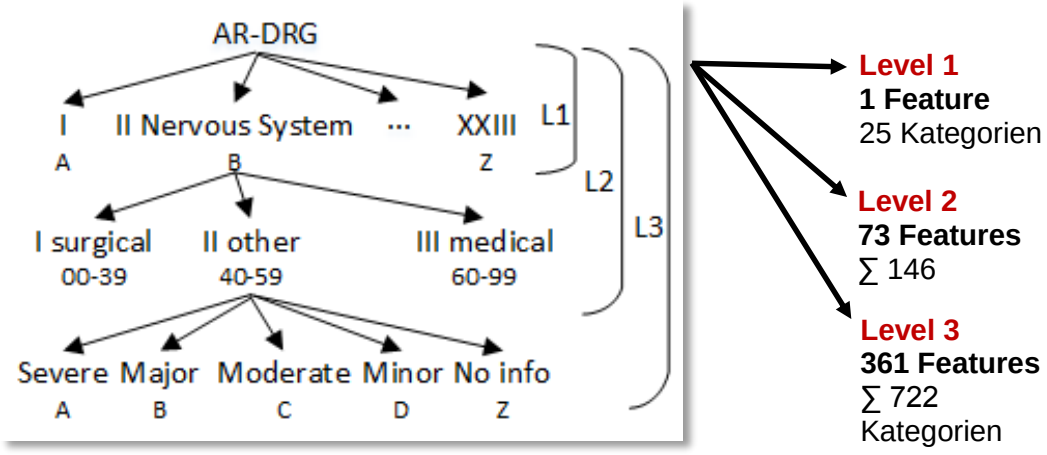

Abbildung 28: Ableiten von Features aus dem semi-hierarchischen DRG-Schema

3.2 Visual Analytics

3.2.1 Framework

Das in Abbildung 29 illustrierte Framework bildet das Rahmenwerk für die einzelnen Tools, welche für sich eigenständige Module darstellen. Das Framework ist dazu angedacht, diese bei Bedarf miteinander kommunizieren zu lassen. Es regelt den Datenfluss zwischen den Tools und gewährleistet ein komfortables und einfaches Öffnen derselben durch Klicken eines Tabs.

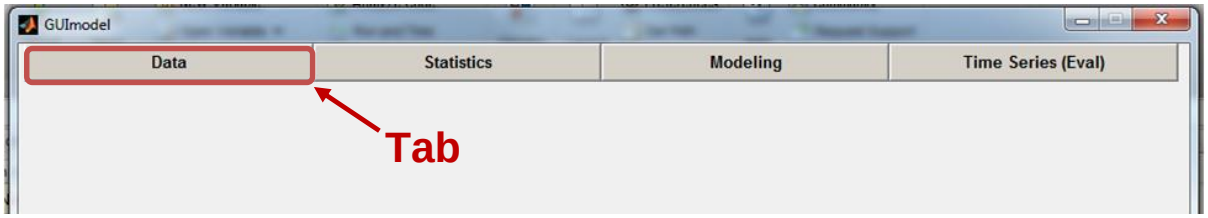

Abbildung 29: Ausschnitt aus dem Visual Analytics Framework

3.2.2 Data-Tool

Das in Abbildung 30 dargestellte Data-Tool ist für die Verwaltung der Daten innerhalb des Rahmenwerkes zuständig und ist als funktioneller Teil des Frameworks unabdingbar. Durch Öffnen dieses Tools mit Hilfe des Befehls *GUIhome* wird zugleich das gesamte Framework geöffnet. Ist ein anderes Tool in Verwendung, so kann durch Klicken des „Data“-Tabs wieder auf die GUIhome-Oberfläche gewechselt werden.

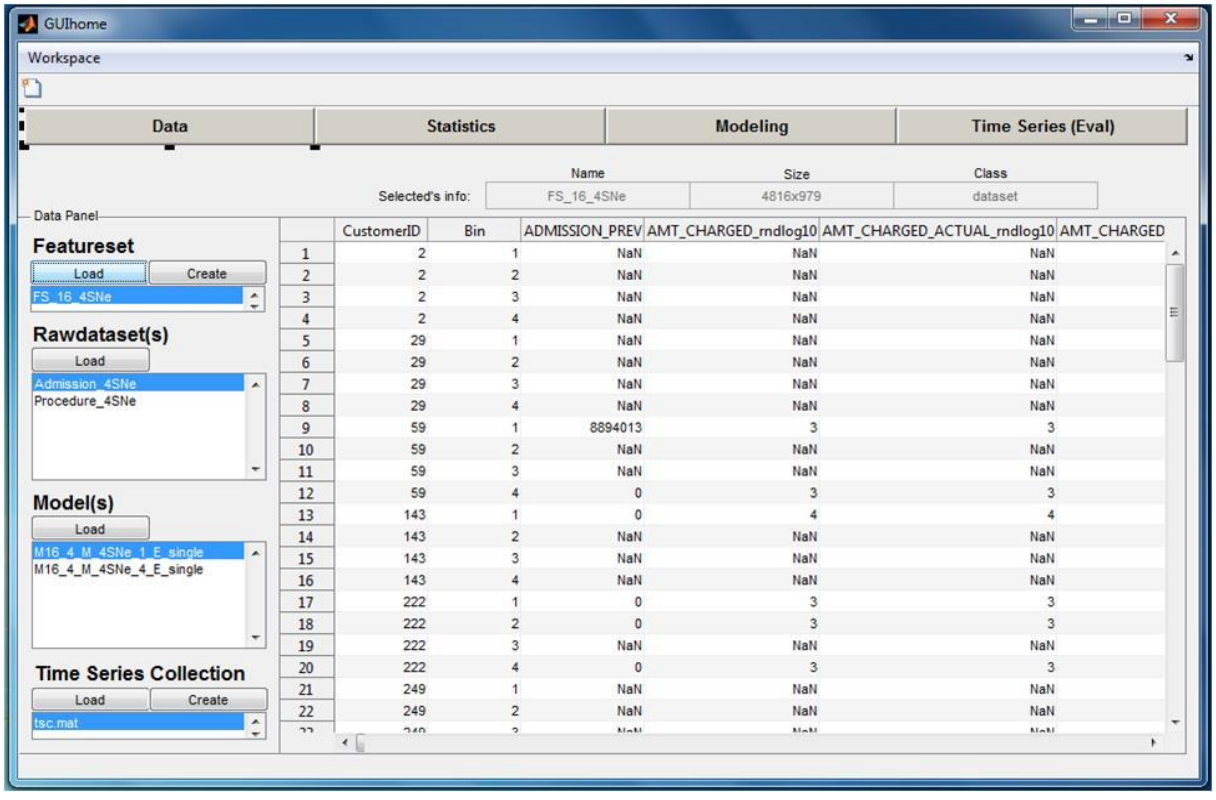

Abbildung 30: Data-Tool (GUIhome)

Das Data-Tool besteht aus einem Data-Panel zum Laden oder Erzeugen von Dateien und aus einer Tabelle, in der die Daten des Featuresets oder eines Rohdatensatzes visualisiert werden können. Die Datenverwaltung wird durch eine automatische Lade-Kaskade vereinfacht. Wird ein Featureset geladen, so werden die dazugehörigen Files automatisch mitgeladen, sofern sie in der standardisierten Ordnerstruktur (Abbildung 13) abgelegt sind. Für den Fall, dass Files geladen wurden, die nicht für das aktuelle Experiment bestimmt sind, können diese in der Listbox markiert und mit Hilfe der „Entf"-Taste wieder gelöscht werden. Wird der standardisierten Ordnerstruktur nicht Folge geleistet, so bleiben die Listboxen leer und die Dateien können nachträglich manuell geladen werden.

Wurde für ein Featureset zuvor genau einmal eine Zeitreihenkollektion erzeugt, so wird diese geladen. Befinden sich mehrere im Ordner des Featuresets, so wird am Ende des automatischen Ladevorganges ein Dialogfenster (Abbildung 31, links) geöffnet, um eine der verfügbaren Dateien auszuwählen. Konnte keine Zeitreihenkollektion gefunden werden, so wird eine Dialogbox geöffnet, die dem User die Möglichkeit bietet, selbst nach einer zugehörigen Datei zu suchen oder neue Zeitreihenobjekte zu generieren (Abbildung 31, rechts).

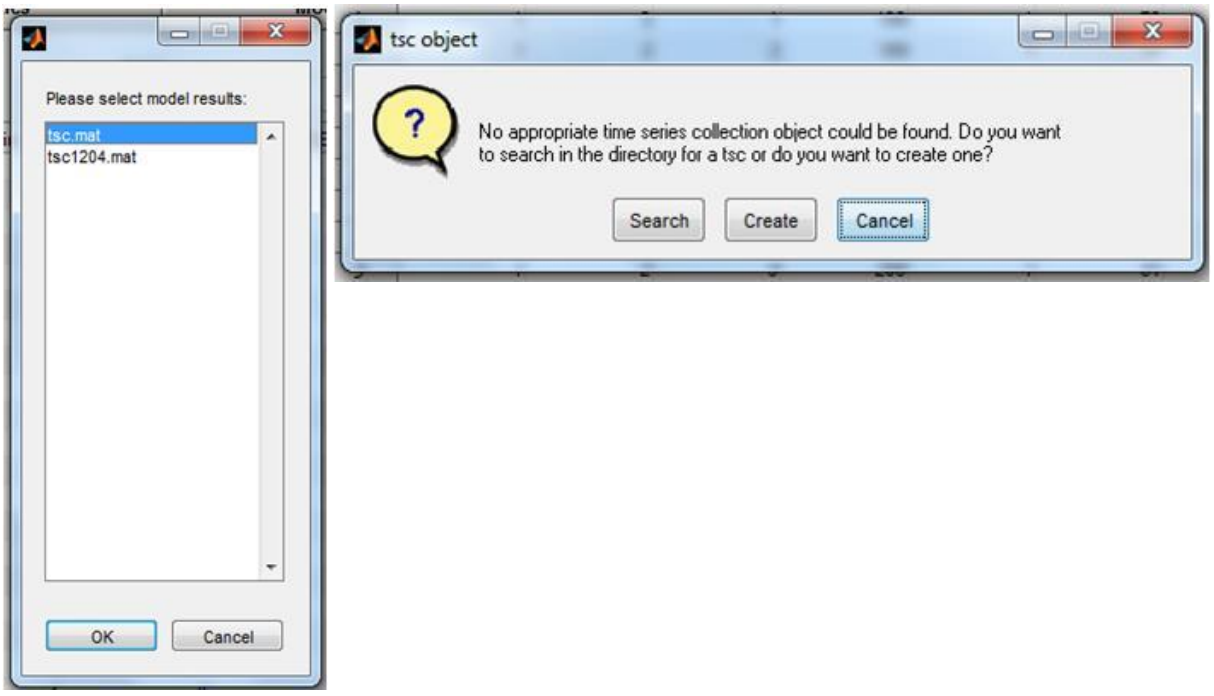

Abbildung 31: Laden der Zeitreihenkollektion

Abbildung 32 zeigt, welche Datensätze bzw. Dateien in welchen Tools Verwendung finden. Die strichlierte Linie bedeutet, dass das Zeitreihen-Tool auch ohne das Visualisieren von Modellen genutzt werden kann.

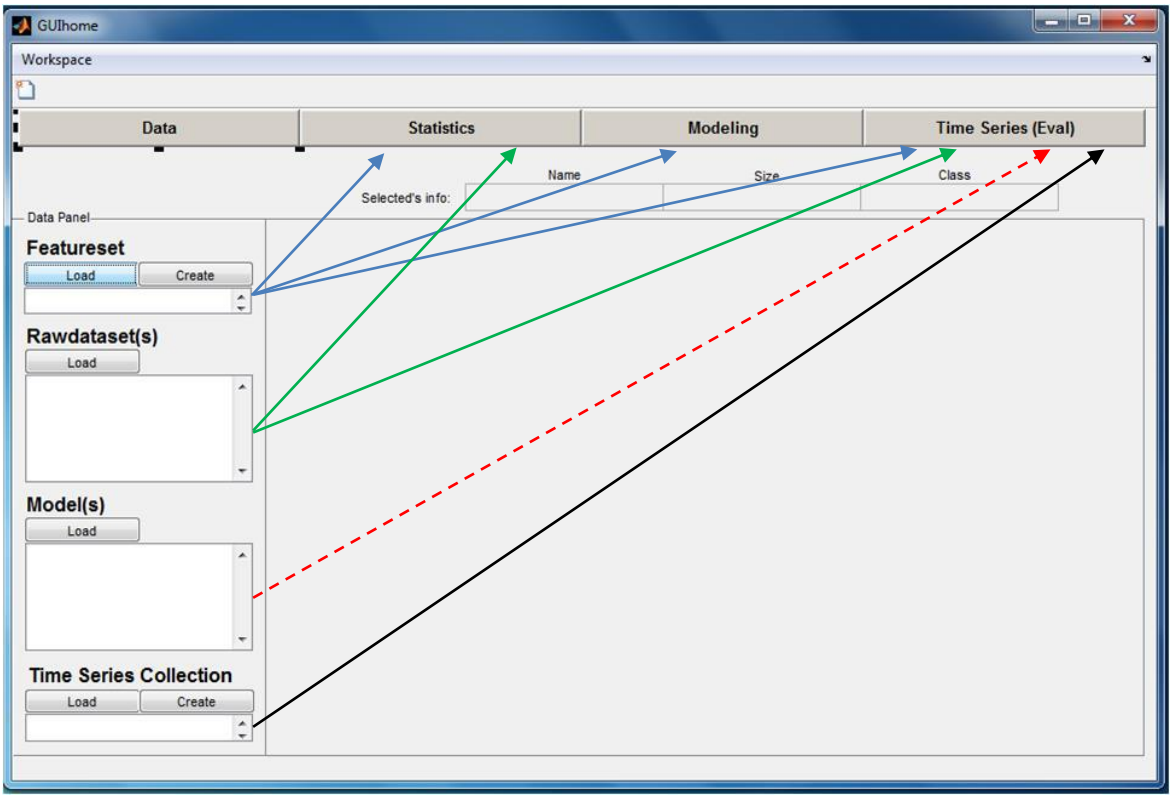

Abbildung 32: Datenmanagement des Frameworks

Um das Statistik-Tool verwenden zu können, muss zumindest das Featureset oder ein Rohdatensatz geladen werden. Soll ein Modell generiert werden, so ist es ausreichend, wenn das Featureset vorhanden ist. Für das Zeitreihen-Tool sind das Featureset, mindestens ein Rohdatensatz und die Zeitreihenkollektion erforderlich. Wenn ein bereits erstelltes Modell auf Patientenebene evaluiert werden soll, muss auch die Datei mit dessen Modellergebnissen geladen werden.

3.2.2.1 Erzeugung eines Featuresets (GUIcreateFeatureset)

Bei Betätigung des „Create“-Buttons im GUIhome wird ein weiteres GUI (Abbildung 33) geöffnet. Um ein Featureset erzeugen zu können, muss zuvor in der Feature-Definition-Table ein neuer Eintrag mit dem Namen und den Parametern des neuen Featuresets angelegt werden. Nach dem Laden dieses Excel-Files werden die Namen der darin definierten Featuresets in der Listbox zur Auswahl zur Verfügung gestellt. Das zu generierenden Featureset ist aus dieser Liste auszuwählen und der Pfad zu dem Ordner, in dem der neue Datensatz gespeichert werden soll, muss angegeben werden. Nach Betätigen des „Create Featureset“-Buttons wird das Featureset erzeugt, die standardisierte Ordnerstruktur angelegt und die

zugehörigen Rohdatensätze darin abgelegt. Sofern dieser Prozess erfolgreich war, wird das neue Featureset in der GUIhome-Oberfläche angezeigt und die Rohdatensätze werden automatisch geladen.

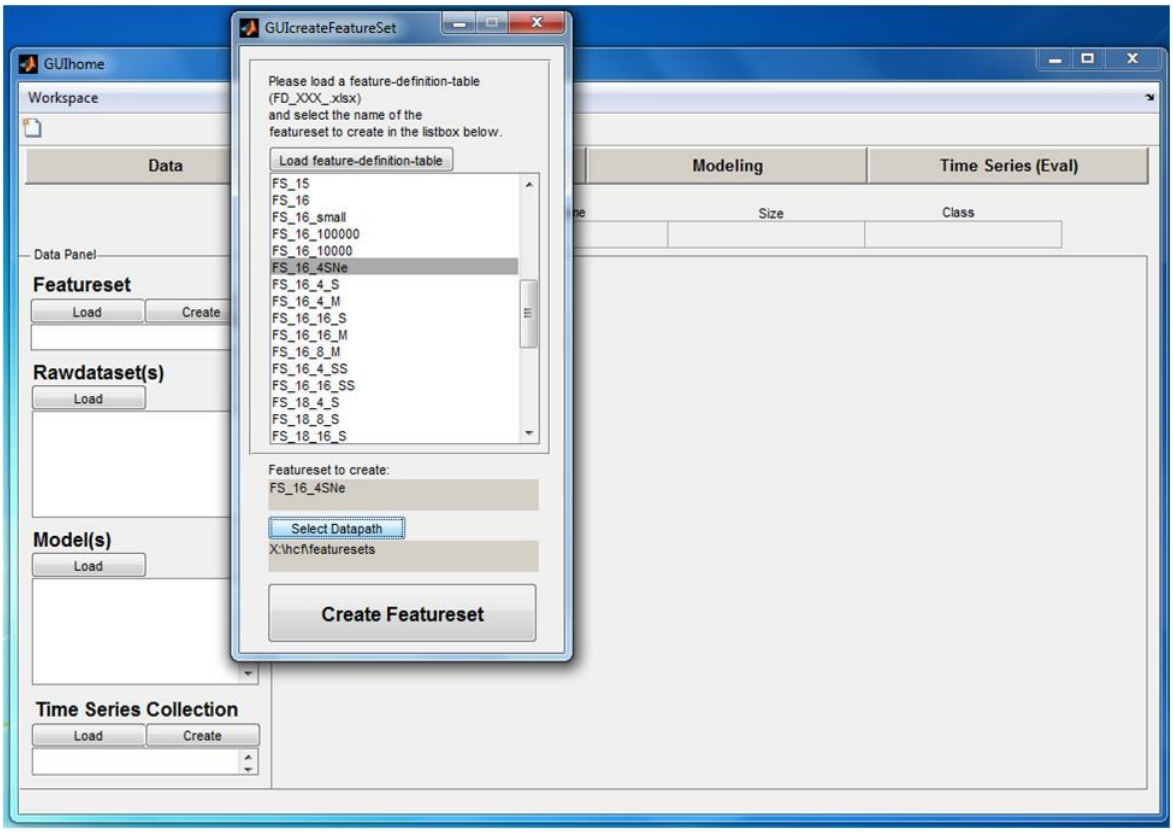

Abbildung 33: GUIcreateFeatureSet

3.2.2.2 Erzeugen einer Zeitreihenkollektion (GUIcreateTSCobj)

Um das Zeitreihen-Tool verwenden zu können, muss zumindest einmal für jedes Featureset eine Zeitreihenkollektion kreiert werden. Zur Erzeugung der Zeitreihenkollektion kann ein eigenständiges GUI (Abbildung 34) herangezogen werden. Dieses wird bei Betätigen des „Create“-Buttons vom GUIhome aus geöffnet.

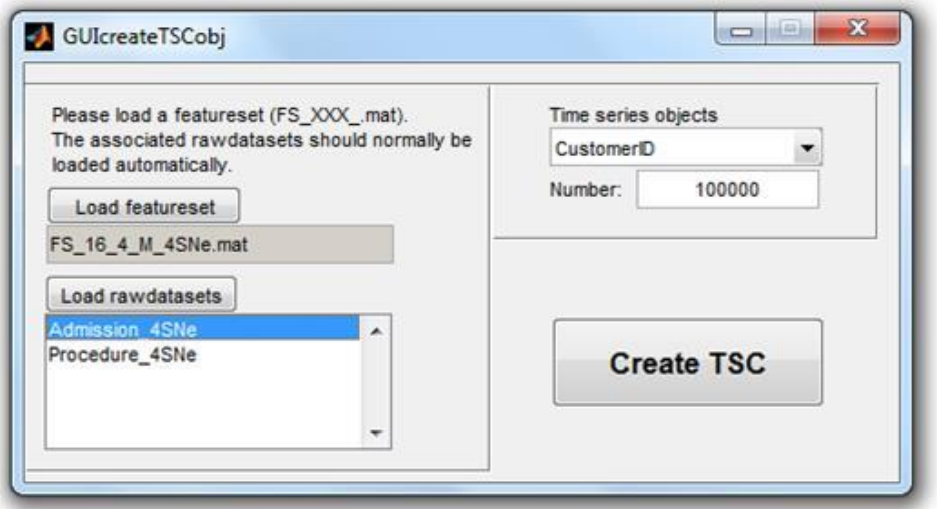

Abbildung 34: *GUIcreateTSCobj* zur Erstellung von Zeitreihenobjekten

Im Popupmenu rechts oben kann das „Objekt der Begierde“ (z.B. Patient oder Klient) ausgewählt werden. Im Edit-Feld darunter ist es möglich, die Anzahl der zu erzeugenden Zeitreihenobjekte zu verringern. Ohne gesonderte Eingabe wird die maximal verfügbare Anzahl an Objekten generiert.

Wurden nach Betätigen des „Create TSC“-Buttons alle Zeitreihenobjekte erzeugt, so öffnet sich ein Dialogfenster, damit die Zeitreihenkollektion (tsc) gespeichert werden kann (Abbildung 35). Der dabei vergebene Name sollte „TSC“ oder „tsc“ beinhalten, damit das File von der automatischen Lade-Kaskade gefunden werden kann. War der Prozess erfolgreich, so wird der Name in der GUIhome-Oberfläche angezeigt.

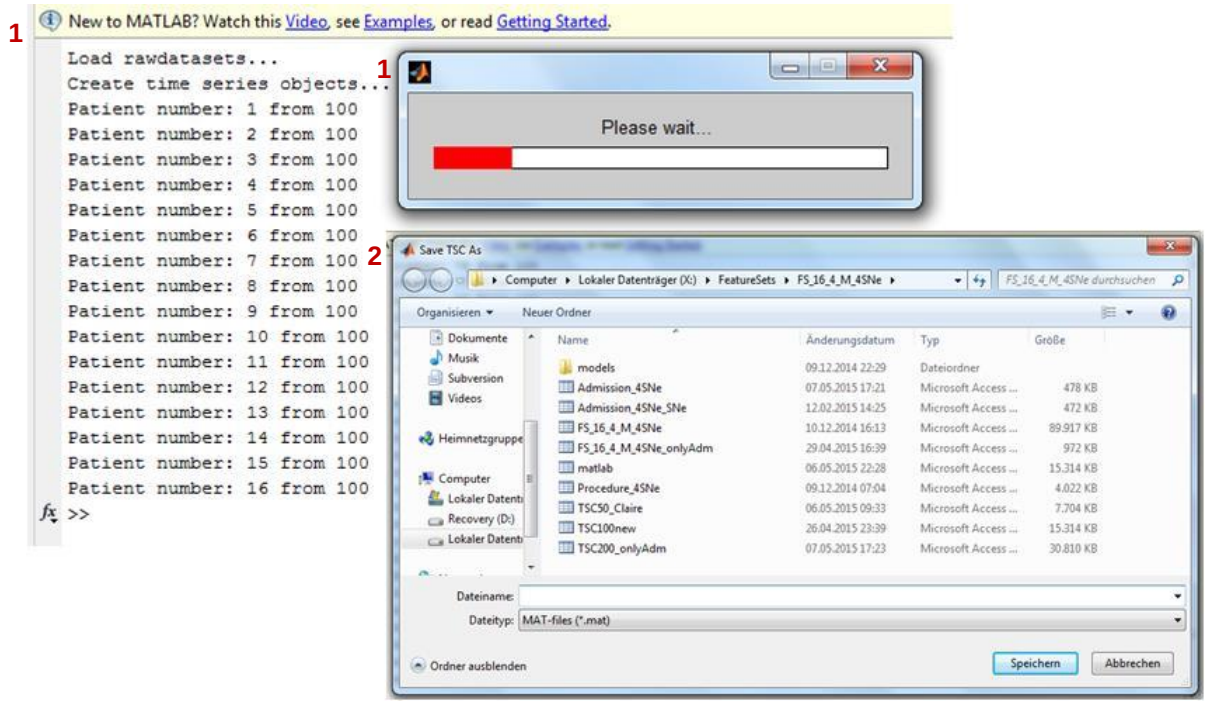

Abbildung 35: *GUIcreateTSCobj*: Erzeugung der Zeitreihenobjekte (1) und Speichern (2)

3.2.3 Zeitreihen-(Eval)-Tool

3.2.3.1 Zeitreihen-Tool: Kurzbeschreibung

Das Zeitreihen-Tool, dargestellt in Abbildung 36, gewährt durch eine gleichzeitige Visualisierung der aus verschiedenen Quellen stammenden Daten eines Patienten einen Gesamtüberblick, der ohne dieses Tool nur mit unzumutbar hohem Zeitaufwand zu realisieren wäre. Der Anwender hat die Möglichkeit, sich neben allgemeinen Informationen wie Alter, Geschlecht, oder das abgeschlossene Versicherungspacket, auch Daten mit Zeitreihencharakter auf beliebig vielen Zeitachsen gleichzeitig darstellen zu lassen. Dadurch können beispielsweise für einen Patienten neben der Hospitalisierungsgeschichte auch das Medikationsverhalten, Gewichtsschwankungen und der wetterbedingte Temperaturverlauf angezeigt

werden, wodurch eine umfassende Analyse der Einflussfaktoren und deren Zusammenhänge gestattet wird. Parallel zu den Zeitreihen besteht die Möglichkeit das Prädiktionsergebnis für den ausgewählten Patienten zu visualisieren, womit Aufschluss über Stärken und Schwächen des Modells auf Patientenebene gewonnen werden kann. Es können verschiedenen Modellen angezeigt werden, um einen Vergleich der Güte unterschiedlicher Modelle durchführen zu können. Ein Filter unterstützt den Analyseprozess, indem eine Betrachtung von Subgruppen (z.B. nur Männer über 60 Jahre, die an Herzinsuffizienz leiden) zugelassen wird.

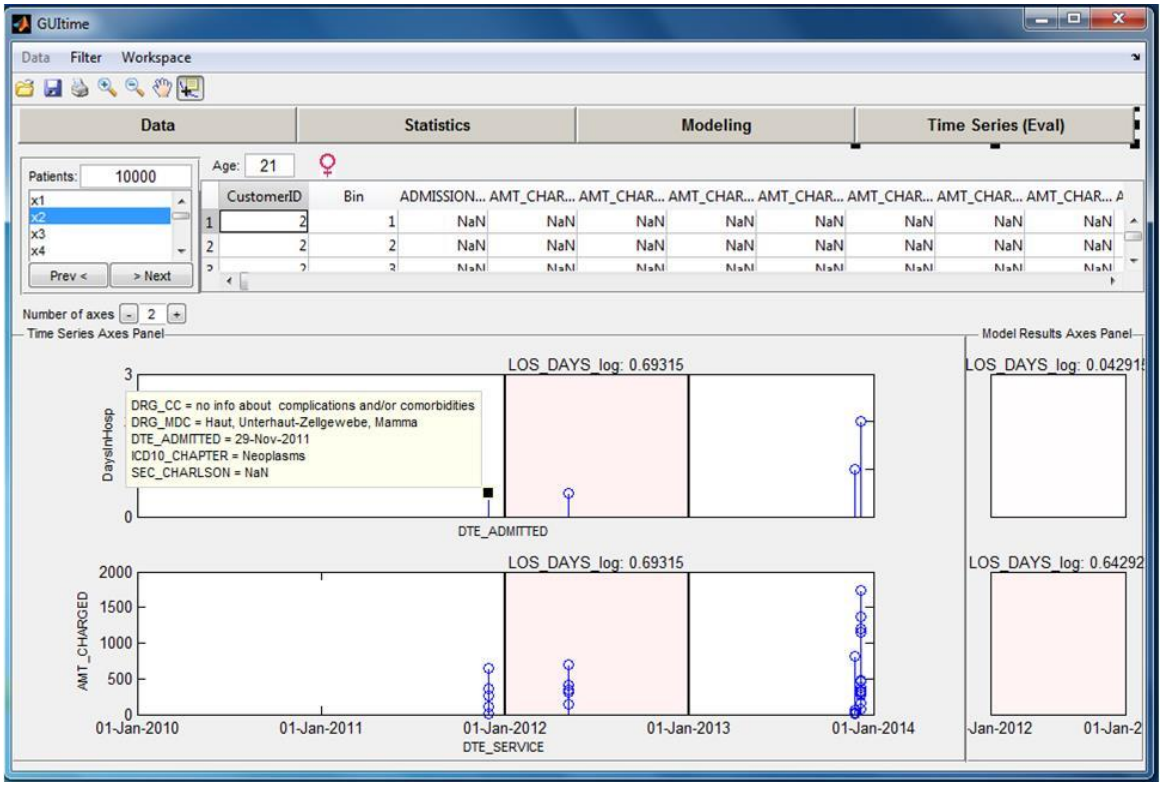

Abbildung 36: Zeitreihen-Tool

3.2.3.2 Zeitreihen-Tool-Kernelemente

In Abbildung 38 sind die vier entscheidenden Elemente gekennzeichnet, aus denen sich das Zeitreihen-Tool zusammensetzt.

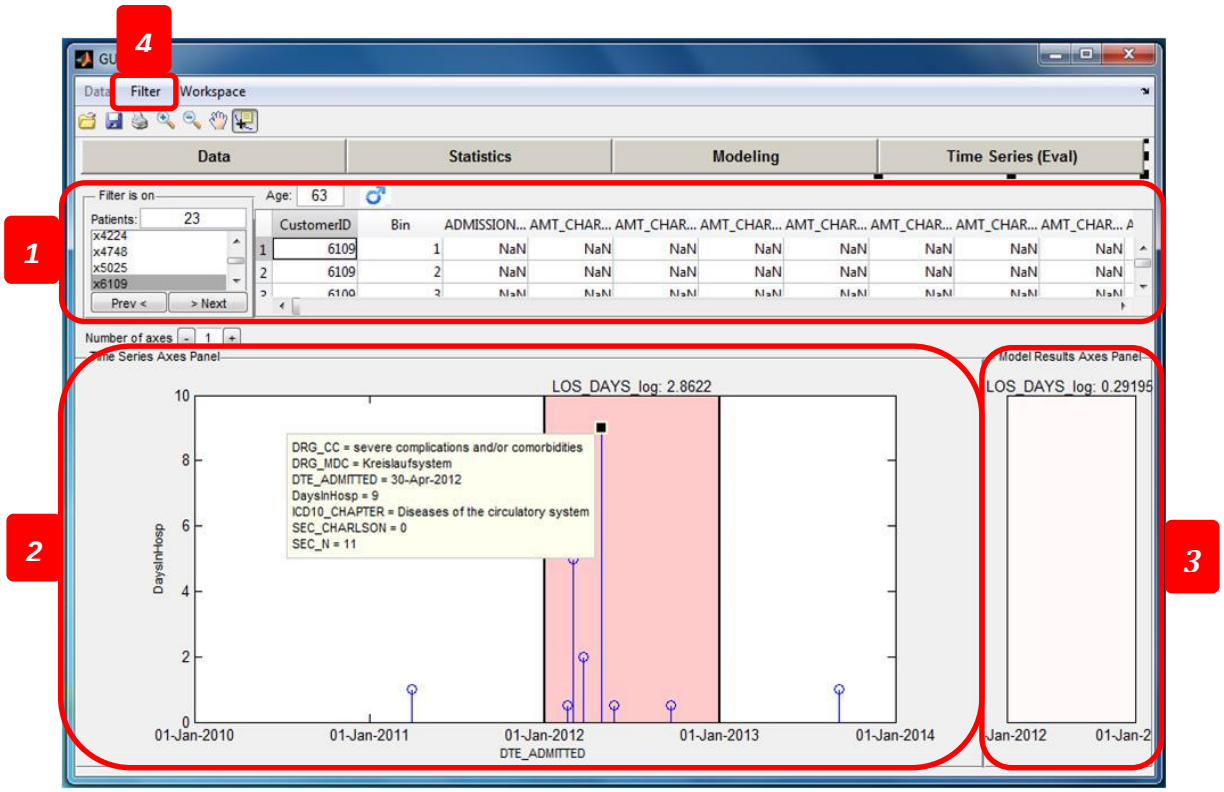

Abbildung 38: Zeitreihen-Tool-Kernelemente

Patientenebene

Das erste Element repräsentiert die Patientenebene. Sie besteht aus einer Listbox, in der die Patienten zur Auswahl gelistet sind, und aus einer Tabelle, welche allgemeine Informationen aus dem Featureset zum aktuell gewählten Patienten beinhaltet. Alter und Geschlecht werden hier gesondert hervorgehoben. Ein Klicken auf einen Patienten in der Listbox führt dazu, dass dessen Daten visualisiert werden.

Zeitreihe

Die Zeitreihenachse stellt das zweite Kernelement dar. In ihr können Daten mit Zeitreihencharakter visualisiert werden, die aus einem der Rohdatensätze stammen.
Die vertikalen Linien trennen die Testperiode von der davor liegenden Trainingsperiode und dem dahinter liegenden nicht betrachteten Zeitraum. Der Hintergrund des Testzeitraumes ist farbcodiert. Umso höher der Wert der zu prädiktierenden Variable (hier die Anzahl der Tage im Krankenhaus im Jahr 2013) ist, umso roter färbt sich der Hintergrund. Sofern für den ausgewählten Patienten Ereignisse vorhanden sind, werden diese als „stem“-Plot angezeigt.

Durch Doppelklick auf eine Zeitachse öffnet sich ein GUI (*GUITimeSeriesAxis*) (Abbildung 39) in dem festgelegt werden kann, was in der Zeitachse dargestellt werden soll. Im Plot-

Panel können die Variablen für die x- und die y-Achse ausgewählt werden und im Data-Cursor-Panel kann definiert werden, was in einem Datatip angezeigt werden soll, wenn auf ein Ereignis in der Zeitreihe geklickt wird.

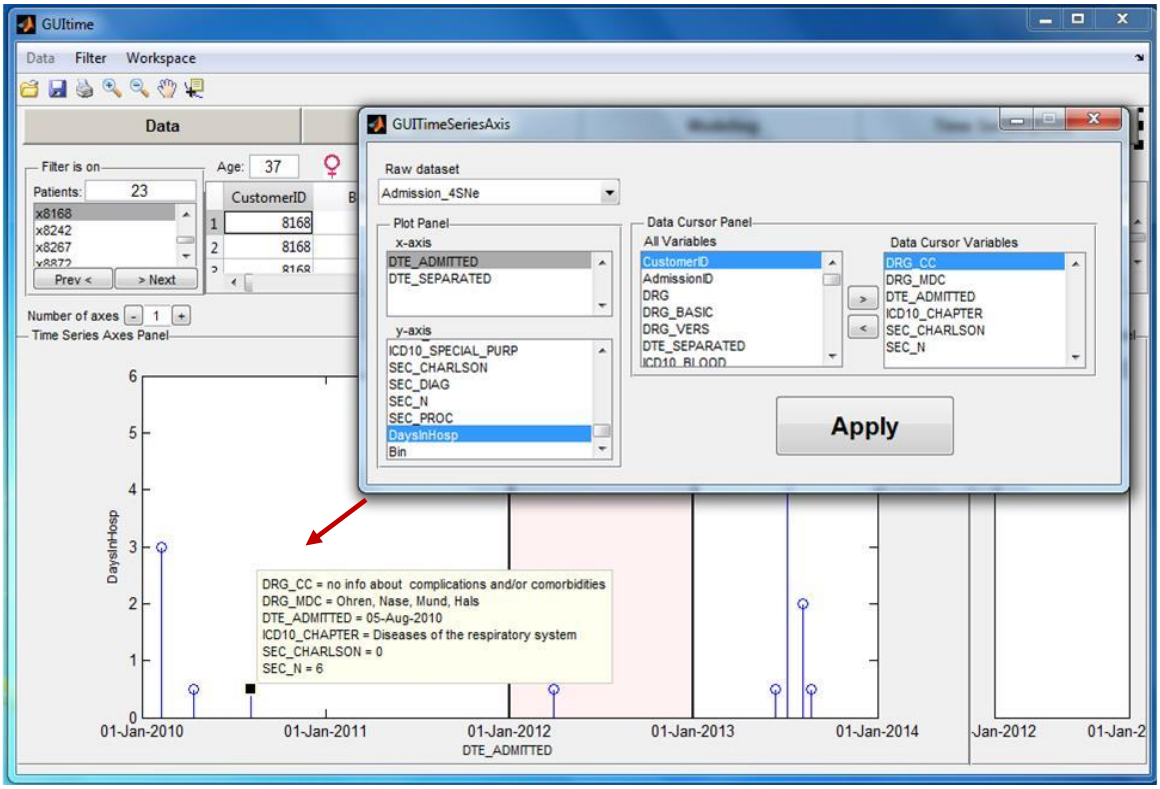

Abbildung 39: Zeitreihen-Tool: Doppelklick auf Zeitreihenachse (GUITimeSeriesAxis)

Modellergebnis

Die Achse für die Visualisierung des Prädiktionsergebnisses stellt das dritte Kernelement dar. Durch Doppelklick auf die Modellachse wird ein Dialogfeld mit einer Liste der verfügbaren Modelle geöffnet (Abbildung 40). Der Hintergrund der Modellachse ist in der gleichen Art und Weise farbcodiert wie jener der Testperiode in der Zeitreihenachse, wodurch ein schnelles Erkennen der Modellgüte ermöglicht wird. Stimmen die Hintergrundfarben überein, so wurde vom Modell eine korrekte Prädiktion für den Testzeitraum durchgeführt.

Die Zeitreihenachse und die Modellachse stellen zusammen eine Ebene dar. Die Anzahl der Ebenen kann mit Hilfe des „Plus“- und „Minus“-Pushbuttons variiert werden.

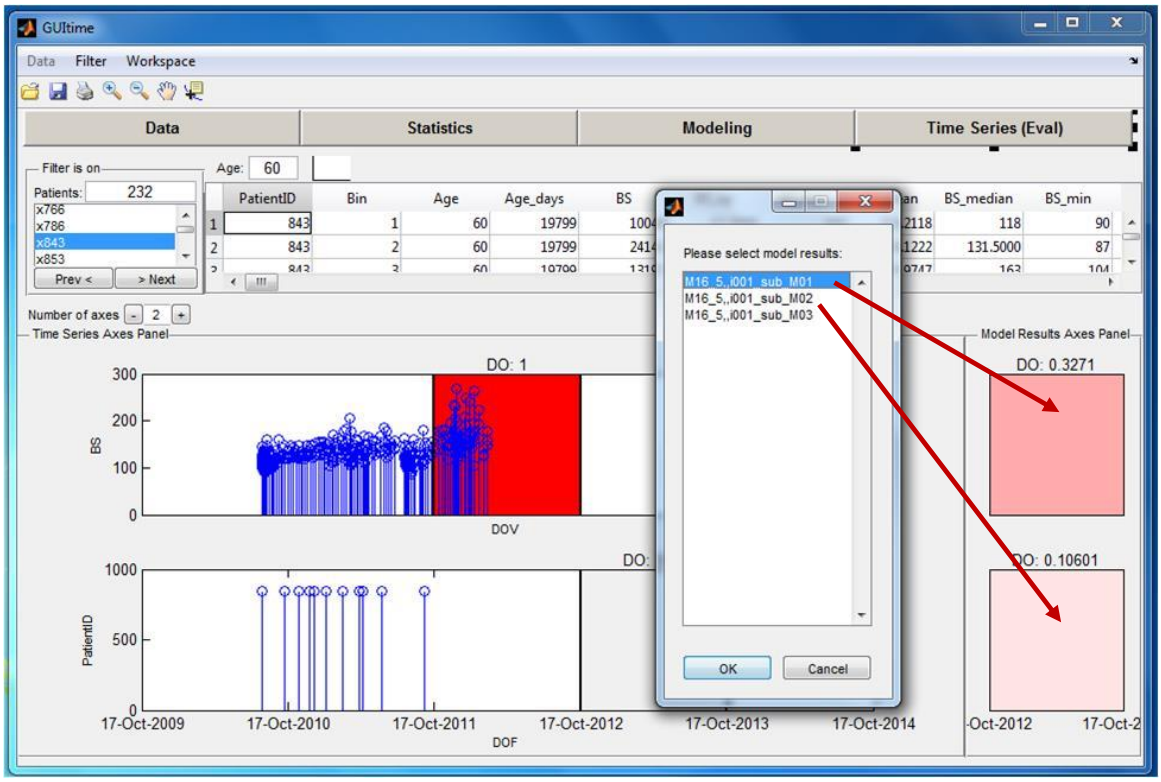

Abbildung 40: Zeitreihen-Tool: Doppelklick auf Modellachse

Filter

Das vierte Kernelement ist der Filter, der in Abbildung 41 dargestellt ist. Er bietet die Möglichkeit, die Analysen auf Subpopulationen zu beschränken. Hierfür kann aus allen Kategorien der verfügbaren Datasets gewählt werden. Die „Admission Table“ enthält beispielsweise die ICD-10 Kategorien der ersten drei Ebenen des Hierarchiesystems. Somit können einzelne Krankheitsgruppen, wie Herz-Kreislauf-Erkrankungen, Tumorerkrankungen oder Schwangerschaften, mit unterschiedlichem Detaillierungsgrad in der Diagnose betrachtet werden. Mit Hilfe der „Procedure Table“ können unter anderem Einschränkungen auf bestimmte Leistungen erfolgen, wie beispielsweise Strahlentherapie. Das Featureset enthält zusätzliche Informationen, die keinen Zeitreihencharakter haben, wie Alter oder Geschlecht. Zudem ist es möglich auch nach numerischen Variablen zu filtern. Hierfür muss der entsprechende MATLAB-Code ins Edit-Feld eingegeben werden.
Durch Klicken des „Ok“-Buttons erfolgt die Filterung nach allen Kriterien, die in der „Filter“-Listbox eingetragen sind. Ist der Filter aktiv, so wird in der Patientenebene die Anzahl der auswählbaren Patienten angepasst und „Filter is off“ mit „Filter is on“ ersetzt.

Abbildung 41 illustriert das Filtern des Patientenkollektivs mit Hilfe des Filter-GUIs. Es ist ersichtlich, dass sich unter den 10 000 Patienten nur 2 Kinder mit einem Alter unter 10 Jahren befinden, bei der eine Tumorerkrankung diagnostiziert wurde.

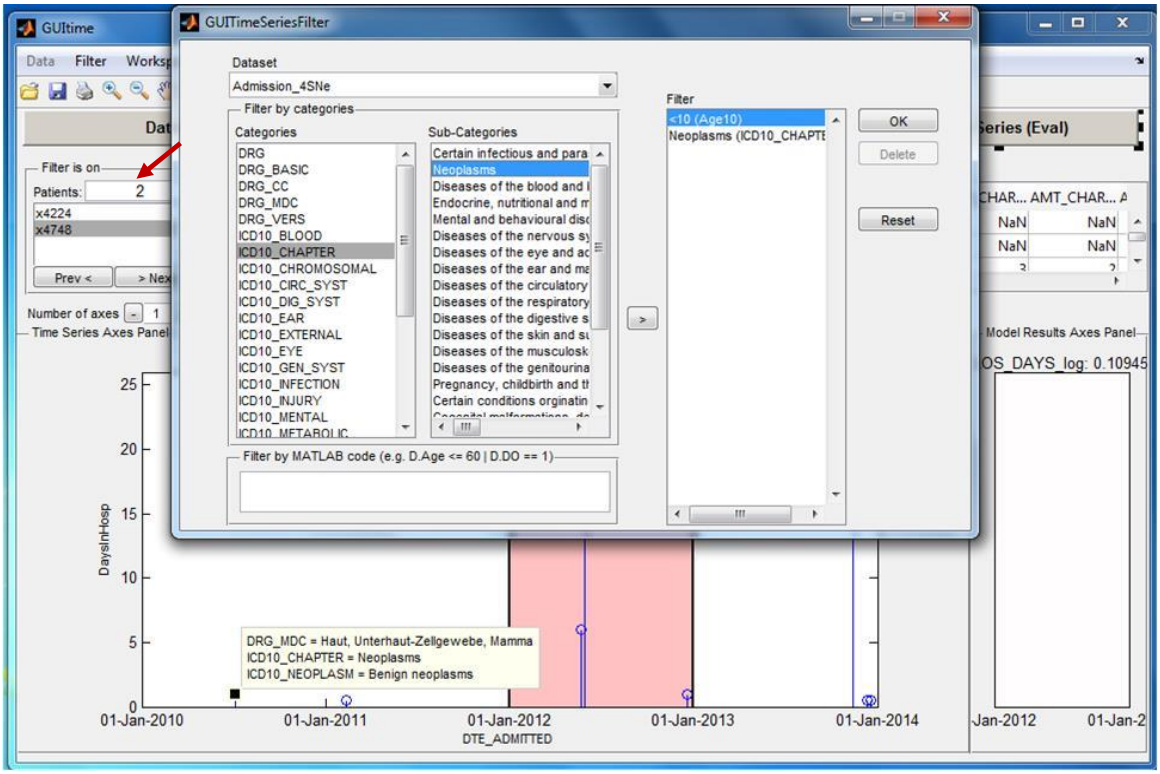

Abbildung 41: Zeitreihen-Tool: Filter-GUI (GUITimeSeriesFilter)

3.2.4 Statistik-Tool

Das Statistik-Tool dient primär der Vollständigkeit des Frameworks und stellt für sich allein noch kein Visual Analytics Tool dar. Infolgedessen wird es in diesem Abschnitt nur kurz vorgestellt.

3.2.4.1 Statistik-Tool: Kurzbeschreibung

Das Statistik-Tool in Abbildung 42 ermöglicht eine schnelle und einfache Analyse von Variablen eines MATLAB-Datasets mit grundlegenden Methoden der Statistik. Ursprünglich wurde es zur Anwendung auf die Variablen eines Featuresets konzipiert. Es können aber auch Rohdatensätze damit analysiert werden.

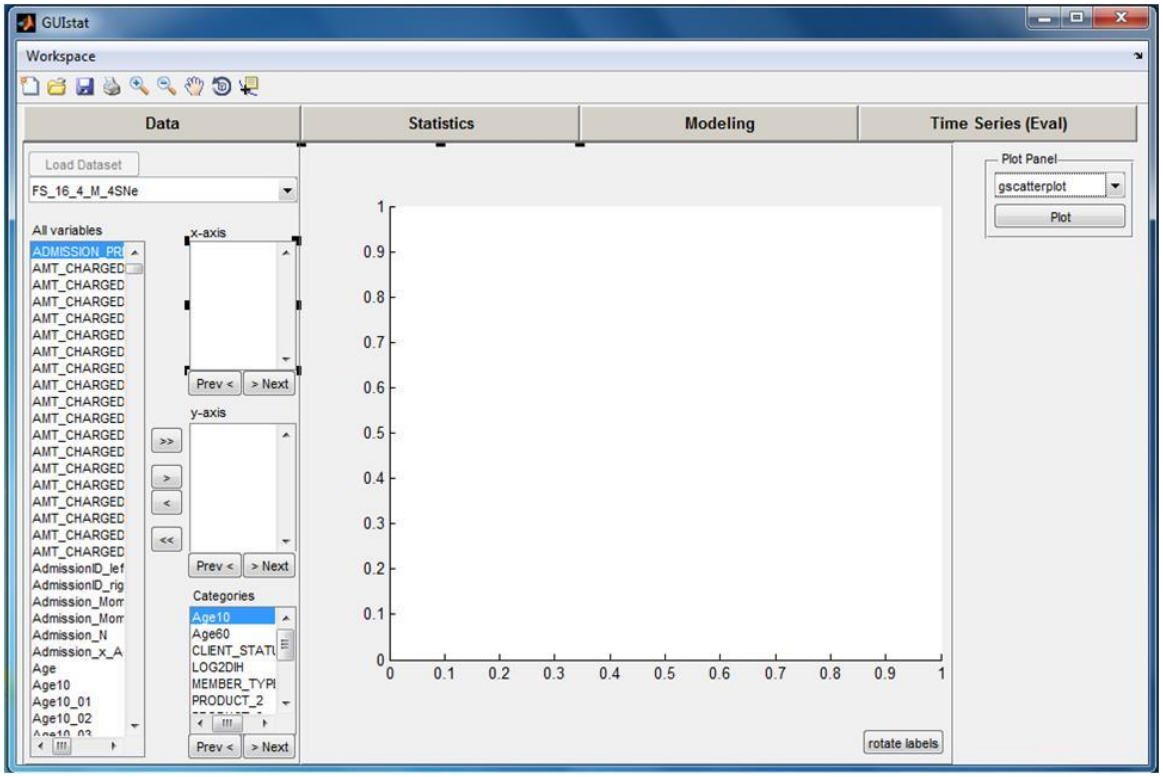

Abbildung 42: Statistik-Tool (GUIstat)

Im Popupmenu auf der linken Seite kann zwischen dem Featureset und eines der Rohdatensätze gewählt werden. Dessen Variablen werden in der „All variables“-Listbox angezeigt. Die „Categories“-Listbox enthält ausschließlich kategorische Variablen.

Im Popupmenu auf der rechten Seite ist eine Auswahl zwischen drei Darstellungen möglich:

1. Histogramme (Abbildung 43)
2. Boxplots (Abbildung 44)
3. Gruppierte Scatterplots (Abbildung 45)

Wird eine Darstellungsart gewählt, so sind immer nur jene Listboxen sichtbar, die für diesen Plot-Typ relevant sind. Somit ist für Histogramme nur die „x-axis“-Listbox sichtbar, für Boxplots wird zusätzlich die „y-axis“-Listbox und für die Gruppierte-Scatterplot-Darstellung auch die „Categories“-Listbox angezeigt.

Mit den „Pfeil“-Pushbuttons können die Variablen zwischen der „All variables“-Listbox und der „x-axis“- oder der „y-axis“-Listbox übertragen werden. Variablen in der „x-axis“-Listbox werden auf der X-Achse und jene in der „y-axis“-Listbox werden auf der Y-Achse abgebildet. Die kategorischen Variablen in der „Categories“-Listbox dienen der Gruppierung in der Scatterplot-Darstellung und können der Z-Achse gleichgesetzt werden.
Mit Hilfe der „Prev <“- und „Next >“-Pushbuttons unter einer Listbox kann eine Variable nach der anderen schnell und einfach ausgewählt und visualisiert werden.

Die Achsenbeschriftung erfolgt automatisch. Sofern eine Kurzbeschreibung zur aktuell ausgewählten Variable vorhanden ist, kann diese durch Rechtsklick auf die Variable im Context-Menu angezeigt werden.

Abbildung 43 illustriert die Verwendung der Histogramm-Darstellung. Sie zeigt die Altersverteilung des Patientenkollektivs im Featureset. Auffallend ist, dass auch Patienten mit einem Alter von weniger als 0 und mehr als 110 Jahren in diesem Datensatz enthalten sind. Das ist nicht untypisch für „Big Data“ – Projekte, wo grundsätzlich auch mit fehlenden und fehlerhaften Datensätzen gerechnet werden muss.

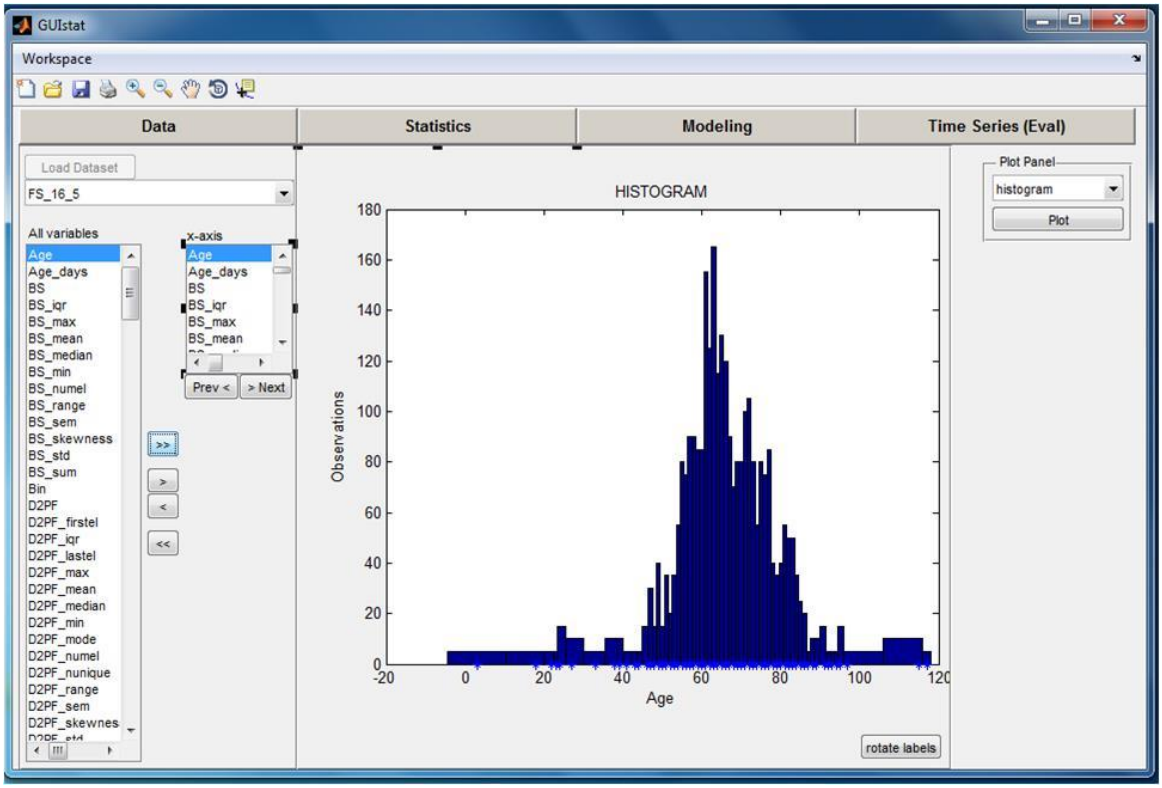

Abbildung 43: Statistik-Tool: Histogramm

Abbildung 44 zeigt die Boxplot-Darstellung. In diesem Beispiel wird der Zusammenhang zwischen Dropouts und der Anzahl an Feedbacks im Gesundheitsdialog-Diabetes-Projekt dargestellt. Es ist ersichtlich, dass in den Zeiträumen, in denen ein Dropout stattgefunden hat, im Median die Anzahl an gegebenen Feedbacks signifikant geringer war, als in jenen, in denen die Patienten nicht aus dem Telemonitoring-Programm ausgestiegen sind.

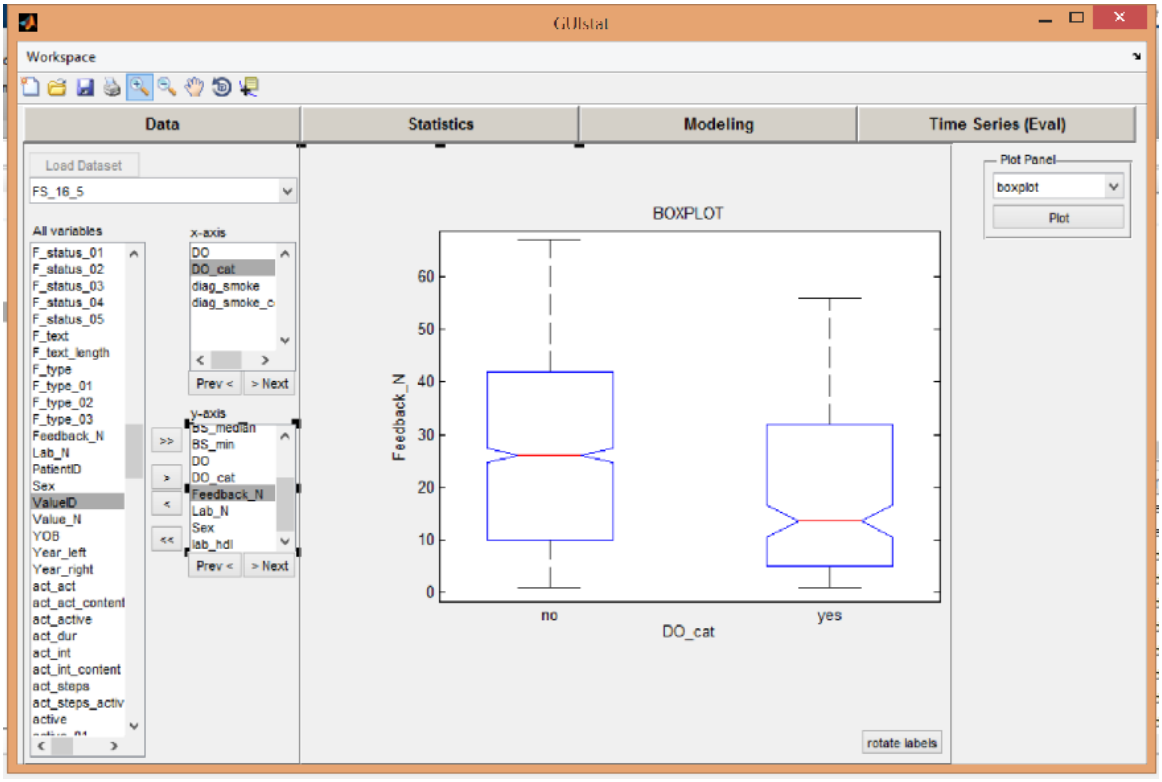

Abbildung 44: Statistik-Tool: Boxplots

Abbildung 45 veranschaulicht die Gruppierte-Scatterplot-Darstellung am Beispiel der österreichischen Transfusionspraxis für Männer und Frauen. Sie stellt das transfundierte Erythrozytenvolumen (tEV) in Abhängigkeit des verlorenen Erythrozytenvolumens (vEV) dar und zeigt, dass Frauen bereits bei geringerem Blutverlust transfundiert werden als Männer.

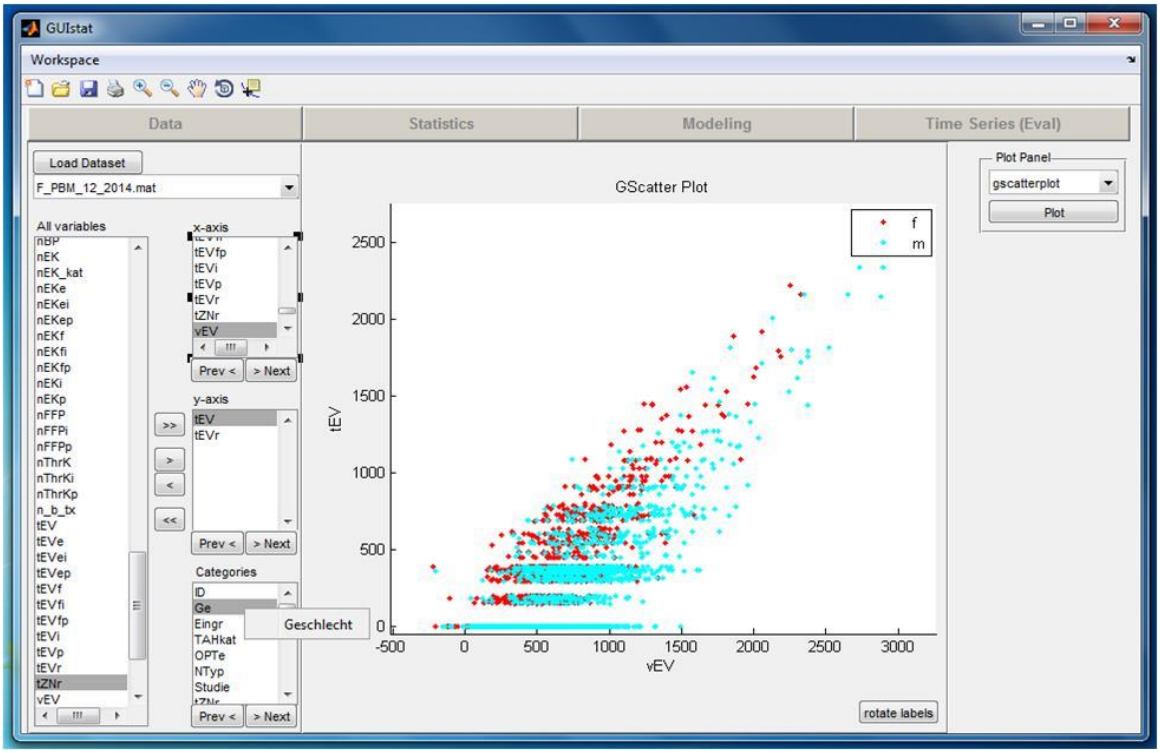

Abbildung 45: Statistik-Tool: Gruppierter Scatterplot

3.2.5 Modellierungs-Tool

Das Modellierungs-Tool in Abbildung 46 dient der Erstellung von Vorhersagemodellen auf Basis eines Featuresets. Diese Version stellt einen ersten Rohentwurf dar und beinhaltet ausschließlich die minimal notwendigen Elemente, um ein Modell generieren zu können.

Im Select-Features-Panel kann eine Teilmenge an Features für die Modellierung ausgewählt werden. Auf der rechten Seite werden die Namen der Modelle und deren Parameter aus dem Model-Definition-File angezeigt. Es handelt sich dabei - wie bei der Feature-Definition-Table zur Erzeugung eines Featuresets - um ein Excel-File, in dem zu generierende Modelle definiert werden können. Nach Auswahl eines Modells kann die Modellierung durch Betätigen des „Train“-Buttons gestartet werden. Wurde das Modell erfolgreich erstellt, so können die Evaluierungsergebnisse zu diesem Modell in der Achse visualisiert werden.

Abbildung 46 zeigt die Area Under The Receiver Operating Characteristic Curve (AUC) für das Modell „M16_4_Sandra_GUI“. Für dieses Modell wurde das gesamte Featureset zum Trainieren des Bagged-Tree-Algorithmus verwendet.

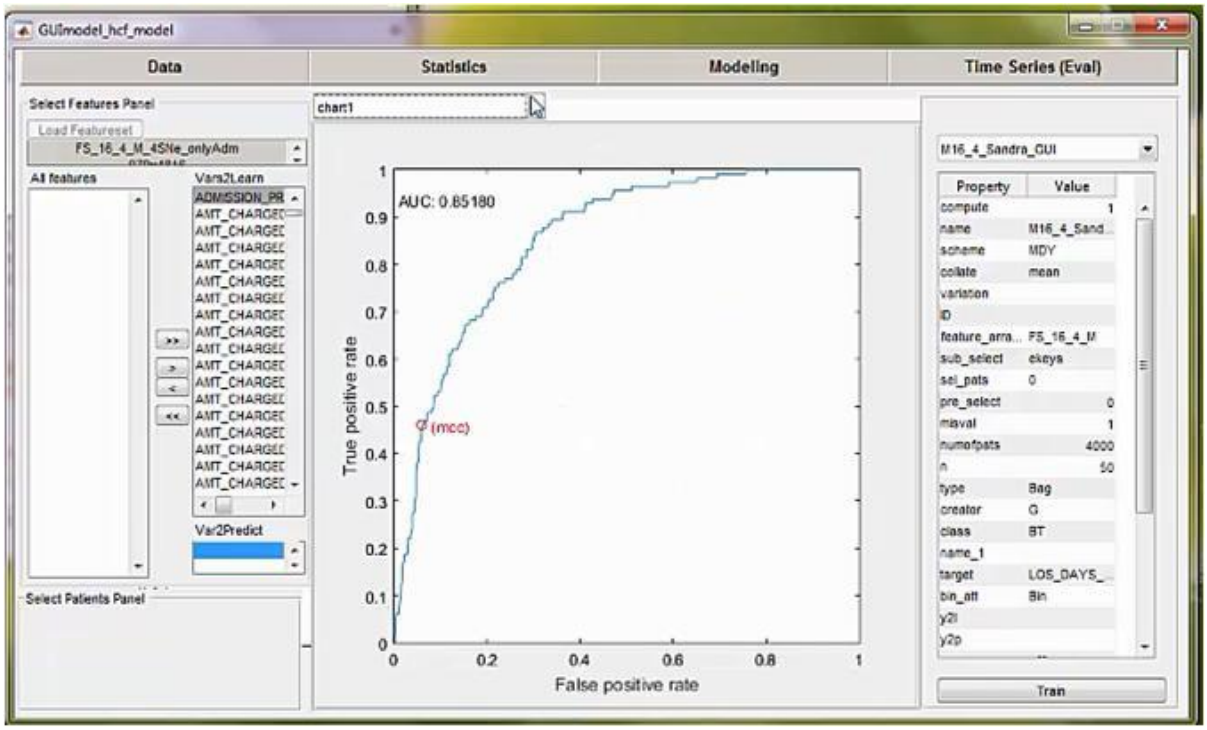

Abbildung 46: Modellierungs-Tool: Area Under The Receiver Operating Characteristic Curve

In Abbildung 47 ist die Vier-Felder-Tafel für das aktuelle Modell visualisiert. Es ist ersichtlich, dass das Modell für 1077 Patienten vorhergesagt hat, dass im Prädiktionszeitraum keine Einweisung ins Krankenhaus stattfinden wird. 1025 dieser Patienten sind in diesem Zeitraum tatsächlich nicht im Krankenhaus vorstellig geworden. 52 Patienten hingegen hatten eine Hospitalisierung aufzuweisen.

Zudem wären laut Modell im Prädiktionszeitraum 127 Patienten ins Krankenhaus eingewiesen worden. Tatsächlich wurden von dieser Patientengruppe nur 61 Patienten hospitalisiert.

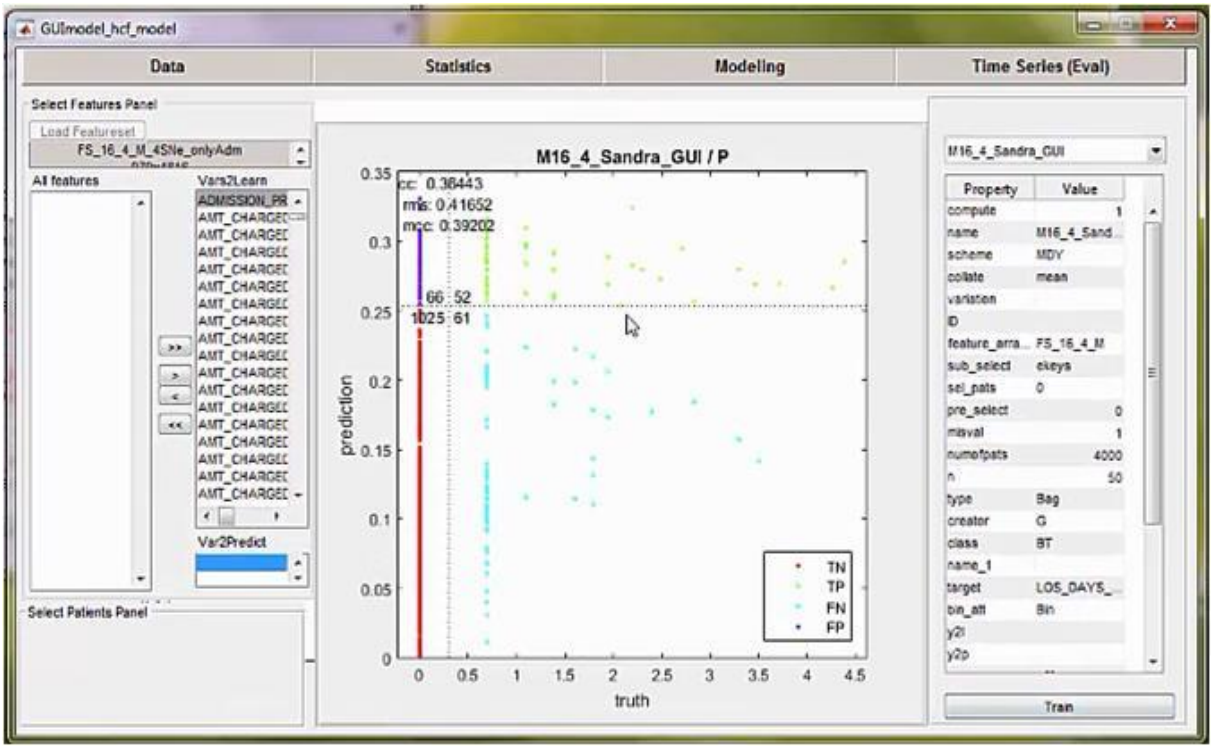

Abbildung 47: Modellierungs-Tool: Vier-Felder-Tafel

Abbildung 48 zeigt den Out-of-Bag-Error, der eine spezifische Kenngröße des „Bagged Tree“-Algorithmus ist. Er variiert mit der Anzahl an erzeugten Bäumen und unterstützt die Wahl einer optimalen Anzahl von Bäumen. Im gegebenen Fall scheint eine Steigerung der Zahl der Bäume über 30 hinaus keine konsistente Verbesserung der Modellvorhersagen mehr zu bewirken.

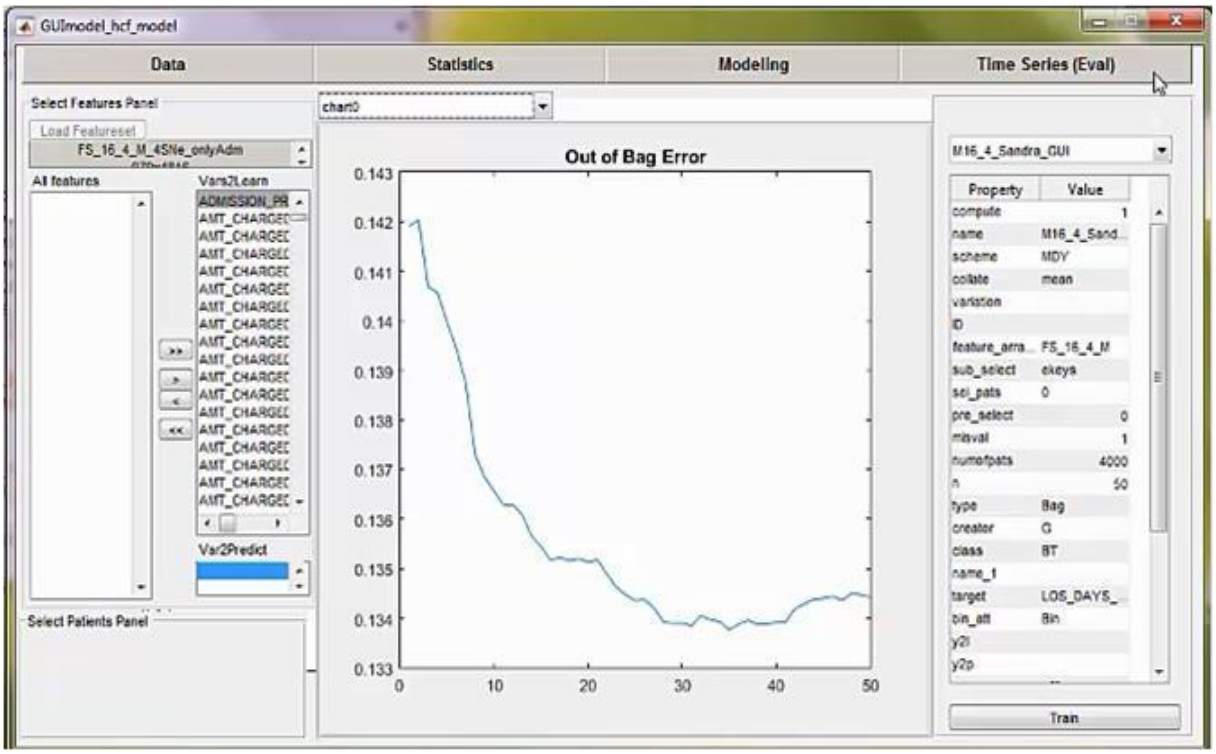

Abbildung 48: Modellierungs-Tool: Out-of-Bag-Error

4 Diskussion

4.1 Feature-Extraktion

Der Theorie zufolge zeichnet sich beim induktiven Lernen ein guter Trainingsdatensatz unter anderem durch eine Balance zwischen dem Stichprobenumfang und der Dimensionalität des Datensatzes aus. Oftmals ist die Anzahl an Patienten gering im Vergleich zur Anzahl an Features. In diesem Fall werden Methoden aus dem Bereich „Feature-Selektion“ herangezogen, um die prädiktionskräftigen Features von den weniger informativen zu trennen und das optimale Subset an Trainingsdaten dem Lernalgorithmus zur Verfügung zu stellen. Methoden der „Feature-Extraktion“ finden Anwendung, wenn sich hochdimensionale Features unter den Lerndaten befinden. Darunter sind Features mit sehr vielen Ausprägungen zu verstehen, wie beispielsweise das nach ICD-10 codierte „Hauptdiagnose“-Feature mit etwa 13 000 Kategorien. Damit der Informationsgehalt dieses Features nicht verloren geht, müssen daraus weitere Features mit weniger Kategorien extrahiert werden, sodass sich die Wahrscheinlichkeit erhöht, zumindest den Großteil der Kategorien mit bekannten Beispielen aus den vorhandenen Patientenfällen besetzen zu können.

In diesem Projekt wurde der „Bagged Tree“-Algorithmus zur Modellgenerierung verwendet. Dieser zeichnet sich dadurch aus, dass er mit einer hohen Dimensionalität in der Breite (siehe Seite 4) sehr gut umgehen kann. Das Problem der Tiefe stellt jedoch auch dieses Ensemble-Verfahren vor eine Herausforderung. Aus diesem Grund wurde das Augenmerk vorerst auf die Feature-Extraktion gelegt.

4.1.1 ICD-10 Hauptdiagnosen und AR-DRGs

Nachdem es für die ICD-10 Codes und die AR-DRGs bereits standardisierte Gruppierungsschemata gibt, war es naheliegend, diese für die Dimensionalitätsreduktion heranzuziehen. In einem ersten Schritt wurde lediglich für die ersten beiden Ebenen der ICD-10 Codehierarchie (Abbildung 26) eine Feature-Extraktion durchgeführt, sowie einfache Features aus der DRG Systematik (Abbildung 27) abgeleitet. Diese Features waren Teil eines Experiments, bei dem der Einfluss von Features auf die Performance von Modellen für ausgewählte Klientengruppen des australischen Versicherungsunternehmens untersucht wurde (Xie et al., 2015b). Eine Erkenntnis daraus war, dass einzelne, hierarchisch gruppierte ICD-10 Features im Einflussranking des „Bagged Tree“-Algorithmus einen höheren Stellenwert erlangten als das ursprüngliche ICD-10 Hauptdiagnose-Feature mit den rund 13 000 Ausprägungen bzw. als einzelne binäre Transformationen daraus.

Infolgedessen wurde der Einfluss der Features aus den ersten drei Ebenen der ICD-10 und AR-DRG Codehierarchien genauer untersucht. Die Hypothese war, dass mit zunehmender Anzahl an Patienten im Trainingsdatensatz die Wichtigkeit von Level-2- bzw. Level-3-Features gegenüber jenen aus dem ersten Level zunimmt. Entgegen der Erwartungen konnte diese Hypothese nicht verifiziert werden (Xie et al., 2015a). Es konnte ein weiteres Paper gefunden werden, in dem der Einfluss der ICD-10 Codehierarchie auf die Modell-Performance im Zusammenhang mit der Identifikation unerwünschter Nebenwirkungen von Arzneimitteln untersucht wurde (Zhao et al., 2005). In diesem Fall hat sich gezeigt, dass erwartungsgemäß bei relativ geringen Stichprobengrößen von 48 bis 3586 Probanden, Features der ersten Hierarchieebene eine signifikante Verbesserung der Modellperformance bewirken und bei Verwendung der weiteren Hierarchieebenen keine Verbesserung der Modellgüte mehr zu verzeichnen ist.

Die Unterschiede in den Ergebnissen können einerseits auf das Adressieren verschiedener Fragestellungen und andererseits auf die Verwendung unterschiedlicher Stichproben zurückzuführen sein, wodurch die Theorie der Abhängigkeit der optimalen Feature-Repräsentation von zahlreichen Einflussfaktoren bestärkt wird.

Die Erkenntnis aus den Experimenten ist, dass eine Gruppierung von klinischen Codes zwar einen Mehrwert bringen kann, allerdings ist die Verwendung eines allgemeinen Klassifikationskonzepts nicht immer ausreichend. Demzufolge könnte eine gezieltere Gruppierung der Codes durch Berücksichtigung der spezifischen Fragestellung erfolgsversprechend sein, wobei Visual Analytics Tools, wie das Zeitreihen-(Evaluierungs)-Tool, helfen können.

4.1.2 ICD-10 Codes: Nebendiagnosen

Aus den Nebendiagnosen können Informationen über den Allgemeinzustand des Patienten abgeleitet werden, welcher einen entscheidenden Einfluss auf das Risiko hospitalisiert zu werden haben kann. Erkrankt beispielsweise ein junger, agiler Mensch an einer Grippe, so wird die Wahrscheinlichkeit ins Krankenhaus eingewiesen zu werden bei ihm geringer sein als bei einem älteren, multimorbiden Menschen.

Erste Analysen haben gezeigt, dass dieses Feature die Prädiktionskraft des Modells nicht entscheidend verbessert. Der Grund dafür liegt sehr wahrscheinlich darin, dass es sich beim verwendeten Datensatz um Versicherungsdaten handelt, welche nur dünn mit Klienten besiedelt sind, die über Sekundärdiagnosen verfügen.

4.2 Visual Analytics

Maschinelles Lernen ist zwar äußerst hilfreich, um die optimale Auswahl an Parametern zu finden und die Informationen von Features schnell und optimal miteinander in Beziehung zu setzen. Dennoch gibt es zahlreiche Herausforderungen, die in diesem Zusammenhang zu bewältigen sind. Eine davon liegt in der Intransparenz maschinell generierter Modelle. Insbesondere Ensemble-Verfahren bilden derart komplexe Modelle, sodass der Weg zum Prädiktionsergebnis nicht mehr nachvollziehbar ist. Demzufolge ist die Modellgenerierung einer Blackbox gleichzusetzen, die mit Trainingsdaten gefüttert wird und am hinteren Ende ein Ergebnis liefert. Visual Analytics Tools bieten die Möglichkeit Daten und Ergebnisse von unterschiedlichen Perspektiven aus zu beleuchten und somit ein tieferes Verständnis für die Prozesse zu erhalten, die hinter den Kulissen ablaufen. Die daraus gewonnenen Erkenntnisse können schließlich wieder zur Optimierung der Modelle eingesetzt werden.

4.2.1 Zeitreihen-Tool

Das Konzept des Zeitreihen-(Evaluierungs)-Tools folgt dem sogenannten „Human-In-The-Loop“-Prinzip (Abbildung 50). Aufgrund einer interaktiven und explorativen Analyse der Modellergebnisse auf Patientenebene kann ein tieferes Verständnis für die Stärken und Schwächen der Vorhersagemodelle erlangt werden. Die gleichzeitige Visualisierung der Daten zu einem Patienten, die aus unterschiedlichen Quellen stammen, schafft einen umfassenden Überblick über die Einflussfaktoren und deren Zusammenhänge. Verborgene Muster, die vom Algorithmus nicht erkannt werden, können auf diese Weise vom Menschen identifiziert und in Form von Features wieder dem Lernalgorithmus zur Verfügung gestellt werden.

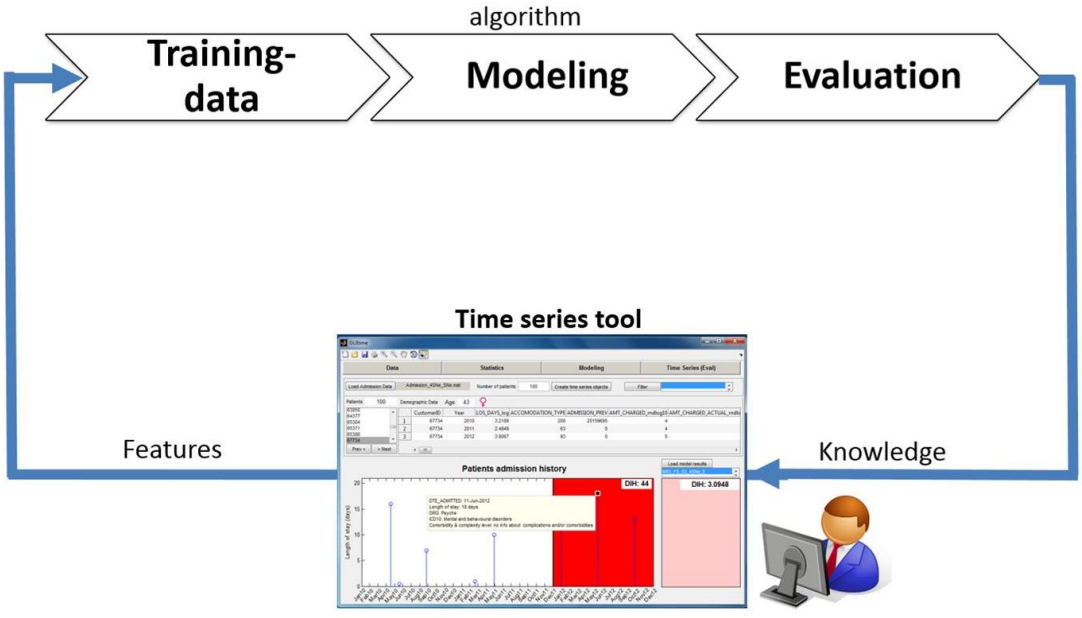

Abbildung 50: Human-In-The-Loop

Abbildung 51 illustriert den Nutzen des Zeitreihen-(Evaluierungs)-Tools am Beispiel der Daten aus dem Gesundheitsdialog-Diabetes-Projekt.

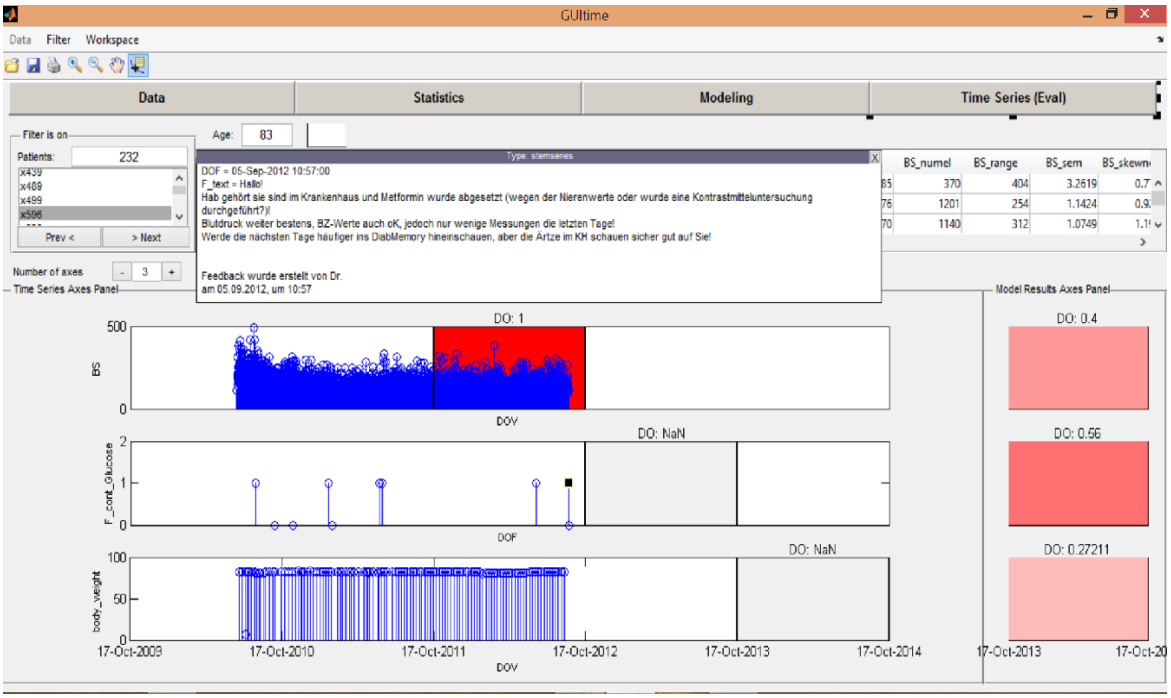

Abbildung 51: Zeitreihen-Tool: Beispiel aus dem Telemonitoring-Projekt „Gesundheitsdialog Diabetes“

In der ersten Zeitachse ist der Verlauf der Blutzuckerwerte des 83-jährigen Patienten „x596“ dargestellt. Es ist ersichtlich, dass er im Jahr 2012 aus dem Telemonitoring-Programm ausgestiegen ist und das Modell für den Zeitraum von Oktober 2011 bis Oktober 2012 eine Droupout-Wahrscheinlichkeit von 40 % vorhergesagt hat. Das Modell in der zweiten Ebene prädiktiert für den nachfolgenden Zeitraum zwischen Oktober 2012 und Oktober 2013 eine Austritts-Wahrscheinlichkeit von 56 %, obwohl der Patient bereits im Vorjahr aus dem Programm ausgestiegen ist. Ebenso zeigt das dritte Modell eine Dropout-Wahrscheinlichkeit von rund 27 % für den Zeitraum 2013/2014 an. Offensichtlich muss dem Lernalgorithmus noch ein Feature zur Verfügung gestellt werden, das die Information beinhaltet, ob es bereits ein Dropout gegeben hat oder nicht. Zudem ist in der zweiten Zeitreihenachse der Verlauf des Arzt-Feedbacks dargestellt. Aus dem letzten Feedback geht hervor, dass sich der Patient unmittelbar vor dem Ausstieg aus dem Telemonitoring-Programm im Krankenhaus befunden hat. Nun müsste nach den Dropout-Patienten gefiltert und überprüft werden, ob ein Krankenhausaufenthalt ein potentieller Indikator für das Aussteigen aus dem Programm sein könnte.

4.2.1.1 Zeitreihen-Tool Erwartungen

Nachfolgende Beispiele stammen aus dem HCF-Projekt, das auf die Vorhersage der Anzahl zukünftiger Krankenhausaufenthalte abzielt.

Hypothesen prüfen

Der Anwender verfügt über Domänenwissen, welches Hypothesen zulässt, wie beispielsweise darüber, dass eine schwangere Frau spätestens innerhalb von neun Monaten wieder ins Krankenhaus eingewiesen werden wird. Das sind allgemeine Regeln bzw. Muster, die das Modell erkennen sollte. Aus Abbildung 52 ist ersichtlich, dass das aktuelle Modell diese Regel noch nicht erkennt. Folgedessen muss ein Feature zur Verfügung gestellt werden, das dem Algorithmus dabei unterstützt, diese Fälle zu identifizieren.

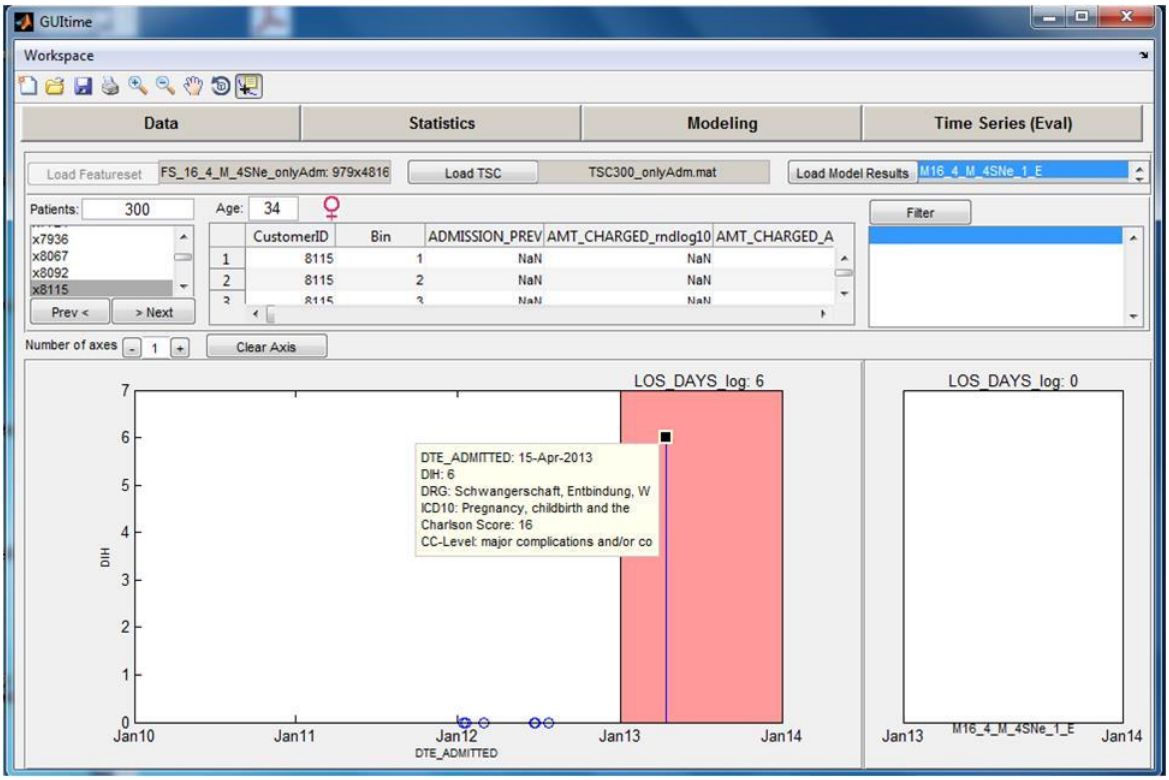

Abbildung 52: Zeitreihen-Tool-Erwartungen: Hypothesen prüfen

Mustererkennung

Nachdem der Mensch sehr gut in der Lage ist, Muster schnell visuell zu erkennen, besteht eine Erwartung darin, noch unbekannte Muster aufzudecken und zu ergründen, in der Hoffnung wieder ein allgemein gültiges Feature in den Trainingsdatensatz einbinden zu können.

In Abbildung 53 ist die Hospitalisierungsgeschichte einer 50 jährigen Frau dargestellt. Die Periodizität in den Krankenhauseinweisungen sticht sofort ins Auge. Nachdem bei der Frau eine Tumorerkrankung diagnostiziert wurde und es sich bei den periodisch wiederkehrenden Ereignissen um halbtägige Aufenthalte handelt, lässt dies auf eine Strahlentherapie rückschließen, die üblicherweise fraktioniert erfolgt und somit vom Algorithmus erkannt werden könnte.

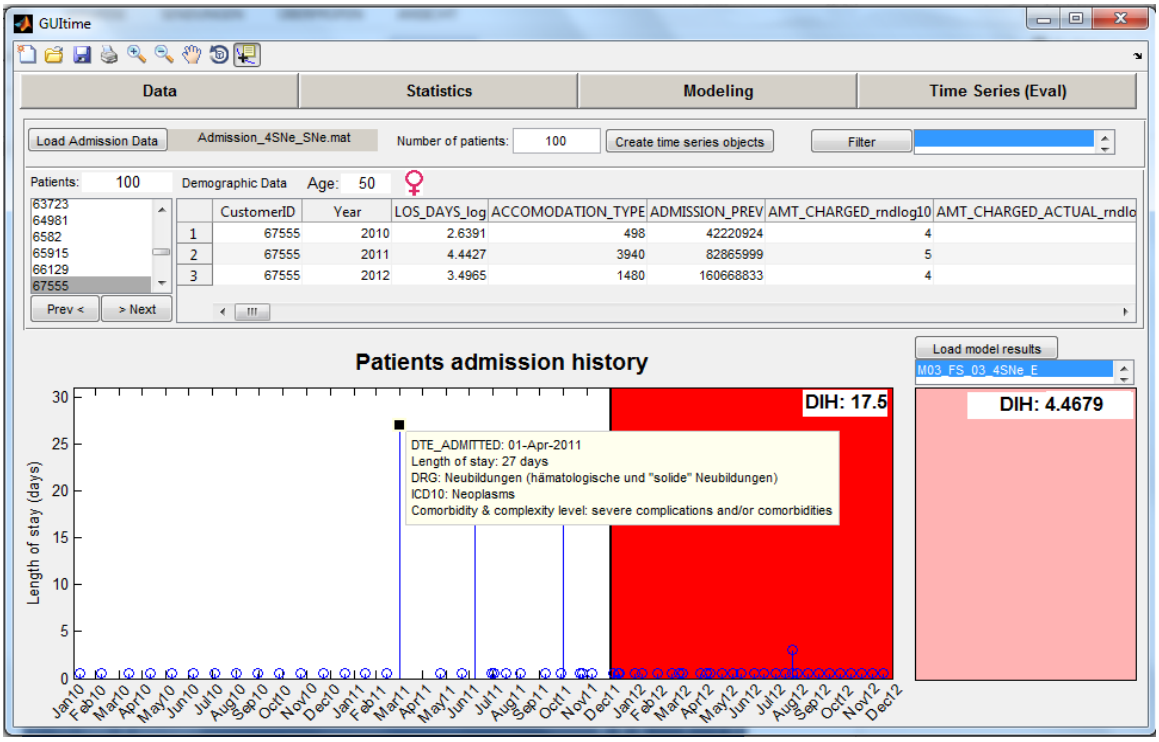

Abbildung 53: Zeitreihen-Tool-(Prototyp)-Erwartungen: Mustererkennung

Modelle vergleichen

Durch die Möglichkeit mehrere Modelle gleichzeitig zu visualisieren, können diese für einzelne Patientenfälle und –gruppen sehr einfach miteinander verglichen werden.

Wie in Abbildung 54 ersichtlich ist, liefert das obere Modell für die 104-jährige Frau ein Falsch-Positives-Ergebnis, wohingegen das untere Modell für diesen Patientenfall eine richtige Prädiktion liefert. Ursächlich könnte sein, dass das obere Modell das Feature „Alter“ stärker gewichtet und das untere Modell mehr Wert auf vergangene Hospitalisierungen oder medizinische Features legt.

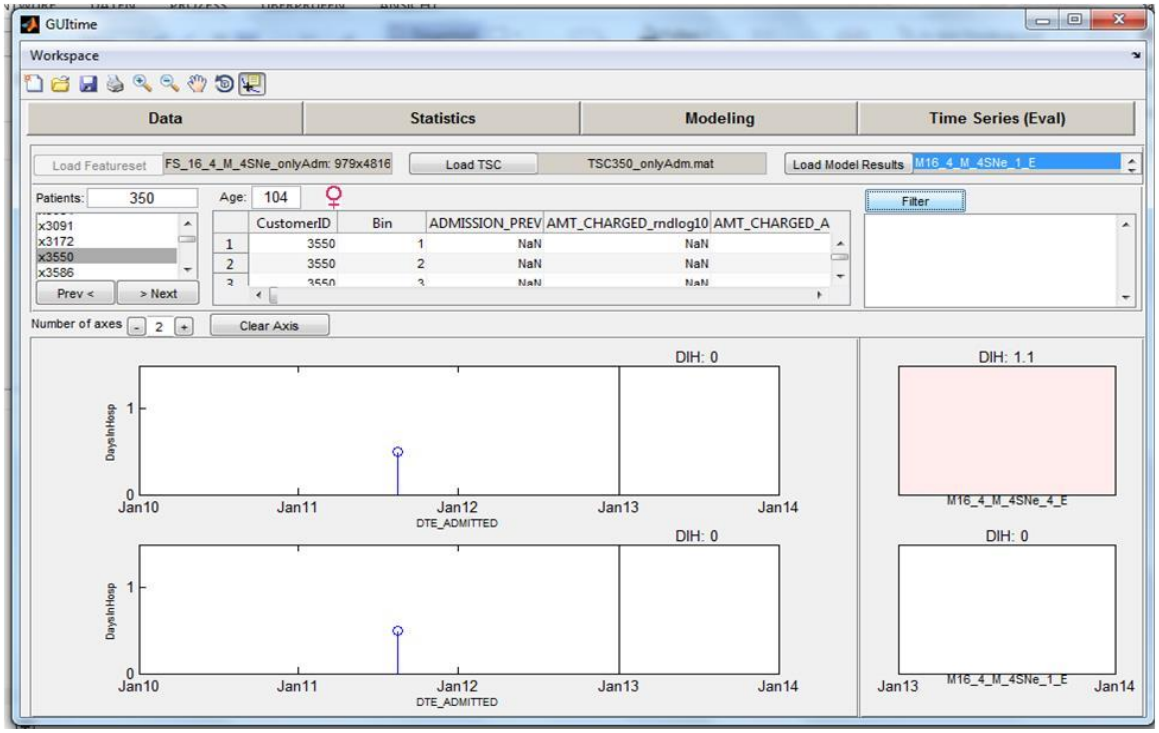

Abbildung 54: Zeitreihen-Tool-Erwartungen: Modelle vergleichen

4.2.2 Statistik-Tool

Das Statistik-Tool kann noch nicht als Visual Analytics Tool bezeichnet werden. Es bildet lediglich die Basis für die Entwicklung eines solchen. Beispielsweise könnte eine Erweiterung die Möglichkeit zur interaktiven Bereinigung des Datensatzes sein, sodass im Plot einzelne Patienten, wie jene mit einem negativen Alter, markiert und gelöscht werden können.

4.2.3 Modellierungs-Tool

Ebenso befindet sich das Modellierungs-Tool noch im Anfangsstadium der Entwicklung. Denkbar wäre hier das interaktive Wählen der Modellparameter, das Hinzufügen einer Option zur Durchführung einer Cross-Validation, eine AUC, die sich verändert, sobald ein neues Feature zum Antrainieren des Lernalgorithmus hinzugefügt wird, das interaktive Eruieren der vom Lernalgorithmus durchgeführten Gewichtung der Features, oder das visuelle Evaluieren mehrerer Modelle gleichzeitig (Abbildung 55).

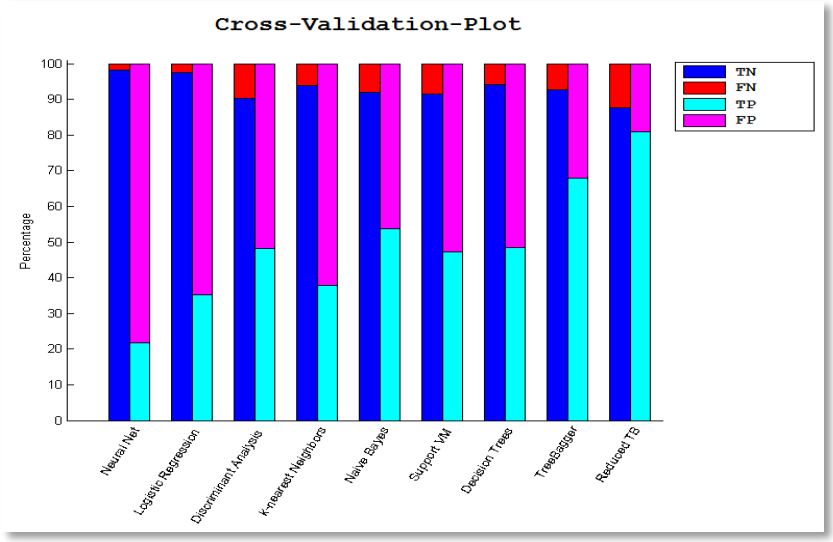

Abbildung 55: Potentielle Erweiterung des Modellierungs-Tools: gleichzeitige Evaluierung von Modellen unterschiedlicher Algorithmen

Letztendlich wird insbesondere das Zusammenspiel der Tools des Frameworks einen Mehrwert für den Entwickler bringen. So ist beispielsweise denkbar, dass der User in Zukunft nur die Falsch-Negativen-Fälle im Zeitreihen-Tool analysieren möchte. Diese sollte er in der Vier-Felder-Tafel einfach markieren können, um sie im Zeitreihen-Tool zur Anzeige zu bringen (Abbildung 56).

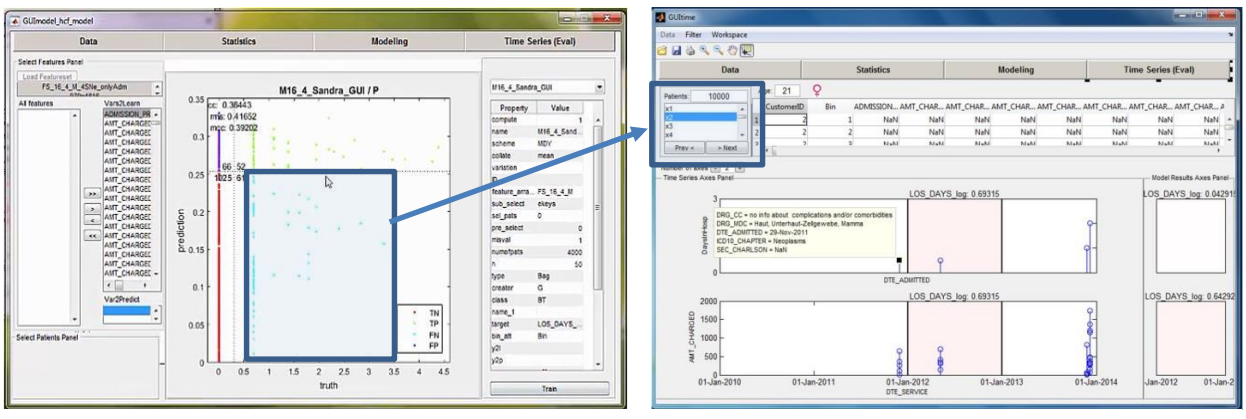

Abbildung 56: Potentielle Interaktion zwischen Tools des Frameworks

5 Zusammenfassung und Ausblick

Das Generieren von Wissen aus Big Data ist wesentlich mehr als nur das einfache Anwenden eines maschinellen Lernalgorithmus. Es handelt sich dabei um einen Prozess, der sich von der Stichprobenauswahl, Datenvorverarbeitung, Datentransformation, Auswahl und Bedienung eines geeigneten maschinellen Lernalgorithmus, Evaluierung bis hin zur Interpretation der Modellergebnisse erstreckt.

Entscheidend für die Modellgüte ist ein qualitativ hochwertiger Trainingsdatensatz, welcher durch eine optimale Repräsentation der Features und deren Informationsgehalt bezüglich der zu prädiktierenden Variable charakterisiert ist. Die Herausforderung bei der Aufbereitung dieser Lerndaten besteht einerseits in der Abhängigkeit der einzelnen Prozessschritte untereinander, sowie in der fehlenden Transparenz der maschinell generierten Modelle.

Abhilfe können Visual Analytics Tools bieten, die als Schnittstelle zwischen Mensch und Maschine das Domänenwissen und die Stärken des Menschen zur Mustererkennung in den maschinellen Lernprozess miteinfließen lassen können. Die besten Ergebnisse können erzielt werden, wenn sich der intelligente Mensch und die schnelle Maschine über Visual Analytics Tools die Hand reichen.

Ausblick:
Letztendlich besteht das Ziel darin, das gewonnene Wissen in die Praxis umzusetzen. Hierfür braucht es zum einen Experten und zum anderen eine höhere Transparenz der Modellergebnisse. Es werden neben Modellen zukünftig auch Tools erforderlich sein, die dem Experten eine interaktive Analyse der Modellergebnisse erlaubt, sodass sich dieser das vom Modell bereitgestellte Wissen aneignen kann, um somit die richtigen Entscheidungen treffen zu können.

Schlussfolgerung:
Auch die Medizin bewegt sich in Richtung einer zunehmend datenzentrierten Domäne, die sowohl für Patienten als auch für das Gesundheitssystem enormes Verbesserungspotential mitbringt. Die ersten Schritte in die richtige Richtung sind getan, doch wir befinden uns erst am Anfang eines sehr, sehr langen Weges…

… who cares – *„The journey of a thousand miles begins with one step“ (Lao Tzu).*

Abbildungsverzeichnis

Abbildung 1: The Knowledge Pyramid (Vgl. Landauer, 1998) 2

Abbildung 2: Predictive Modelling mit maschinellen Lernverfahren 3

Abbildung 3: The Knowledge Pyramid (Landauer, 1998): Knowledge Discovery/Predictive Modelling 3

Abbildung 4: Maschineller Lernprozess 4

Abbildung 5: The Knowledge Pyramid (Landauer, 1998): Data Mining zur Unterstützung des Predictive Modellings 4

Abbildung 6: The Knowledge Pyramid (Landauer, 1998): Data Mining, Predictive Modelling und Visual Analytics 5

Abbildung 7: Predictive Modelling Pipeline 6

Abbildung 8: Prädiktionsvorhaben 7

Abbildung 9: Hierarchie der Datenstruktur 8

Abbildung 10: MATLAB-Dataset 8

Abbildung 11: Metadaten des Featuresets: UserData mit Kategorie-Bezeichnungen 9

Abbildung 12: Bestehende Routinen der unternehmensinternen Predictive Modelling Pipeline 9

Abbildung 13: Standardisierte Ordnerstruktur der Predictive Modelling Pipeline 10

Abbildung 14: ICD-10 Feature-Extraktion 11

Abbildung 15: Struktur zur Verwaltung von Daten des Frameworks 13

Abbildung 16: Matlabs „application data“: Speichern und Abrufen 13

Abbildung 17: Metadaten des Featuresets: Rohdatensätze 14

Abbildung 18: GUIcreateFeatureSet für die prä-existente Featureset-Generierungs-Routine 15

Abbildung 19: Zeitreihenkollektion (tsc) mit den Zeitreihenobjekten 16

Abbildung 20: Zeitreihen-Tool: Prototyp 16

Abbildung 21: Metadaten der Zeitreihenachsen (links) und der Modellachsen (rechts) 17

Abbildung 22: Informationstransfer zwischen Featureset und Metadaten der Zeitreihenachsenobjekte 17

Abbildung 23: Informationstransfer zwischen GUITimeSeriesAxis und Metadaten der Zeitreihenachsenobjekte 18

Abbildung 24: Informationstransfer zu den Metadaten der Modellachsenobjekte 19

Abbildung 25: Zwischenspeichern der Ergebnisse der Teilfilterungen 20

Abbildung 26: Ableiten von Features aus ICD-10 Hauptdiagnose-Codes 22
Abbildung 27: Ableiten von Features aus der AR-DRG Hierarchie (Fischer, 2001).....25
Abbildung 28: Ableiten von Features aus dem semi-hierarchischen DRG-Schema 26
Abbildung 29: Ausschnitt aus dem Visual Analytics Framework 27
Abbildung 30: Data-Tool (GUIhome) 27
Abbildung 31: Laden der Zeitreihenkollektion 28
Abbildung 32: Datenmanagement des Frameworks 29
Abbildung 33: GUIcreateFeatureSet.....30
Abbildung 34: *GUIcreateTSCobj* zur Erstellung von Zeitreihenobjekten 30
Abbildung 35: *GUIcreateTSCobj*: Erzeugung der Zeitreihenobjekte (1) und Speichern (2) ...31
Abbildung 36: Zeitreihen-Tool.....32
Abbildung 38: Zeitreihen-Tool-Kernelemente 33
Abbildung 38: Zeitreihen-Tool-Kernelemente 33
Abbildung 39: Zeitreihen-Tool: Doppelklick auf Zeitreihenachse (GUITimeSeriesAxis).....34
Abbildung 40: Zeitreihen-Tool: Doppelklick auf Modellachse 35
Abbildung 41: Zeitreihen-Tool: Filter-GUI (GUITimeSeriesFilter) 36
Abbildung 42: Statistik-Tool (GUIstat).....37
Abbildung 43: Statistik-Tool: Histogramm 38
Abbildung 44: Statistik-Tool: Boxplots 39
Abbildung 45: Statistik-Tool: Gruppierter Scatterplot 39
Abbildung 46: Modellierungs-Tool: Area Under The Receiver Operating Characteristic Curve 40
Abbildung 47: Modellierungs-Tool: Vier-Felder-Tafel 41
Abbildung 48: Modellierungs-Tool: Out-of-Bag-Error 41
Abbildung 49: Zeitreihen-Tool: Zeitreihen-Ebenen.....42
Abbildung 50: Human-In-The-Loop.....44
Abbildung 51: Zeitreihen-Tool: Beispiel aus dem Telemonitoring-Projekt „Gesundheitsdialog Diabetes“.....45
Abbildung 52: Zeitreihen-Tool-Erwartungen: Hypothesen prüfen.....46
Abbildung 53: Zeitreihen-Tool-(Prototyp)-Erwartungen: Mustererkennung 47
Abbildung 54: Zeitreihen-Tool-Erwartungen: Modelle vergleichen 47
Abbildung 55: Potentielle Erweiterung des Modellierungs-Tools: gleichzeitige Evaluierung von Modellen unterschiedlicher Algorithmen 48

Abbildung 56: Potentielle Interaktion zwischen Tools des Frameworks 48

Tabellenverzeichnis

Tabelle 1: ICD-10-Level-1-Feature: Krankheitskapitel 23

Tabelle 2: ICD-10-Level-2-Feature: Neubildungen 23

Tabelle 3: ICD-10-Level-3-Features: In situ Neubildungen 23

Tabelle 4: Ursprüngliche und upgedatete Charlson-Gewichte (Vgl. Quan et al., 2011, Sundararajan et al., 2004) 24

Tabelle 5: DRG-Feature: Major Diagnostic Categories (DRG_MDC) 25

Tabelle 6: DRG-Feature: Sub-Major-Diagnostic-Categories (DRG_SUB_MDC) 26

Tabelle 7: DRG-Feature: Comorbidity or Complication (DRG_CC) 26

Literaturverzeichnis

ALTMAN, Y. M. 2011. *Undocumented secrets of MATLAB-Java programming*, CRC Press.

CHARLSON, M. E., POMPEI, P., ALES, K. L. & MACKENZIE, C. R. 1987. A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. *Journal of chronic diseases,* 40**,** 373-383.

DONSA, K., SPAT, S., BECK, P., PIEBER, T. R. & HOLZINGER, A. 2015. Towards personalization of diabetes therapy using computerized decision support and machine learning: some open problems and challenges. *Smart Health.* Springer.

DONZÉ, J., AUJESKY, D., WILLIAMS, D. & SCHNIPPER, J. L. 2013. Potentially avoidable 30-day hospital readmissions in medical patients: derivation and validation of a prediction model. *JAMA internal medicine,* 173**,** 632-638.

FISCHER, W. 2001. Grundzüge von DRG-Systemen. *ARNOLD, M./LITSCH, M. und H. SCHELLSCHMIDT (Hrsg.)***,** 13-32.

GANTZ, J. & REINSEL, D. 2012. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the Future,* 2007**,** 1-16.

GUYON, I. 2006. *Feature extraction: foundations and applications*, Springer Science & Business Media.

HALFON, P., EGGLI, Y., PRÊTRE-ROHRBACH, I., MEYLAN, D., MARAZZI, A. & BURNAND, B. 2006. Validation of the potentially avoidable hospital readmission rate as a routine indicator of the quality of hospital care. *Medical care,* 44**,** 972-981.

HALL, M. A. 1999. *Correlation-based feature selection for machine learning.* The University of Waikato.

HASAN, O., MELTZER, D. O., SHAYKEVICH, S. A., BELL, C. M., KABOLI, P. J., AUERBACH, A. D., WETTERNECK, T. B., ARORA, V. M., ZHANG, J. & SCHNIPPER, J. L. 2010. Hospital readmission in general medicine patients: a prediction model. *Journal of general internal medicine,* 25**,** 211-219.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., HASTIE, T., FRIEDMAN, J. & TIBSHIRANI, R. 2009. *The elements of statistical learning*, Springer.

HEALTH, N. C. F. C. I. 2004. *The International Statistical Classification of Diseases and Related Health Problems, 10th Revision, Australian Modification (ICD-10-AM)*, National Centre for Classification in Health.

IDC 2014. The Digital Universe: Driving Data Growth in Healthcare. *EMC DIGITAL*.

KOTSIANTIS, S. B., ZAHARAKIS, I. & PINTELAS, P. 2007. Supervised machine learning: A review of classification techniques.

LANDAUER, C. Data, information, knowledge, understanding: computing up the meaning hierarchy. Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on, 1998. IEEE, 2255-2260.

LANGLEY, P. 1994. *Selection of relevant features in machine learning*, Defense Technical Information Center.

LENT, C. S. 2013. *Learning to Program with MATLAB: Building GUI Tools: Building GUI Tools*, Wiley Global Education.

MAYER-SCHÖNBERGER, V. & CUKIER, K. 2013. *Big data: A revolution that will transform how we live, work, and think*, Houghton Mifflin Harcourt.

NAISBITT, J. 1982. *Megatrends: Ten New Directions Transforming Our Lives*, Warner Books.

PANAHIAZAR, M., TASLIMITEHRANI, V., JADHAV, A. & PATHAK, J. Empowering personalized medicine with big data and semantic web technology: Promises, challenges, and use cases. Big Data (Big Data), 2014 IEEE International Conference on, 2014. IEEE, 790-795.

PECHENIZKIY, M. 2005. The impact of feature extraction on the performance of a classifier: kNN, Naïve Bayes and C4. 5. *Advances in Artificial Intelligence.* Springer.

QUAN, H., LI, B., COURIS, C. M., FUSHIMI, K., GRAHAM, P., HIDER, P., JANUEL, J.-M. & SUNDARARAJAN, V. 2011. Updating and validating the Charlson comorbidity index

and score for risk adjustment in hospital discharge abstracts using data from 6 countries. *American journal of epidemiology,* 173**,** 676-682.

SONG, X., MITNITSKI, A., COX, J. & ROCKWOOD, K. 2004. Comparison of machine learning techniques with classical statistical models in predicting health outcomes. *Medinfo,* 11**,** 736-40.

SUNDARARAJAN, V., HENDERSON, T., PERRY, C., MUGGIVAN, A., QUAN, H. & GHALI, W. A. 2004. New ICD-10 version of the Charlson comorbidity index predicted in-hospital mortality. *Journal of clinical epidemiology,* 57**,** 1288-1294.

TAN, P.-N., STEINBACH, M. & KUMAR, V. 2006. *Introduction to data mining*, Pearson Addison Wesley Boston.

WHO 2011. International statistical classification of diseases and related health problems. - 10th revision. *WHO Library Cataloguing-in-Publication Data,* 2.

WU, J., ROY, J. & STEWART, W. F. 2010. Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. *Medical care,* 48**,** S106-S113.

XIE, Y., NEUBAUER, S., SCHREIER, G., REDMOND, S. J. & LOVELL, N. H. 2015a. Impact of Hierarchies of Clinical Codes on Predicting Future Days in Hospital. *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE.* IEEE.

XIE, Y., SCHREIER, G., CHANG, D., NEUBAUER, S., LIU, Y., REDMOND, S. & LOVELL, N. 2015b. Predicting Days in Hospital Using Health Insurance Claims.

XIE, Y., SCHREIER, G., CHANG, D. C., NEUBAUER, S., REDMOND, S. J. & LOVELL, N. H. Predicting number of hospitalization days based on health insurance claims data using bagged regression trees. Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, 2014. IEEE, 2706-2709.

ZHAO, Y., ASH, A. S., ELLIS, R. P., AYANIAN, J. Z., POPE, G. C., BOWEN, B. & WEYUKER, L. 2005. Predicting pharmacy costs and other medical costs using diagnoses and drug claims. *Medical care,* 43**,** 34-43.