KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Clinical Data Miner

Towards more efficient clinical study support

Arnaud Installé

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

June 2014

Clinical Data Miner

Towards more efficient clinical study support

Arnaud INSTALLÉ

Examination committee:
Prof. dr. A. Bultheel, chair
Prof. dr. ir. B. De Moor, supervisor
Prof. dr. D. Timmerman, co-supervisor
Prof. dr. ir. J. Suykens
Prof. dr. B. Van Calster
Dr. Thierry Van den Bosch
Prof. dr. ir. W. Joosen
Prof. dr. T. Bourne
(Imperial College, London (UK))

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

June 2014

© 2014 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Arnaud Installé, Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

ISBN 978-94-6018-859-6

D/2014/7515/83

Preface

In this thesis, I summarize research work that I performed as a PhD student since 2009, within the STADIUS research division of the KU Leuven Department of Electrical Engineering (ESAT). During these years, I have had the opportunity to work in a supporting environment, and to collaborate with many inspiring, leading researchers.

My deepest gratitude goes to my supervisor, Prof. Bart De Moor. Bart has provided me with the opportunity to join the STADIUS group, to work under his supervision. He has provided me the valuable freedom to carve out a project tailored to myself, and has enabled me to attend conferences, which has enabled me to network with many interesting and leading researchers.

I also would like to extend my sincere gratitude to my co-supervisor, Prof. Dirk Timmerman. My first meeting with him planted the seed for the Clinical Data Miner (CDM) project. I would like to thank him for his continuous support, for providing his invaluable user perspective about the necessary requirements for an Electronic Data Capture software tool, for introducing me to the International Endometrial Tumour Analysis (IETA) community, for his feedback and collaboration. I equally would like to thank Dr. Thierry Van den Bosch, for his support and feedback. Between the three of us, we have spent countless meetings prioritizing the features to be implemented in Clinical Data Miner (CDM), as well as discussing analyses I had performed on the data from the International Ovarian Tumour Analysis consortium.

I am indebted to the members of my jury, who have supported my PhD research. I would especially like to thank Prof. Johan Suykens and Prof. Ben Van Calster, whose contributions have helped me to improve my work methodologically. My thanks also go to Prof. Dr. Tom Bourne, for providing me with opportunities to collaborate on interesting research projects. I am grateful to Prof. Wouter Joosen, for taking an interest in my work, and to Prof. Adhemar Bultheel, for accepting to preside my jury.

I am grateful to the members of the International Endometrial Tumour Analysis (IETA) consortium, for submitting patients, and for providing me feedback about their experience with the CDM user interface. In particular, I would like to thank those who have given me the opportunity to extend CDM's use beyond just the studies of the IETA consortium. They include Dr. Dominique Van Schoubroeck, Prof. Dr. Antonia Testa, Prof. Dr. George Condous, Dr. Angelo Votino, and Prof. Dr. Lil Valentin. I should further thank Lil for her valuable feedback during the development of CDM's Electronic Data Capture (EDC) component.

Next, I want to acknowledge the support of my colleagues at BIOI (Amin Ardeshirdavani, Anneleen Daemen, Sarah Elshal, Olivier Gevaert, Griet Laenen, Charalampos Moschopoulos, Ryo Sakai, Nico Verbeeck, and Raf Winand), for providing a pleasant work environment, for interesting discussions, and for broadening my perspective on the world. I would like to thank Peter Konings and Yousef El Aalamat in particular, for offering their statistical expertise, as well as Marc Claesen and Dusan Popovic, for their machine-learning insights. I would further like to thank Inge Thijs, for helping me author a funding proposal with Dirk and Bart, enabling me to continue work on CDM beyond my doctoral research.

I am further obliged for the help I received on many occasions from ESAT's administrative staff, including Ida Tassens, John Vos, Mimi Deprez, and Ilse Pardon.

I would like to thank my parents, sister and parents-in-law, for their continuous support.

Finally, I thank both my children, Janne and Gijs, for always succeeding in bringing a smile to my face. Last but not least, I very warmly thank my beloved wife, Annemie, the bedrock of our little family, whose relentless support was instrumental in the completion of my research and manuscript.

Arnaud

Rotselaar
June 2014

Abstract

Early, accurate diagnosis of disease can dramatically improve prognosis. Clinical diagnostic model research attempts to optimize early diagnosis by designing diagnostic models based on variables obtained by the least invasive means. Diagnostic model research currently involves a complex, multidisciplinary workflow involving data collection by clinicians on the one hand, and data preprocessing and machine-learning by machine-learning experts on the other.

Due to the traditional lack of integration between software packages used in this workflow, preparing data for analysis can require considerable manual effort. Following data extraction, data have to be inspected for conversion issues. The absence of information about a Case Report Form (CRF)'s structure in extracted data further requires manual guidance during preprocessing. As a result, data analysis is typically only performed once, after the data set reaches a certain predetermined size, based on rules of thumb or Monte Carlo simulations.

This thesis presents the Clinical Data Miner (CDM) software framework, which integrates data collection, data preprocessing and machine-learning in a single platform. This integration eliminates the error-prone, time-consuming steps of preparing data for analysis, and enables the automation of preprocessing steps that rely on information about a CRF's structure. The increased automation streamlines the diagnostic model research workflow. With its built-in functionality for generating learning curves, it furthermore provides study coordinators insight into how predictive performance evolves as patient set sizes grow. This allows them to make an informed decision about whether to continue or terminate data collection, thereby respectively avoiding both the creation of weakly performing models, as well as unnecessary data collection.

Thus, as Electronic Data Capture (EDC) has done for patient data collection, the CDM software framework's functionality should improve the efficiency of diagnostic model studies.

Beknopte samenvatting

Vroege, correcte diagnose van ziektes kan zorgen voor een sterk verbeterde prognose. Klinisch diagnostisch modelonderzoek beoogt de optimalisatie van vroege diagnose door het ontwerp van diagnostische modellen gebaseerd op variabelen die zo min mogelijk invasief bekomen worden. Zulk onderzoek vergt momenteel een complex, multidisciplinair proces van verzameling en voorbewerking van gegevens, en machinaal leren.

Het gebrek aan integratie tussen de in dit proces gebruikte software pakketten vereist menselijke interventie bij de voorbereiding van data voor analyse. Geëxtraheerde data moeten gecontroleerd worden op conversiefouten. Het ontbreken van informatie over de structuur van studievragenlijsten in geëxtraheerde data vergt manuele sturing bij de voorbewerking van gegevens. Bijgevolg wordt data analyse typisch slechts eenmalig toegepast, bij het bereiken van een voorafbepaald patiëntenaantal, berekend op basis van vuistregels of Monte Carlo simulaties.

Deze thesis stelt het Clinical Data Miner (CDM) programmatuurraamwerk voor, dat de verzameling en voorbewerking van data, alsook machinaal leren integreert in één enkel platform. Deze integratie maakt het mogelijk de tijdrovende, foutgevoelige voorbereiding van data voor analyse te vermijden, en maakt de automatisatie van voorbewerkingen mogelijk die structurele informatie van studievragenlijsten vergen, wat het diagnostisch modelonderzoek stroomlijnt.De ingebouwde functionaliteit om leercurves te genereren biedt studiecoördinatoren bovendien inzicht in de evolutie van predictieve performantie bij groeiende patiëntenaantallen, wat hen in staat stelt een weloverwogen keuze te maken over de verderzetting van dataverzameling, waardoor zowel onnodige dataverzameling als de creatie van zwakke modellen kunnen worden vermeden.

Bijgevolg, zoals elektronische datacaptatie (EDC) heeft gedaan voor dataverzameling, zou de functionaliteit van CDM moeten leiden tot een verhoogde efficiëntie van diagnostische modelstudies.

Contents

Abstract iii

Contents ix

List of Figures xv

List of Tables xxi

1 Introduction 1

1.1 Clinical diagnostics 1

1.2 Clinical diagnostic model research 2

1.3 Workflow inefficiencies 4

1.4 Clinical Data Miner 5

1.4.1 Electronic Data Capture 5

1.4.2 Data analysis 6

1.5 Automating machine-learning 7

1.6 Main contributions 8

1.7 Chapter-by-chapter overview 8

2 International Endometrial Tumour Analysis 11

2.1 Introduction 11

2.2 Endometrial findings at histopathology 12
2.3 International Endometrial Tumour Analysis (IETA) consortium 12
2.4 Ultrasound imaging technology 13
2.5 Studies 13
2.6 Data collection 15
2.7 Conclusion 15

3 Electronic Data Capture **17**
3.1 Introduction 17
3.2 Existing software 18
3.3 Requirements 20
3.3.1 Visual user interface elements 24
3.3.2 Case Report Forms 26
3.3.3 Remote Procedure Calls (RPCs) 28
3.3.4 Authentication and access control 31
3.3.5 Database structure 33
3.4 Software development methodology 34
3.4.1 Programming language & frameworks 34
3.4.2 Quality assurance 36
3.4.3 Software configuration management 39
3.5 Architecture 40
3.6 Server setup 42
3.7 Results 43
3.8 Conclusion 45

4 Influence of pictograms on data quality **51**
4.1 Introduction 51
4.2 Methods 53

4.2.1 Study design 53
4.2.2 `ImgStudy` user interface 54
4.2.3 `MediaStudy` user interface 55
4.2.4 Analysis 57
4.3 Results 59
4.3.1 Unenhanced ultrasound 59
4.3.2 Sonohysterography 59
4.4 Conclusion 63

5 Feasibility of automating machine-learning 65
5.1 Introduction 65
5.2 Data set 66
5.3 Classification 67
5.3.1 Logistic regression 68
5.3.2 Support Vector Machines 69
5.3.3 Kernel functions 73
5.3.4 Least-Squares Support Vector Machines 73
5.4 Model evaluation 74
5.5 Learning curves 76
5.6 Analysis 77
5.7 Conclusion 80

6 Data analysis integration 85
6.1 Introduction 85
6.2 Data access 87
6.3 Data preprocessing 89
6.4 Machine-learning 98
6.5 Statistical analysis 100

6.6 Jython interface 101
6.7 Development methodology 102
6.8 Conclusion 104

7 Clinical Data Miner results **107**

7.1 Introduction 107
7.2 International Endometrial Tumour Analysis 107
7.2.1 Participants 108
7.2.2 Inclusions 110
7.3 Other studies 111
7.4 Data analysis example scripts 111
7.4.1 Class distribution 112
7.4.2 Contingency tables 113
7.4.3 Learning curves 115
7.4.4 Model predictions 116
7.5 Conclusion 117

8 Conclusions and future research **121**

8.1 Achievements 121
8.2 Future work 122
8.3 Dissemination 124
8.4 Conclusion 125

A Inter-rater agreement studies **127**

A.1 Influence of pictograms 127
A.2 Polycystic Ovaries (PCOs) 127
A.3 Uterine anomalies 128
A.4 Endomyometrial junction 128

A.5 International Endometrial Tumour Analysis #2 128
A.6 Image enhancement 129

B Case Report Forms **131**

B.1 Effect of pictograms on data quality 131
B.1.1 Unenhanced ultrasound 131
B.1.2 Sonohysterography 132

C Feature selection experiments **135**

C.1 Algorithms 135
C.2 Learning curves 135
C.2.1 Performance on the full data set 136
C.2.2 Performance on reduced data sets 137
C.3 Feature selection performance robustness 139

Bibliography **141**

Curriculum vitae **151**

List of publications **153**

List of Figures

1.1 The clinical diagnostic model research workflow. Steps enclosed in green boxes are currently facilitated by the CDM software framework, while steps enclosed in blue will be in the future. . 3

1.2 CDM's Electronic Data Capture (EDC) user interface. 6

1.3 Example of a learning curve generated by CDM's machine-learning functionality. 7

1.4 Overview of chapters and their mutual dependence. 9

3.1 The use of pictograms to clarify questions in a Case Report Form (CRF). (Screenshot of the "Myometrium" section, included in the IETA #5 study, as presented by the CDM user interface.) 25

3.2 CDM allows to visualize the distinct choices of categorical variables using pictograms. (Screenshot of the "Sonohysterography or Fluid" section, included in the IETA #1, #2, and #3 studies, as presented by the CDM user interface.) 25

3.3 CDM uses a "slider" element to represent Visual Analogue Scales (VASs). (Screenshot of the "Validation" section, included in the IETA #1, #3, and #4 studies, as presented by the CDM user interface.) 25

3.4 CDM implements the "skip pattern" by showing or hiding variables in a box positioned below their "parent" variable, depending on the value of that "parent" variable. (Screenshot of the "Sonohysterography or fluid" section, included in the IETA #1 and #3 studies, as presented by the CDM user interface.) 26

3.5 This shows the `Element` class hierarchy for representing sections, fields, and multiple-choice questions of a questionnaire, as well as the `Crf` class containing one or more `SectionItem`s. 27

3.6 These interfaces are used for processing `Crf` objects. Processing order is determined by the choice of `ItemTreeTraversalStrategy` implementation. 29

3.7 The Remote Procedure Call (RPC) protocol defined between client and server, for supporting CDM's EDC functionality. . . 30

3.8 CDM's pluggable access control is governed by means of `Permission` objects. 32

3.9 CDM's database schema. Tables corresponding with business classes have a white background. Tables with a grey background are responsible for establishing a 1:n relationship between business classes. 35

3.10 The Test-Driven Development workflow. 37

3.11 CDM's high-level architecture consists of a set of separate modules. 41

3.12 CDM's user interface *logic* and *presentation* are decoupled by means of the *AbstractFactory* design pattern represented here. The `NodeFactory` interface is an *AbstractFactory*, implemented by the `GwtNodeFactory` class. 42

3.13 CDM's user interface. (Screenshot of the "Sonohysterography or fluid" section, included in the IETA #1, #2, and #3 studies, as presented by the CDM user interface.) 43

4.1 CRF *without* pictograms for the first phase of the study. (Screenshot of the "Observer variability without images" section, included in the study investigating the influence of pictograms on inter-rater agreement, as presented by the CDM user interface.) 55

4.2 CRF *with* pictograms for the second phase of the study. Except for the added pictograms, the questionnaire is identical to that of Figure 4.1. (Screenshot of the "Observer variability with images" section, included in the study investigating the influence of pictograms on inter-rater agreement, as presented by the CDM user interface.) 56

4.3 Unified Modelling Language (UML) diagram of the RPC call introduced in the inter-rater agreement study user interface. With this call, a user's client can obtain a list of images that remain to be evaluated. 56

4.4 Main server classes responsible for handling of generic study types. 58

4.5 Representation of the RPC call implemented by the generalized inter-rater agreement study user interface. This RPC mechanism allows to obtain a list of `Media` objects that a user still needs to evaluate. 58

4.6 Boxplot comparing inter-rater agreement between phases 1 and 2 for the different unenhanced ultrasound variables. Variables marked with (*) did not have pictograms in either phase of the study. 62

4.7 Boxplot comparing inter-rater agreement between phases 1 and 2 for the different sonohysterography variables. Variables marked with (*) did not have pictograms in either phase of the study. . 63

5.1 In their most basic form, Support Vector Machines (SVMs) find a hyperplane separating the two classes of a linearly separable data set. As this figure intuitively shows, hyperplanes with a wider margin devoid of data points around them, will tend to exhibit better generalization. (© Fabian Buërger / Wikimedia commons / CC-BY-SA-3.0 / GFDL) 70

5.2 General form of Receiver Operating Characteristic (ROC) curves. The closer they reach 100% sensitivity and specificity, corresponding with the $(0, 1)$ coordinate in the figure, the better the models. 76

5.3 Workflow used for obtaining learning curves, graphing Area under the ROC curve (AUC), sensitivity, specificity, and accuracy with regard to sample size. 77

5.4 AUC learning curves obtained from applying logistic regression, Least-Squares Support Vector Machines (LS-SVM) using a linear kernel, and LS-SVM using an Radial Basis Function (RBF) kernel, to the raw International Ovarian Tumour Analysis (IOTA) variables listed in Table 5.1. Thick lines indicate median values, with the region around them showing interquartile range (IQR). 79

5.5 AUC learning curves obtained from logistic regression applied to the preprocessed IOTA variables from Table 5.2, and from LS-SVM using linear and RBF kernels applied to the raw IOTA variables listed in Table 5.1. Thick lines indicate median values, with the region around them showing IQR. 80

5.6 AUC values for several combinations of classifiers and feature sets, for different training set sizes. 81

5.7 The two learning curves in this plot show the AUC attained by LS-SVM with an RBF kernel applied to the raw and preprocessed data sets, respectively. The similarity of the results indicates that LS-SVM with an RBF kernel is sufficiently capable of modelling any non-linearities, so that it does not require any additional preprocessing step. 82

6.1 Typical machine-learning workflow for clinical diagnostic modelling. 86

6.2 Simplified `uml` representation of the classes involved in CDM's internal representation of study data. 88

6.3 The `DataManager` interface provides access to data and enables preprocessing, by including methods for the creation and manipulation of `DataDescriptor` and `Data` objects. 90

6.4 Most preprocessors operate on `DataDescriptor` objects, which do not contain the data themselves, but instead hold `SampleIterable`, `DataPointFilter`, and `SectionInfoMap` objects, which respectively describe which data should be loaded, which should be filtered out, and how they should be presented. 93

6.5 Representation of the relationship between `SectionInfoMap` and the different implementations of the `FieldInfo` interface. For clarity, `FieldInfo`'s template arguments, as well as some methods, are omitted. 94

6.6 The `DataPointFilter` leverages the *Composite* design pattern[28] to enable combining basic filters to more complex filters using "and", "or", and "not" operators. 95

6.7 The `SampleIterable` interface provides a means to iterate over patient entries stored in the database. Its default implementation, `StudySampleIterable`, only loads data upon invocation of its `iterator`() method, using a handle to the database and a study identifier. Using the *Composite* design pattern, `CombinedSampleIterable` accepts a list of `StudySampleIterable` objects, enabling iteration over several different studies at once. 95

6.8 Example of the hierarchical structure of questionnaires. Except for questions at the top of the hierarchy, questions only apply for certain values of their "parent" questions, and will be structurally missing otherwise. (Screenshot of the "Ovaries" section, included in the IETA #1, #3, #4 studies, as presented by the CDM user interface.) 97

6.9 Set of interfaces defined by the CDM software framework for interaction with machine-learning algorithms, and their current only implementation, utilizing the Waikato Environment for Knowledge Analysis (WEKA) library. The machine-learning algorithm used can be chosen by supplying `WekaClassifier`'s constructor with a subclass of WEKA's `Classifier` class. . . 99

6.10 *Facade* interface exposing CDM's machine-learning capabilities. 100

6.11 The classes from this UML diagram provide access to CDM's data analysis Application Programming Interfaces (APIs), from within the *Jython* interpreter. To this end, they leverage Spring's Inversion of Control (IoC) container to provide access to Spring components. 102

6.12 The UML diagrams above list the *Jython* modules providing access to CDM's data analysis facilities. The `dm` module enables the use of data access and data preprocessing capabilities; `ml` provides access to CDM's machine-learning API; while `stats` can be used for calculating some common statistical measures. 103

7.1 Geographical distribution of centres participating in the IETA studies. Centres that contributed fewer than 50 total patient entries are marked in blue, while the others are marked in red. 109

7.2 Evolution of patient inclusions over time, for the different phases of the IETA study. 110

7.3 Learning curves showing AUC with respect to sample size for models predicting endometrial malignancy derived from the data sets of the different IETA studies. Variability of the results is assessed by creating several training-test splits, and generating learning curves for each. The thick lines indicate median values, while the region around these lines represent the interquartile range (IQR) of the results. 117

7.4 AUC with respect to sample size, for models predicting endometrial malignancy, obtained from the merged data of all IETA studies combined. 118

C.1 Learning curves showing evolution of AUC with respect to sample size, for logistic regression and LS-SVM, applied to the variables from the IOTA data set listed in Table 5.2. 136

C.2 AUC learning curves for logistic regression and LS-SVM, applied to different subsets of variables from the IOTA data, obtained by feature selection. 138

C.3 Performance comparison of Stepwise Logistic Regression (SLR) followed by either logistic regression or LS-SVM with a linear kernel, and Automatic Relevance Determination (ARD) followed by LS-SVM with an RBF kernel. 140

List of Tables

2.1 Members of the IETA steering committee. 13

3.1 Story list for CDM's EDC module. 24

3.2 Clinicians selected for participation in CDM survey. Of 42 participants, 28 responded, resulting in a 66.7% response rate. . 44

3.3 Quantitative results from the multiple-choice questions in the CDM survey. 46

3.4 Open-ended questions listed in CDM survey, along with a selection of the most relevant answers. 47

3.5 Source Lines of Code (SLOC) per module. Note that these numbers include the line counts from the APIs which will be elaborated on in Chapter 6. 48

3.6 Test coverage for the different modules, as ratios of lines and branches covered, respectively. Averages are weighted according to the modules' production code sizes from Table 3.5. 48

4.1 List of study participants. 53

4.2 Jackknife estimates of Fleiss' κ coefficient for the different variables pertaining to unenhanced ultrasound, in the first ($\hat{\kappa_1}$) and second ($\hat{\kappa_2}$) phases of the study. Columns σ_{κ_1} and σ_{κ_2} show respective standard errors, while columns n_1 and n_2 are the respective number of subjects available for the calculation of these coefficients. 60

4.3 Jackknife estimates of Fleiss' κ coefficient for the different variables pertaining to sonohysterography, in the first ($\hat{\kappa_1}$) and second ($\hat{\kappa_2}$) phases of the study. Columns σ_{κ_1} and σ_{κ_2} show respective standard errors, while columns n_1 and n_2 are the respective number of subjects available for the calculation of these coefficients. 61

5.1 Input features of the "raw" IOTA data set. 67

5.2 Input features of the "preprocessed" IOTA data set. Derived features, obtained intuitively or by preprocessing, are enclosed in a black box. 68

5.3 Definition of "derived" variables used in the generation of the IOTA model. While the introduction of the first four makes intuitive sense, the latter two require an elaborate, time-consuming analysis, described in Ameye[3]. 68

5.4 Contingency table of actual versus predicted outcome. 75

6.1 Default values used by `DataManager.flatten`() for the different feature types. 92

6.2 The questions from the hierarchical structure depicted in Figure 6.8 have several possible values. Except for the top question, they can be structurally missing, depending on the value of their parent question. Using automated methods for generating dummy variables, this table shows how many such dummy variables would be introduced, respectively in the absence or presence of structural information about the CRF. 97

7.1 List of active participants in the studies organized by the IETA consortium. 108

7.2 Number of inclusions for the different IETA studies, as of April 10, 2014. 110

7.3 Outcome distribution for the different IETA studies. 113

7.4 Distribution of menopausal status for the different IETA studies. 113

7.5 Contingency table tabulating the frequency distribution of menopausal status versus outcome. This table's contents are produced by the script from Textbox 7.3. 115

C.1 Input features of the *[LR2]* data set. Derived features are enclosed in a black box. 137

C.2 Input features of the *[SLR6]* data set. Derived features are enclosed in a black box. 137

C.3 Input features of the *[ARD6]* data set. Derived features are enclosed in a black box. 138

Chapter 1

Introduction

1.1 Clinical diagnostics

Many types of disease can effictively be treated or managed if properly diagnosed. This is especially important for potentially lethal diseases, such as cancer. For these, early diagnosis and treatment have a considerable impact on patient survival. In the U.K. for example, Richards[61] estimates that late diagnosis is responsible for between 5000 and 10000 deaths, yearly, which could be avoided with early diagnosis. The impact of early versus late diagnosis on patient survival can also be observed from comparing endometrial with ovarian cancer: The former is, with an estimated 40100 new cases in the U.S. in 2008, the fourth most common cancer diagnosis in women, while ovarian cancer, with 21650 estimated new diagnoses, is only the eigth most common[40]. By contrast, with an estimated 15520 cancer deaths for 2008, ovarian cancer is the fifth most common cause of cancer deaths in women, while endometrial cancer only ranks eighth, with 7470 deaths[40]. The relatively low number of cancer deaths for endometrial cancer is caused in large part by patients presenting with symptoms indicative of endometrial cancer at an early stage, facilitating their early diagnosis. Ovarian cancer, on the other hand, is more difficult to diagnose timely.

Thus, early diagnosis and treatment can considerably improve patient survival in lethal diseases such as cancer. Early diagnosis can, however, be hampered by several factors. Absence of symptoms during the early stage of some diseases, as in ovarian cancer, is one such factor preventing early diagnosis. The other factor hindering early diagnosis is the invasive nature of most diagnostic procedures.

Indeed, most diagnostic procedures involve invasive procedures, such as the use of surgery for taking biopsies. Due to this invasive nature, patients will only be submitted through such diagnostic procedures in the presence of strong indications of disease, causing diseased patients with weaker indications to be missed. Finally, the cost of diagnostic procedures can be prohibitive, preventing their use in all but the patients with the strongest indications of disease.

1.2 Clinical diagnostic model research

In order to avoid the impediments to early diagnosis, mentioned in the previous section, cheap, non-invasive diagnostic procedures must be pursued. These will lower the barrier for the examination of symptomatic patients with diagnostic tests. Non-invasive procedures also enable the introduction of screening tests. In contrast with diagnostic tests, which are applied to symptomatic patients only, screening tests are applied to an entire population of patients, whether symptomatic or not. Such a population could for example include all patients older than a certain minimum age. Obviously, screening tests should avoid invasive procedures even more than diagnostic tests should,

Diagnostic and screening tests based on variables obtained non-invasively, and at relatively low cost, thus could help to detect disease at an early stage, improving patient prognosis, and, in case of lethal disease, patient survival. For internal diseases, non-invasive variables preclude direct examination of the disease tissue, so the use of indirect approaches must be sought instead. These may include a survey of a patient's demographic data, or features detected from assessment of imaging-based modalities, such as ultrasound images, radiographies, or Magnetic Resonance Imaging (MRI) scans. While many variables obtained non-invasively can be suggestive of disease, they are usually not as unambiguous indicators as the variables obtained directly from relevant tissue. In order to obtain non-invasive diagnostic tests with performance comparable to that of invasive diagnostic procedures, diagnostic models need to be designed combining several non-invasive variables, indicative of disease, which, together, provide sufficient assurances for correct diagnosis.

Clinical diagnostic model research aims to design such diagnostic models. This research is spurred by the availability of sophisticated machine-learning algorithms, resulting in the publication of diagnostic models for many pathologies, including ovarian tumours[70, 71, 72], recurrence of prostatic cancer[43, 32], rheumatoid arthritis[83], epithelial cancer[29], and renal transplant rejection[47]. Their development follows a standard pattern along the lines of Figure 1.1. Broadly, one distinguishes five phases in clinical diagnostic

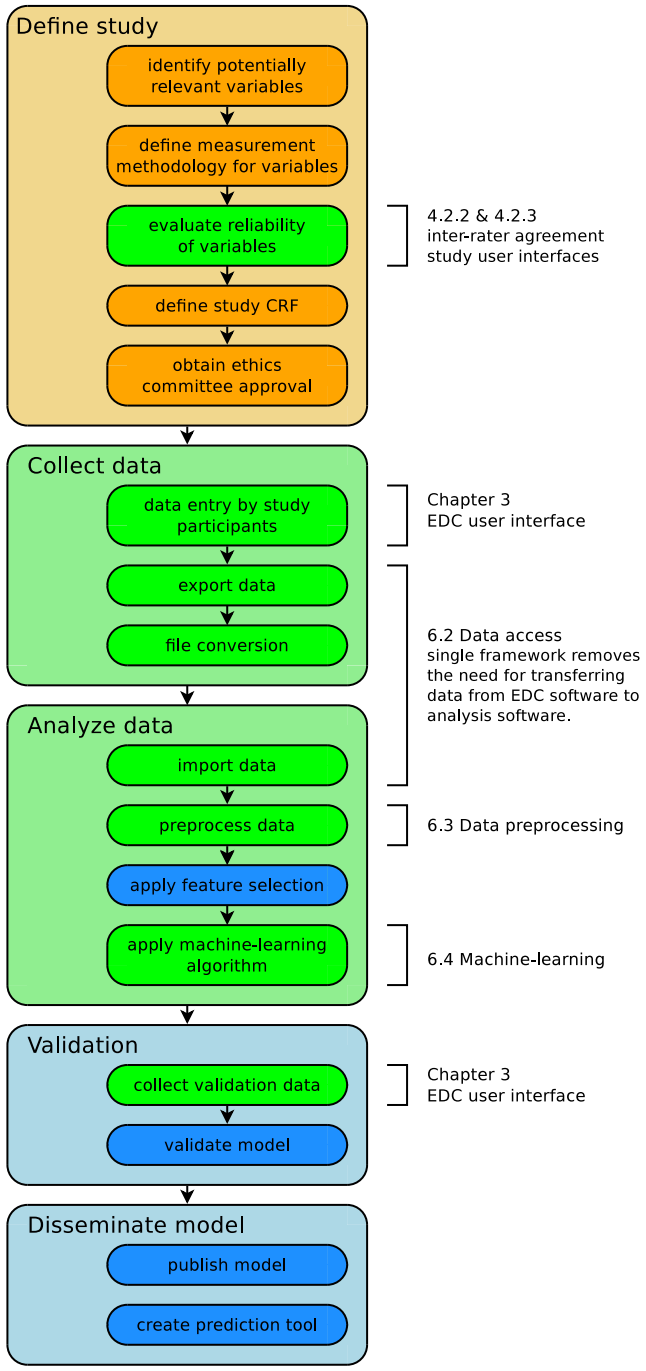


Figure 1.1 – The clinical diagnostic model research workflow. Steps enclosed in green boxes are currently facilitated by the Clinical Data Miner (CDM) software framework, while steps enclosed in blue will be in the future.

model research. First, the study's diagnostic goals are set, and variables that could be relevant for diagnosis are listed. Possible values for these variables are defined. For continuous variables, this requires a measurement methodology, while for categorical variables, the different categories are described. Their reliability can be analyzed with inter-rater agreement studies. The end result of this initial phase is the definition of a Case Report Form (CRF), containing fields for all variables to be investigated in the study. Before starting the next phase, approval from the ethics committee will have to be obtained.

The second phase in clinical diagnostic model research entails data collection. Data are entered by study participants, ideally affiliated to diverse, geographically distributed centres. If sufficient data have been entered, they are collected in a single file, which is converted to a different format if needed, and curated to remove conversion errors, in order to prepare for analysis.

Third, curated data are analyzed. This may involve preprocessing, including data normalization, dealing with missing variables, etc. Since study design in the first phase often introduces many irrelevant variables, in order to avoid missing relevant ones, as well as redundant variables, at this point a feature selection step will attempt to eliminate these, taking into account their relative "costs", objective or subjective. Classification algorithms are then applied to create a diagnostic model.

The fourth phase assesses clinical validity and generalizability of the obtained model. To that end, additional patient data are collected, ideally from centres that did not participate in the initial data collection. These data can then be used to assess validity of the model.

The fifth and final phase aims to valorize the produced model through dissemination. This may include publishing the model in a peer-reviewed journal, as well as the creation of a diagnostic tool that can be used in clinical practice.

1.3 Workflow inefficiencies

Since clinical diagnostic model research has such a large impact on patient prognosis, many more studies can be expected in future, so that it becomes worthwhile to examine and optimize the efficiency of its workflow.

One area that currently involves a relatively laborious, time-consuming process is the preparation of data for analysis, caused by the use of different tools for data collection and data analysis. Depending on the data collection method used, this may require copying data from paper Case Report Forms (CRFs) to

a database, or it may require merging data files from several centres to a single file. This data file may need conversion to a different format, compatible with the analysis tool. Most importantly, it demands careful, manual curation of the data to spot incompatibilities between the file formats used by the data collection and data analysis software.

Another consequence of the use of disparate tools for data collection and analysis is the loss of information about CRF fields and structure. As variables are preprocessed according to their respective types, this loss of information requires data preprocessing either to be performed manually, or to use heuristics for inferring field types. Since the latter use the underlying data to guess variable types, the validity of these guesses will have to be verified manually.

Finally, deciding when to terminate data collection and start data analysis introduces an inefficiency as well: without knowledge about the quality of models that can be obtained from the data, sample size estimates have to be used for assessing when to terminate this data collection. These estimates can be based on rules, or calculated from power analysis. These may both underestimate or overestimate the true sample size required for obtaining an adequate model, leading to inefficiencies in either case.

1.4 Clinical Data Miner

The Clinical Data Miner (CDM) project aims to eliminate the inefficiencies described in the previous section, by integrating data collection and data analysis in a single software framework.

1.4.1 Electronic Data Capture

Development of CDM started with a specific requirement for the data collection software to be used by the International Endometrial Tumour Analysis (IETA) consortium, described in Chapter 2. Since the variables collected by the consortium's different CRFs mostly contain variables obtained from the evaluation of sonographic images, many of them categorical in nature, and since, as part of the IETA consensus paper[45], pictograms had been designed to distinguish the different variables' categories, the consortium wished to integrate these pictograms in the software to be used for data collection. Since no such software existed at the time, I developed CDM's Electronic Data Capture (EDC) software component. Its user interface is demonstrated in Figure 1.2.

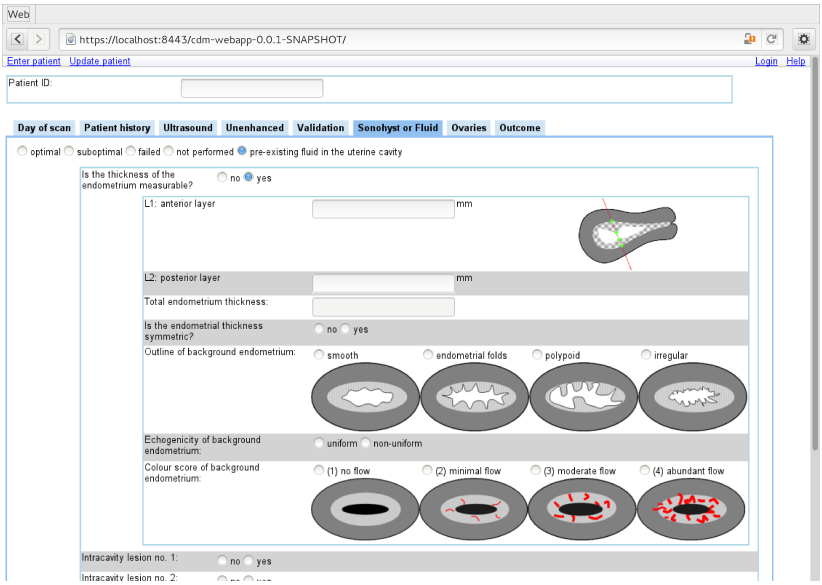


Figure 1.2 – CDM's EDC user interface.

The requirements and process for this development are described in detail in Chapter 3.

While the inter-rater agreement study described in Chapter 4 was unable to conclusively determine if the inclusion of pictograms within a CRF improves inter-rater agreement, a survey about CDM's EDC user interface clearly demonstrated users were very enthusiastic about this feature. Also, this first inter-rater agreement study led to the development of a user interface, derived from CDM's EDC component, that has since served to organize several other inter-rater agreement studies, which are listed in Appendix A.

With its web-based user interface, CDM further simplifies the organization of multi-centre studies.

1.4.2 Data analysis

Integrating data analysis in the same framework as data collection eliminates the inefficiencies listed in Section 1.3. It removes the need for the often laborious curation of data as preparation for analysis. It makes the types of variables, as well as the CRF structure, available during data analysis, obviating the need for manual or heuristics-based preprocessing, reducing errors and improving

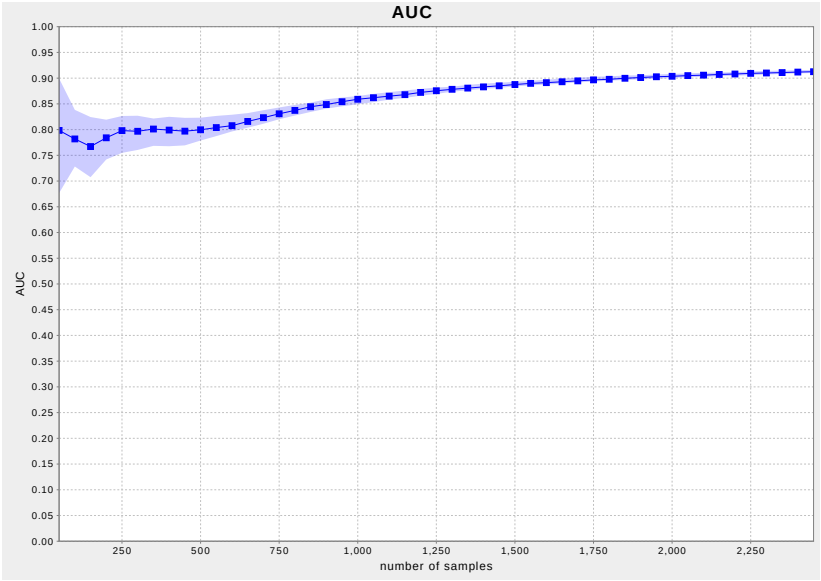


Figure 1.3 – Example of a learning curve generated by CDM's machine-learning functionality.

efficiency. Finally, the integration of preprocessing and machine-learning in the CDM software framework permits the generation of learning curves, such as that of Figure 1.3. These enable straightforward monitoring of study progress, and allow study coordinators to assess when to terminate data collection, avoiding under- or overestimation of required sample size.

Data analysis capabilities that I have currently developed within CDM include *Java* Application Programming Interfaces (APIs) for preprocessing, classification, and calculation of κ-coefficients of inter-rater agreement. Their design is elaborated in Chapter 6. A number of *Jython* modules provide access to these *Java* APIs from within *Jython*, transforming CDM into an interactive experimentation platform for measuring the effect of various preprocessing and machine-learning algorithms on model quality.

1.5 Automating machine-learning

The classification algorithms traditionally used in medicine, such as logistic regression, require complex, time-consuming preprocessing, in order to obtain good models. This hampers further automation of the clinical diagnostic

research workflow. My analysis from Chapter 5 shows that, by using more sophisticated classification algorithms, such as Least-Squares Support Vector Machines (LS-SVM)[67], good models can be achieved without preprocessing.

More analysis is required to verify this conclusion extends to machine-learning workflows which include feature selection. However, this finding encourages further automation of CDM's machine-learning workflow, and its eventual integration in CDM's user interface. This should ultimately empower clinicians to manage most of the clinical diagnostic model research workflow by themselves, diminishing their dependence on machine-learning and Information Technology (IT) expertise.

1.6 Main contributions

The main contributions of this thesis have already been mentioned in the previous section. They include software components for data collection and data analysis, integrated in a single software framework, simplifying diagnostic model research. The EDC component has been in successful use since 2011, by the studies organized by the IETA consortium, with several other studies in the design stage.

A variant of CDM's EDC component considerably simplifies the organization of inter-rater agreement studies. This user interface has been used for six such studies so far.

With the possibility to generate learning curves, CDM's data analysis capabilities facilitate evaluation of a study's progress, enabling study coordinators to assess whether to terminate data collection. With their accessibility from within an interactive *Jython* console, these data analysis capabilities additionally provide an ideal experimentation platform.

I have further shown that sophisticated machine-learning algorithms, applied to raw data, match the performance of more traditional algorithms, such as logistic regression, applied to a data set that has been extensively manually preprocessed. This observation enables increased automation of the clinical diagnostic model research workflow.

1.7 Chapter-by-chapter overview

The present manuscript is organized as indicated in Figure 1.4.

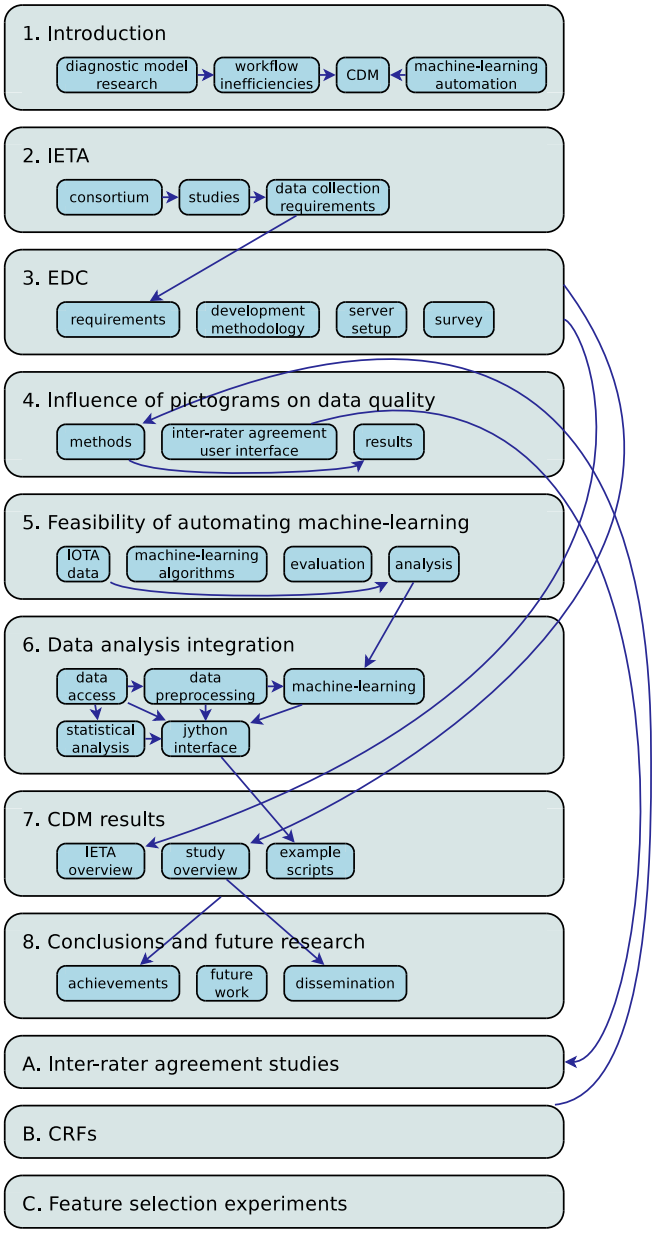


Figure 1.4 – Overview of chapters and their mutual dependence.

Following this chapter, Chapter 2 describes the IETA consortium, which, by requiring the integration of pictograms in its data collection software, initiated development of CDM's EDC component, and delivered its pilot studies. Chapter 3 describes development of this EDC component, including requirements analysis, and development process, as well as results of a survey conducted amongst its users.

Then follows an analysis in Chapter 4, to analyze if the integration of pictograms improves data quality, evaluated by using inter-rater agreement as a proxy measure. This includes a description of the specialized user interface derived from CDM's EDC component, for the organization of inter-rater agreement studies.

Analysis of the learning curves in Chapter 5 shows that sophisticated machine-learning algorithms, such as LS-SVM[67], applied directly to raw data, match the performance of logistic regression after extensive manual preprocessing of this data. This observation allows to considerably simplify automation of the machine-learning workflow, motivating the integration of data preprocessing and machine-learning components in CDM. This resulted in the development of a set of APIs, described in Chapter 6. Future work should integrate these APIs further in CDM's user interface, simplifying management of the machine-learning workflow. In the meantime, these APIs can be used interactively from within a *Jython* console, providing an ideal experimentation platform.

Chapter 7 lists some of the achievements in more detail, including examples of some of the possibilities provided by CDM's experimentation platform. Chapter 8 concludes this manuscript with a summary of achievements, and projections of future work and avenues for dissemination.

Appendix A describes the inter-rater agreement studies that have been organized using CDM's modified EDC user interface, described in Chapter 4, while Appendix B lists the CRFs used for the study described in that same chapter. And finally, Appendix C presents the results of some experiments with some feature selection algorithms.

Chapter 2

International Endometrial Tumour Analysis

The International Endometrial Tumour Analysis (IETA) consortium aims to analyze the relationship between visual features assessed from ultrasound images on the one hand, and endometrial pathology on the other. Requiring patient data for their analysis, the consortium needed an Electronic Data Capture (EDC) software component, which formed the basis of the Clinical Data Miner (CDM) software project.

2.1 Introduction

In order to establish the context in which this project was developed, this chapter introduces the International Endometrial Tumour Analysis (IETA) consortium. This international consortium studies endometrial pathology with the aim to improve its diagnosis.

First, some information about endometrial pathology is provided. Second, the International Endometrial Tumour Analysis (IETA) consortium is described and its consensus paper mentioned. The ultrasound technologies used by the consortium to examine endometrial pathology are explained. Next follows a description of the consortium's currently defined studies. This is followed by a description of the IT infrastructure required for organizing these studies, which formed the starting point for this project. Finally, some concluding remarks are formulated.

2.2 Endometrial findings at histopathology

Endometrial findings at histopathology can be subdivided into benign conditions (atrophy, proliferative endometrium, secretory endometrium, endometrial hyperplasia without atypia, endometrial polyp, and endometritis), precursors of malignancy (atypical hyperplasia), and malignant tumours (endometrioid adenocarcinoma, sarcoma, and other rare malignancies). Of these, endometrial malignancy being life-threatening, it is the primary research focus of the IETA consortium.

For the year 2014, the U.S. expect 52639 new endometrial malignancy diagnoses, and 8590 associated deaths[64]. Both in the U.S. and in Belgium, endometrial malignancy is the fourth most frequently occurring cancer in women[1, 64]. Fortunately, since it is often accompanied by abnormal uterine bleeding, a majority of 68% of cases are diagnosed at an early stage. As a result, it only ranks as seventh most frequent cause of cancer death in the U.S.[64], and fourteenth in Belgium[1]. This clearly shows the importance of accurate diagnostic techniques for endometrial cancer.

Currently, gold standard diagnosis for endometrial cancer is obtained by means of invasive procedures. These include endometrial biopsy, hysteroscopy, and dilation & curettage (D & C). No non-invasive procedures currently exist for endometrial cancer diagnosis. Only a minority of women presenting with abnormal uterine bleeding, however, will eventually be diagnosed as having endometrial cancer. In UZ Leuven, for example, only around 6% of women presenting with abnormal uterine bleeding will prove to have malignant disease. This percentage varies depending on the type of centre: oncological centres will have a higher incidence of cancer than regional centres. In all centres though, a considerable proportion of women needlessly undergo invasive diagnostic procedures, with risk of morbidity or impaired fertility as a result.

2.3 IETA consortium

The IETA consortium is an international consortium of gynaecologists specialized in ultrasound. The members of the IETA steering committee are listed in Table 2.1. They collaborate to research endometrial pathology, in order to improve endometrial pathology diagnostics with models that only rely on variables obtained by non-invasive means.

Variables that may be of interest for such a diagnostic model include demographic features, patient history, and, most importantly, features obtained from

D. Timmerman	T. Bourne	L. Valentin	F.P.G. Leone
E. Epstein	B. Van Calster	B. De Moor	T. Van den Bosch

Table 2.1 – Members of the IETA steering committee.

subjective assessment of ultrasound imaging modalities. These imaging modalities may be obtained through conventional, unenhanced ultrasound imaging modalities, or through Gel Instillation Sonohysterography (GIS). Since the ultrasound variables relevant to endometrial cancer diagnosis are largely unexplored territory, the members of the IETA consortium have enumerated all potentially relevant variables, and, for each, have defined how to measure them, or which categories should be distinguished. Where appropriate, they created pictograms to clarify definitions. They have published this information in the consortium's consensus paper[45], to serve as the basis for studies about endometrial pathology in general, and the consortium's own studies in particular.

2.4 Ultrasound imaging technology

Ultrasound imaging is an imaging technique for the visualization of inner organs and body tissues. It uses sound waves with frequencies above the audible frequency range, produced by a piezoelectic transducer. The waves transmit through the body, and scatter at the interface between different body tissues, as a result of differing acoustic velocities. The transducer registers both the intensity and the delay of the scattered waves. By sweeping an area with this technique, an image can be produced.

Gel Instillation Sonohysterography (GIS) is a variant of this technique, in which a sterile gel is instilled into the uterine cavity, while the latter is being examined with ultrasound technology. This gel improves visualization of the cavity, resulting in more detailed images.

2.5 Studies

The IETA consortium has defined a number of studies, all using the terminology of its consensus paper[45]. They are defined as follows:

- IETA #1 – This study, led by T. Van den Bosch, examines endometrial pathology in female patients who present with abnormal uterine bleeding.

The latter being a potential indication of endometrial cancer, these patients are followed up with invasive diagnostic procedures. These can be endometrial biopsy, hysteroscopy, or hysterectomy. Patients subjected to either of these procedures are included in this study, for the development of a diagnostic model for endometrial cancer. Other endometrial pathology will be modelled as well.

The IETA #1 study is further subdivided into a number of substudies. They are the following:

- IETA #1a –This is the "basic" protocol, containing the most important variables, which should be selected if the clinician has insufficient time for completing the IETA #1b protocol.
- IETA #1b – This protocol expands on the basic protocol with more detailed questions about bleeding pattern and medical history.
- IETA #1c – This substudy was defined when the IETA studies had already started. It should only recruit post-menopausal patients, and it includes additional questions inquiring about patients' lifestyle and anthropometric data.

- IETA #2 – This examined inter-rater agreement for some of the variables defined by the consortium's consensus paper[45]. It involved the evaluation of two sets of 122 video clips, using grayscale and Doppler ultrasound, respectively. The study was led by L. Valentin, and data collection for this study took place between July 10, 2012 and January 21, 2013. A publication of the results will follow.

- IETA #3 – This study, examining endometrial pathology in patients with no abnormal bleeding, is led by F.P.G. Leone. Data are collected for women with indications for hysterectomy, laparoscopy or hysteroscopy. Since the patients are asymptomatic, their data could be used for designing a screening test, which, contrary to diagnostic tests, are applied broadly, without any indication of disease.

- IETA #4 – Led by E. Epstein, it examines the relation between ultrasound findings and patients who have been previously diagnosed with endometrial cancer, and are scheduled for hysterectomy.

Final approval by the UZ Leuven ethics committee board for the organization and coordination of the multi-centric IETA studies was granted on April 19, 2011. Study participants from other centres are responsible for obtaining approval from their respective centres' ethics committees.

Additional studies examining the predictive value of endometrial morphology, described using the IETA terminology, on fertility and on pregnancies of unknown location (PULs), are planned.

2.6 Data collection

To enable the creation of diagnostic models, these studies naturally require data. In order for these models to be internationally valid, the data need to be collected from different locations around the world. This is facilitated by involving multiple centres from geographically diverse locations. To support multi-centric data collection, participants require a user-friendly, efficient means of collecting data. Web-based EDC software thus constituted a logical choice.

D. Timmerman and T. Van den Bosch wished to integrate the pictograms, published in the IETA consensus paper, into the EDC software to be used for the IETA studies, to guide clinicians during data collection. As this did not exist at the time, D. Timmerman requested me to develop such software.

2.7 Conclusion

In 2009, a group of gynaecologists specialized in ultrasound formed the IETA consortium, with the aim to examine endometrial pathology in general, and endometrial cancer in particular. To that end, they organized a number of studies, each targeting different patient populations. Collecting data from multiple centres in an efficient manner required the use of EDC software. Since most of the variables collected are assessed from the interpretation of ultrasound imaging modalities, the ability to show pictograms, such as those published in the consortium's consensus paper[45], was submitted as one of its requirements. The request to create such software formed the basis of my involvement in the IETA project.

Chapter 3

Electronic Data Capture

The previous chapter established the need for Electronic Data Capture (EDC) software, for conducting the International Endometrial Tumour Analysis (IETA) studies, providing the ability to integrate pictograms. In this chapter, I describe the requirements and methodology that I used for the development of Clinical Data Miner (CDM)'s EDC component.

3.1 Introduction

Patient data form the basis of Evidence-Based Medicine. While in the past, patient data were collected using paper Case Report Forms (CRFs), the use of EDC software for collecting data has steadily been gaining ground in the past decades[4, 13]. El Emam et al.[21] estimate that, between 2006 and 2007, 41% of Canadian clinical trials were using an EDC system. EDC adoption in clinical studies is spurred by the broad availability of generic software such as *REDCap*[34], or the open-source *OpenClinica*®[55] on the one hand, and EDC's important advantages over Paper-based Data Collection (PDC) on the other.

One of these advantages are efficiency gains[48], resulting in lower study costs[57]. Indeed, without the need for copying paper CRFs to a spreadsheet, collecting data using EDC is less resource-intensive. Simulations by Pavlović et al.[57] estimate the cost savings of EDC with respect to PDC to be between 49% and 62%, depending on study parameters. One of the most important of these parameters is study size: the larger the patient cohort, the larger the expected savings.

Two factors contribute to another important advantage of EDC: both the elimination of a copying step, as well as the possibility to integrate automatic validation of data, lead to reduced data errors. In a study by Walther et al.[88], for example, data error rate for EDC *without* double data entry, but with automatic completeness checks, was similar to that of PDC, *with* double data entry.

Many recently developed EDC systems are web-based. If these are developed to support a wide array of web browsers, this reduces infrastructure costs even further, as it obviates the need for costly, time-consuming installation and upgrades on individual study participants' computers, and instead only requires the central management of a single system. This is especially compelling for multi-centre studies, which may include centres with widely disparate IT infrastructure, as is the case for the IETA studies.

Presented in chapter 2, these IETA studies aim to develop models for the diagnosis of endometrial tumours, based on, amongst other, demographic data, but primarily on characteristics observed from subjective assessment of ultrasound images. In order to optimally research endometrial pathology, the IETA consortium defined terminology and measurements to be used for the characterization of endometrial features in their consensus paper[45]. Since most of these terms describe visual features of ultrasound images, it is reasonable to assume that reference pictograms, visualizing a variable's possible categories, could aid study participants in selecting the appropriate category more accurately. Thus, T. Van den Bosch developed pictograms to clarify the terms described in Leone et al.[45], while D. Timmerman requested me to develop an EDC platform allowing the integration of pictograms in CRFs.

This chapter describes how this platform was developed. A first subsection describes my attempt at integrating pictograms in the user interface of existing software. As this yielded unsatisfying results, I then started development on the new Clinical Data Miner (CDM) EDC system. The second subsection lists the results of the requirements analysis we performed for this project. Next follow descriptions of the software development methodology, the architecture of the system, and server setup. Section 3.7 discusses some results, including some source code metrics. Finally, a number of conclusions complete this chapter.

3.2 Existing software

Since the creation of a new EDC system would mean a substantial development effort, I initially investigated the possibility of extending the user interface of existing software packages. This necessitated the availability of source code.

OpenClinica®[55] being the most widely known open-source EDC system, it appeared to be the best starting point.

OpenClinica® claims to be the world's first commercial open-source EDC system. Both community and enterprise editions of *OpenClinica*® exist. Source code to its community edition is freely available; the enterprise edition contains additional features and is commercially supported. Thanks to the open-source nature of the community edition, it has gained wide adoption in clinical research, resulting in over 15000 registered users since its initial release nine years ago. The availability of source code further enables the development of third party functionality, making it suitable for implementing the extension required for the IETA studies.

Adding the desired functionality to *OpenClinica*® involved modifications in two locations. Changes to the user interface code, on one hand, would enable displaying pictograms alongside fields in the CRF. These pictograms could be supplied as separate image files in JPEG or PNG format. On the other hand, the parsing code would have to be extended to allow the interpretation of additional columns in the CRF definition files. These additional columns would describe which pictograms should be displayed, and for which variable choices.

I focused on the latter. All of the parsing code proved to be contained in a single file. This file was poorly designed, consisting of several 1000's of lines of code, and including a high number of branches. Moreover, there were virtually no automated tests, for verifying correctness of the program. Combined, the parsing code's poor structure, and the near absence of automated tests considerably complicated the implementation of extensions to the parsing code, as any modification would have to be verified through extensive, time-consuming manual testing.

Therefore, I abandoned the idea of extending *OpenClinica*® in favour of creating a new EDC software project. In doing this, one of the aims was to provide a better user experience than *OpenClinica*® does, with a user interface requiring fewer static page transitions, leveraging a more modern Asynchronous Javascript and eXtensible Markup Language (XML) (AJAX) approach. Compared with the extension of an existing software project, the creation of a new software project provides the additional benefit of simplifying the implementation of one of this thesis project's key goals, namely the integration of data analysis functionality, further described in Chapter 6.

3.3 Requirements

A requirements analysis was the initial step of the CDM project, for which D. Timmerman and T. Van den Bosch provided invaluable input. This obviously included the ability to integrate pictograms into the user interface. I further opted to implement a web-based user interface, to avoid requiring software installation on end users' computers (beyond perhaps the installation of a sufficiently recent web browser), which brings an important advantage to multi-centre studies. As an additional requirement, the design had to enable a potential future conversion to a desktop or smartphone application. For the definition of CRF questionnaires, I aimed for compatibility with *OpenClinica*®'s CRF definition spreadsheet format. While not entirely compatible, this should facilitate migration from *OpenClinica* to CDM.

Using an iterative scrum software development approach[63], the project was organized around fixed time intervals – *sprints* in scrum terminology – of two weeks initially, lowered to one week towards the delivery of the EDC component. Requirements were defined, prioritized, and scheduled during sprint planning meetings.

The list of requirements, or *stories* in scrum terminology, were tracked along with their sequence number, the dates when they were registered, and when their implementation started and ended. This list allowed to estimate the amount of stories processed per sprint, defined as the *velocity* of the project. Multiplied by the number of unfinished stories, this velocity allowed to estimate the number of sprints required to finish the project, which enabled progress reporting to the IETA consortium.

Table 3.1 lists sequence numbers, end dates, and descriptions of all identified requirements. Some of these will be elaborated further in this section.

Nr.	Finished	Task
#1	2010/03/03	Extensible architecture for visual elements
#2	2010/03/03	Patient identification form
#3	2010/03/03	Checkbox implementation
#5	2010/03/09	Prototype with dependent entries
#4	2010/03/09	Multiple choice questions with child questions dependent on state of parent question
#35	2010/03/05	Decouple from *OpenClinica*®
#36	2010/03/05	Add build target for code coverage report
#8	2010/03/26	Hierarchical object structure for question entry widgets

Nr.	Finished	Task
#34	2010/03/26	Serializable hierarchical object structure for items
#38	2010/04/01	Web service for querying `RootItems`
#9	2010/04/01	Factory to convert `CaseReportForm` into `EntryForm` objects (requires: #8)
#18	2010/04/01	Interface to commit patient data to database
#22	2010/04/01	Provide a way to select the study sheet to open
#39	2010/04/02	Change `EntryForm.process()` to take a parameter `Continuation` + implement a `SafeContinuation` that obsoletes `SafeEntryForm`
#41	2010/04/08	Implement patient database
#42	2010/04/08	Upon `PatientDbService.store()`, add to patient database (requires: #41)
#43	2010/04/13	Implement crf database
#44	2010/04/13	Upon `StudyService.store()`, add sample to crf database (requires: #43)
#17	2010/04/13	Collect data results from EntryForm objects and add to database (requires: #9, #18)
#51	2010/04/15	Split codebase in several parts: `common`, `server`, `client`, `gwt`
#7	2010/04/30	Layout data collection page (CSS rules)
#20	2010/05/05	Manage different case report forms at the same time
#21	2010/05/05	Way to compose different case report forms from studies
#46	2010/05/05	Refactor messaging code: • create `Message` class that verifies # of objects passed • move `Enum<?>` out of `GeneralException` • make a `*Messages` class per package containing an `Enum<*Messages>` and methods that allow to easily create the messages
#19	2010/05/10	Convert excel sheet/xml file into `CaseReportForm` objects
#53	2010/05/09	Ensure that children of a `ParentItem`, `ChoiceItem`, `SectionListItem`, `Section` can't have the same identifier
#63	2010/05/10	Create a `SafeCrfFactory` instance per `Crf`: it should be allowed to have different items with same id in different `Crf`s. (requires: #53)
#57	2010/05/18	New implementation of `GwtDateEntryNode`

Nr.	Finished	Task
#6	2010/05/22	Multiple choice questions with images
#66	2010/05/22	Prepare for putting application online
#58	2010/05/28	`FormNode` that displays a header/subheader
#68	2010/06/29	Render subsequent fields in alternating colours.
#69	2010/06/29	Fix tabpanel size.
#67	2010/07/01	Choice field with image (i.e. not for the individual choices, but for the choice itself.)
#70	2010/07/01	Choice fields with same captions are lumped together in a single html radio group; fix.
#37	2010/07/04	Increase test coverage.
#24	2010/07/15	Allow registration of constraint validators on `EntryForm` objects (validation should be done on client as well as on server)
#55	2010/07/15	Install continuous integration server
#72	2010/07/27	Save data per tab pane
#12	2010/08/17	Login page
#13	2010/08/17	`CapabilityManager` - interface to determine if a user account has certain rights.
#14	2010/08/18	Use `CapabilityManager` in data collection, account creation, study creation. Profiles: administrator, study manager, data entry
#11	2010/08/18	Implement user account code
#81	2010/08/20	*[BUG – reported by: Arnaud]* `ChoiceEntryNode` doesn't display errors.
#23	2010/08/24	Parse validation constraints from excel sheet
#25	2010/08/24	Construct constraint validators from output of validation constraint parser (requires: #23)
#26	2010/08/24	Register constraint validators (requires: #24, #25)
#86	2010/09/02	*[BUG – reported by: Thierry]* Webapp shuts down because aulne tomcat doesn't read `applicationContext*.xml` from libraries in classpath.
#82	2010/09/27	*[BUG – reported by: Arnaud]* When committing tab pane takes a long time, user clicks to another tab before committed tab is deactivated, newly selected tab may become deactivated instead.
#29	2010/09/10	Calculations on <integer>/<float> entry fields.
#80	2010/09/20	Uniform logging (backend: `slf4j`)
#62	2010/09/10	Add versioning to `Crf` database record (filled with number 1 for now)

Nr.	Finished	Task
#30	2010/09/18	Node object (UI element) that shows the result of a calculation
#45	2010/09/18	Support "calculations", i.e. dependent fields that are filled in automatically when the field it depends on is filled in (see also: #32)
#79	2010/09/20	Implement comment boxes
#75	2010/09/20	Horizontal choices
#52	2010/09/21	Menu on top of pages to easily switch between `PageStates` – menu should be dependent on current `PageState`
#73	2010/10/03	Introduce concept of a `Study` as a combination of a number of `Crf`s
#78	2010/10/04	Simplify patient info entry
#88	2010/10/08	*[BUG – reported by: Lil]* Pages with calculations fail validation: value of calculated items are null
#92	2010/10/09	Human-readable error messages
#91	2010/10/09	*[BUG – reported by: Arnaud]* 1. Select Sonohyst → optimal → measurable: yes ⇒ subpanel for measurable appears 2. Select Sonohyst → suboptimal ⇒ subpanel for measurable disappears
#59	2010/11/26	Page where patient can be selected, and samples taken for that patient for different dates are shown, where one can be selected.
#49	2010/11/29	Auto-completion for examiner, ultrasound system
#60	2010/12/07	Page where patient and sample are selected, and where user can select the additional CRFs to complete.
#61	2010/12/07	Page where patient sample can be completed with additional CRFs. (Can this reuse the current data collection page?)
#76	2010/12/07	Spring-ify server code
#94	2010/12/17	*[BUG – reported by: Arnaud]* On pages with both vertical & horizontal choices, images displayed for the vertical options are aligned right of the images of the horizontal choices. This results in very wide pages. Solution: horizontal choices should span 2nd & 3rd columns. Modify `Grid` such that it doesn't allow variadic arguments

Nr.	Finished	Task
#27	2010/12/09	Use encrypted communication
#87	2011/01/06	*[BUG – reported by: Arnaud]* When logging in as a user who doesn't have the rights required to some functionality, `AccessDeniedException` is shown as a `Window.alert()` instead of showing the message: "User doesn't have required priviledges. Please login as a different user".

Table 3.1 – Story list for CDM's EDC module.

3.3.1 Visual user interface elements

(Refer to stories #6 and #67 from Table 3.1.)

The evaluation of imaging modalities for classification according to certain categorical variables can be simplified by illustrating the distinctions between categories with pictograms. This is especially the case when participants are still familiarizing themselves with a study's terminology. Since the IETA studies require classification of many variables based on ultrasound images, the IETA consensus paper[45] includes many such pictograms. A common understanding of the various categories involved could be further improved by integrating pictograms in the associated CRFs. While this is straightforward for paper-based CRF, it is not commonly possible in electronics CRFs. Therefore, the EDC system used in the context of the IETA studies was required to enable such integration in its user interface.

CDM therefore enables two uses of pictograms. The first allows clarifying questions in a CRF with a pictogram, as illustrated in Figure 3.1. The second use involves the clarification of the distinct options of categorical variables. An example of the latter is shown in Figure 3.2.

Another visual element that generally is more readily added to PDC than in EDC, are Visual Analogue Scales (VASs). These are line segments, with minimum and maximum values indicated on either side, allowing the evaluation of a variable by positioning a mark along the line segment, thereby indicating its relative value with respect to the line segment's bounds. Visual Analogue Scales (VASs) measurements produce continuous values within a certain range. A VAS is typically used for assessing the level of pain experienced by a patient. Due to their continuous nature, measurement by a VAS can be more exact than



Figure 3.1 – The use of pictograms to clarify questions in a CRF. (Screenshot of the "Myometrium" section, included in the IETA #5 study, as presented by the CDM user interface.)

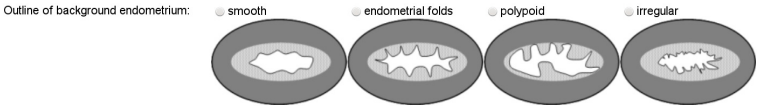


Figure 3.2 – CDM allows to visualize the distinct choices of categorical variables using pictograms. (Screenshot of the "Sonohysterography or Fluid" section, included in the IETA #1, #2, and #3 studies, as presented by the CDM user interface.)

measurement by a Likert scale[27], which discretize the possible values to a number of categories, and requires fewer explanations.

Despite their advantages as measurement devices, they are seldom used: adding them to paper-based CRFs requires measuring the distances of the marks to the line segment bounds, for each VAS individually, which can be cumbersome if there are many. Integrating them in an electronic CRF, however, removes this obstacle, while retaining their advantages. Since some of the IETA studies include a VAS, the possibility to integrate such elements into a CRF was propounded as one of the requirements for the development of CDM. An example of such functionality is depicted in Figure 3.3.



Figure 3.3 – CDM uses a "slider" element to represent Visual Analogue Scales (VASs). (Screenshot of the "Validation" section, included in the IETA #1, #3, and #4 studies, as presented by the CDM user interface.)

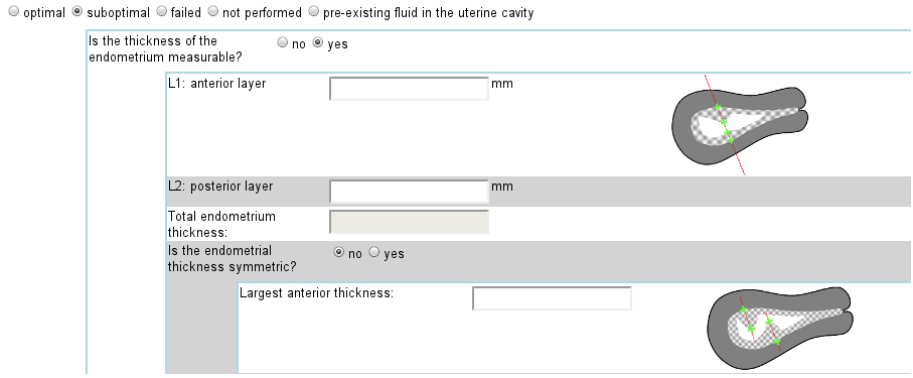


Figure 3.4 – CDM implements the "skip pattern" by showing or hiding variables in a box positioned below their "parent" variable, depending on the value of that "parent" variable. (Screenshot of the "Sonohysterography or fluid" section, included in the IETA #1 and #3 studies, as presented by the CDM user interface.)

Finally, the "skip pattern" is visually presented as a hierarchical tree structure. The "skip pattern" is commonly applied in data collection, to skip variables that become irrelevant due to the value filled in for another variable. As an example, the variable "Years past menopause" becomes irrelevant when "menopausal status" has the value "pre-menopausal". In CDM, variables that depend in this manner on the value of a "parent" variable are only shown if they become relevant, and are framed in a blue box positioned below the parent variable, as in Figure 3.4.

3.3.2 Case Report Forms

CRF representation

(Refer to stories #1, #4, #8, #34, #20, and #21 from Table 3.1.)

CRF questionnaires are composed of several sections, shown in CDM's user interface as different tabpages, and which themselves are composed of several fields of different types. Sections and fields are represented by interfaces included in the class hierarchy derived from the `Element` interface, shown in Figure 3.5.

In this class hierarchy, `SimpleItem` objects represent simple fields containing a value such as a number, a string, or a date. Most of these are shown as text boxes in CDM's user interface, except `DiscreteStepItem`, which is visualized by a slider. `ChoiceItem`s represent categorical variables, corresponding to multiple-

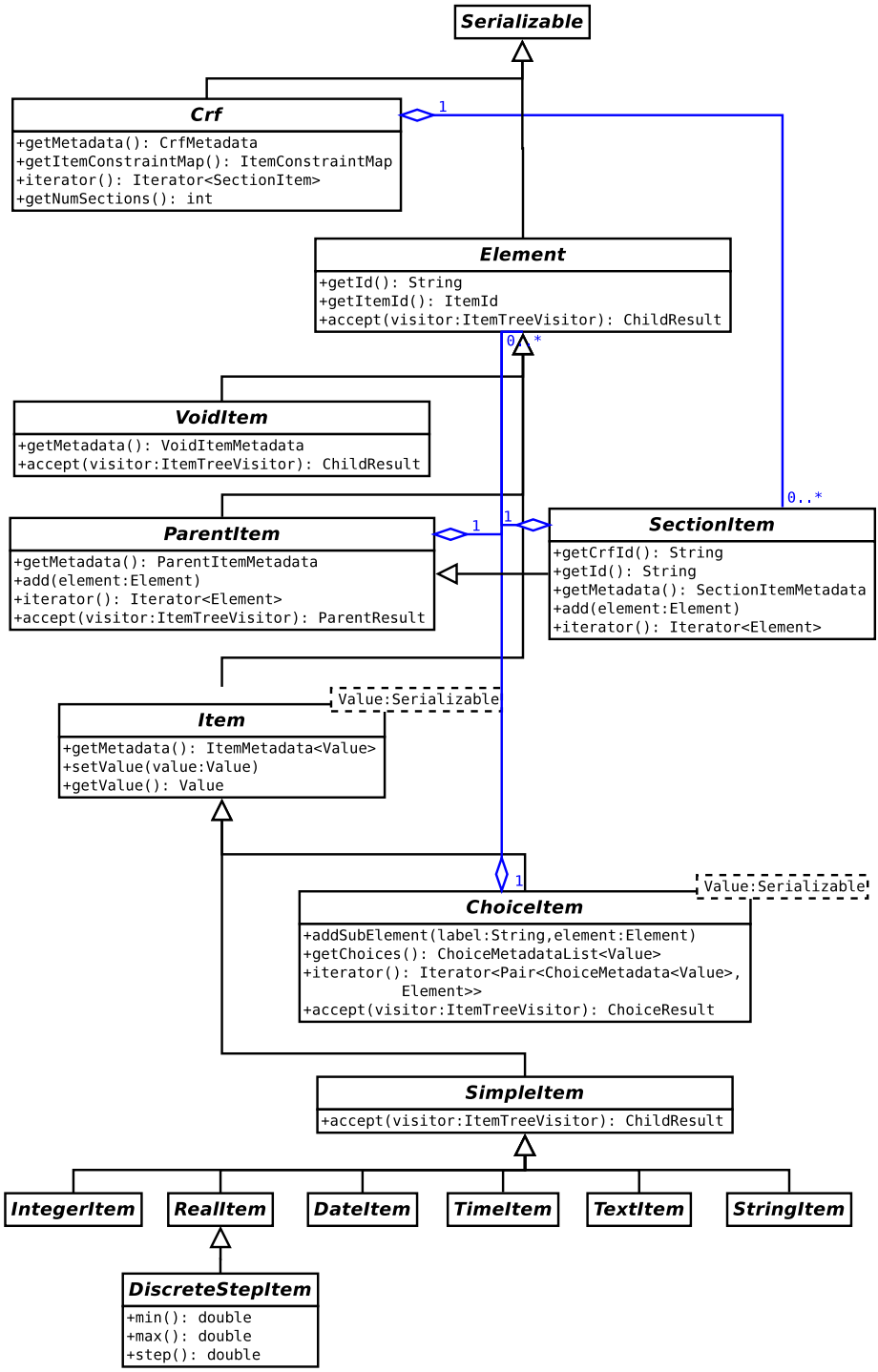


Figure 3.5 – This shows the `Element` class hierarchy for representing sections, fields, and multiple-choice questions of a questionnaire, as well as the `Crf` class containing one or more `SectionItems`.

choice questions. Elements can be added to a category of such `ChoiceItem` objects, which will then only be activated if that particular category is selected. Finally, the sections mentioned above correspond to `SectionItem` objects, which act as containers for other elements, such as `IntegerItem` or `ChoiceItem` objects.

All interfaces in the `Element` class hierarchy, as well as the `Crf` interface, are serializable, facilitating the implementation of the client-server Remote Procedure Call (RPC) protocol discussed in Subsection 3.3.3.

CRF processing

(Refer to stories #9 and #17 from Table 3.1.)

Several use cases require the processing of the `Element` objects contained in a `SectionItem`. This occurs for example when user interface widgets need to be constructed for these elements, in order to assemble a tab page. As another example, storing the contents of a CRF to database requires collecting the values of all elements contained in the CRF's `SectionItem`s. In both these cases, each element type should be processed differently. To enable this, I implemented a variant on the *Visitor* design pattern[28], pictured in Figure 3.6. Compared to the default *Visitor* pattern, this implementation allows returning a processing result, the type of which depends both on the implementing `ItemTreeVisitor` class as well as the type of element processed.

Since a section's elements have a tree structure, they may be processed in several different orders, and inactive branches may be taken into account or not. The `ItemTreeTraversalStrategy` exhibits the *Strategy* design pattern to provide different strategies for ordering element processing. Currently, two implementations exist: `DepthFirstItemTreeTraversalStrategy` allows a depth-first traversal of the entire tree, while `ActiveItemTreeTraversalStrategy` only processes active items, also in a depth-first order.

3.3.3 Remote Procedure Calls (RPCs)

(Refer to stories #34, #38, #44, #72 from Table 3.1.)

The RPC protocol used between client and server is represented in Figure 3.7. It allows clients to query the server for a list of available studies, or the structure of a particular study. Also, clients can request the creation of a new patient entry. They can request the server to add data to, or return the contents of, an existing patient entry. Finally, clients can obtain information about patient

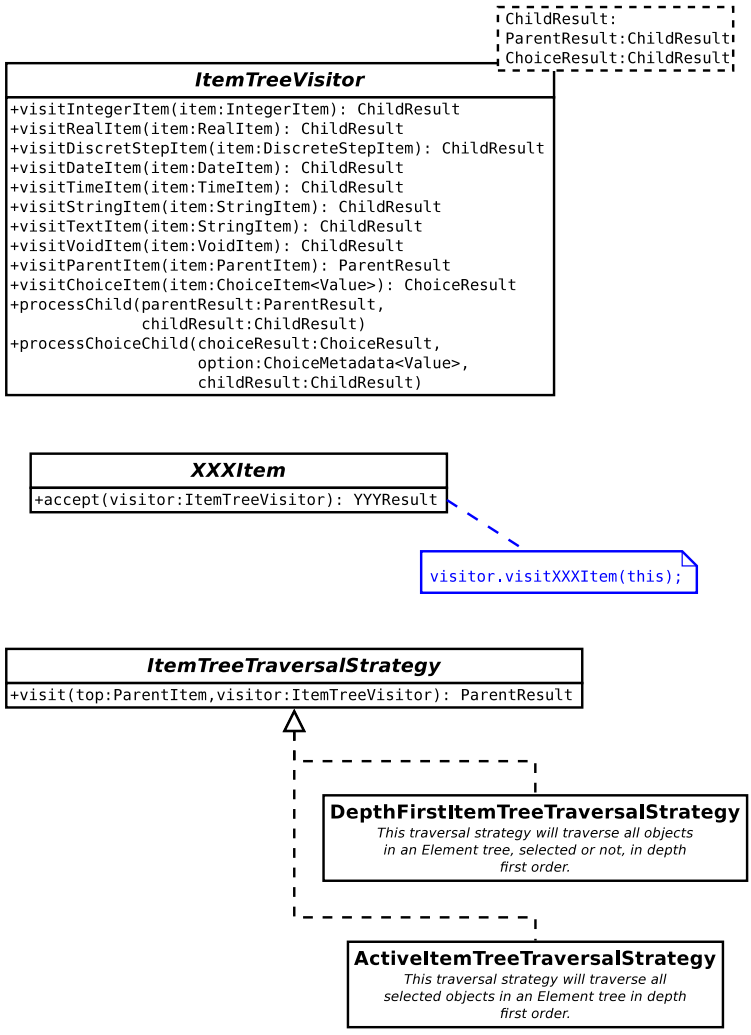


Figure 3.6 – These interfaces are used for processing `Crf` objects. Processing order is determined by the choice of `ItemTreeTraversalStrategy` implementation.

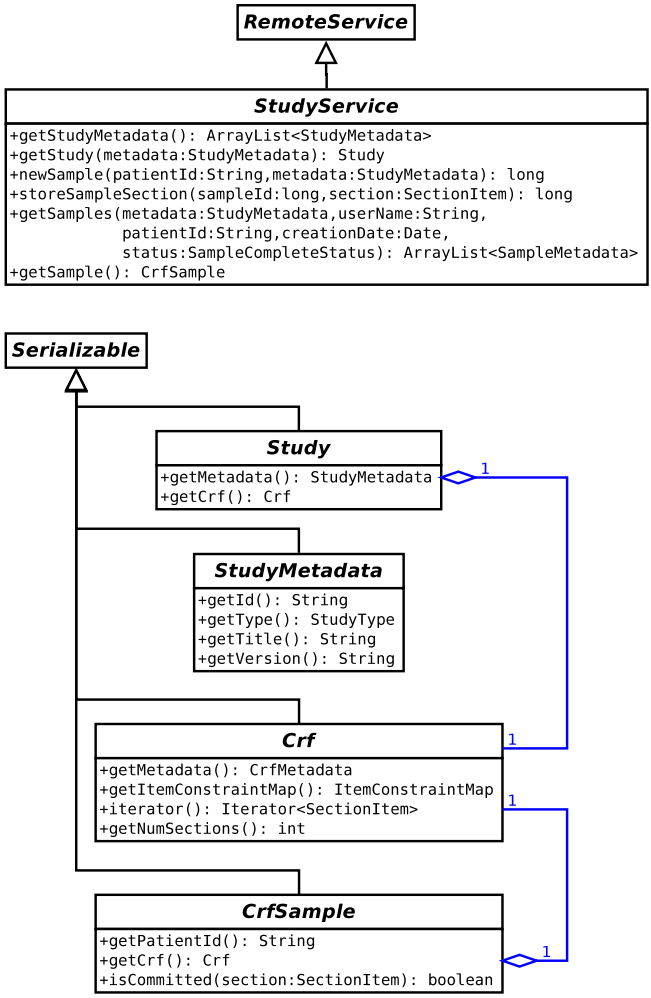


Figure 3.7 – The RPC protocol defined between client and server, for supporting CDM's EDC functionality.

entries stored on the server. The latter will be restricted to entries the clients have access to. Call parameters can restrict the results to a specific study, user, patient, creation date, or completeness status, or a combination thereof.

By extending `StudyService` from Google Window Toolkit (GWT)'s `RemoteService` interface, the GWT compiler will automatically generate source code implementing a communication protocol enabling the RPC calls of the `StudyService` interface, serializing and deserializing parameters and results before transmission and after receipt.

3.3.4 Authentication and access control

(Refer to stories #12, #13, #14, #11, #27, #87 from Table 3.1.)

I used the *Spring Security* framework for handling authentication and authorization. The general configuration of this framework was performed through configuration files. This included setup of the authentication mechanism and annotation-based access control. The latter allowed to control access to specific server methods by means of *Java* annotations.

Java annotations allow to add metadata in a preamble to syntactic elements such as classes, methods, variables, etc. The *Spring Security* framework enables the use of annotations to control access to methods.

The *Spring Security* framework allows to move access control out of method implementations, into configuration files or annotations. This relieves methods of an additional responsibility, reducing complexity and improving design. Compared with file-based configuration, annotation-based configuration further offers the advantage of robustness against refactoring, by keeping access control configuration in the vicinity of the methods under control.

Annotation-based access control can occur both prior to entering critical server methods, granting or disallowing access to certain methods, as well as after a method has returned, to filter out results the user is not allowed access to. *Spring Security* provides standard annotation expressions for common access control patterns, such as limiting access to users of a certain group. It additionally, however, provides an extensible mechanism, using a `PermissionEvaluator` implementation provided by the application, to define project-specific access control mechanisms. Three parameters are supplied to this custom class upon which access control decisions can be based: an `Authentication` object identifying the user, the object to which access should be controlled, and the type of permission requested. Leveraging this mechanism, I designed the pluggable design presented in Figure 3.8.

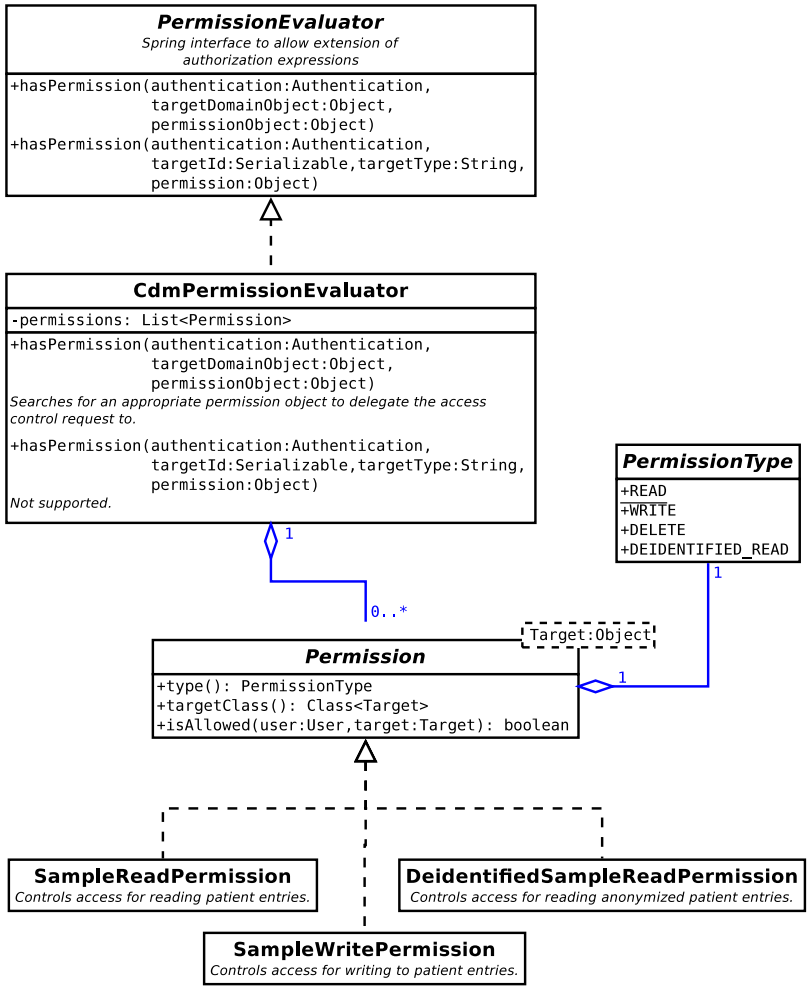


Figure 3.8 – CDM's pluggable access control is governed by means of `Permission` objects.

In the CDM framework, permission types are categorized as read, write, and delete operations, as well as reading of anonymized data. The latter will be used in the future to provide study coordinators access to data included in their study by participants from a different hospital and/or department. Each class implementing the `Permission` interface handles a specific combination of permission and object types. Using *Spring*'s Dependency Injection (DI) mechanism, each `Permission` subclass is instantiated, and injected into `CdmPermissionEvaluator`'s constructor. The latter handles permission requests on behalf of the *Spring Security* framework, by delegating them to the appropriate `Permission` object. This flexible design enables the use of *Spring Security* annotations such as listed in Textbox 3.1. For the moment, these are used to restrict read and write access to patient entries created by users from the same department.

```
// Prevents returning a sample the user is not authorized
// to read.
@PostAuthorize("hasPermission(returnObject, 'read')")
public Sample load(final long sampleId) {

[...]

// Only call the method if the user is authorized to write
// to the supplied sample parameter.
@PreAuthorize("hasPermission(#sample, 'write')")
public void store(@NotNull final Sample sample) {

[...]

// Filter out the samples from the returned collection of
// samples that the user is not authorized to read, and
// return the remaining objects.
@PostFilter("hasPermission(filterObject, 'read')")
public List<Sample> loadSamples(final LoadPolicy loadPolicy) {

[...]
```

Textbox 3.1 – Examples of the *Spring Security* annotations that can be handled by the design from Figure 3.8.

3.3.5 Database structure

(Refer to stories #43, #44, and #62 from Table 3.1.)

Using the *Hibernate* Object-Relational Mapping (ORM) framework provides an abstraction layer mapping CDM's object-oriented domain model to a traditional

Structured Query Language (SQL) database. Hence, this layer decouples CDM from the actual database system used, whether it be MySQL®, PostgreSQL, or Microsoft® SQL Server.

In order to support generic studies, rather than creating a database table per study, in which columns represent fields from the study, I created the database schema displayed in Figure 3.9. It contains the tables `samples`, `sampleSections`, and `sampledata`, corresponding to patient entries, sections of patient entries, and fields in a patient entry, respectively. Tables with a grey background in Figure 3.9 establish 1:n relationships between these tables, linking a set of sections to a patient entry, and a set of fields to a section. This database schema enables recording of patient entries for different studies by including a study identification field in the `samples` table.

The database further stores user information, using tables `centres`, corresponding to hospital centres, `realms`, representing hospital departments, and finally `users`, containing information about study participants. While a primary realm is associated to each user, users can be a member of multiple realms. These latter associations are encoded in the `realms_users` table. This grouping of users in realms supports CDM's access control, by providing users only access to entries from the `samples` table which have a `realm_id` of a department they are affiliated with.

3.4 Software development methodology

In this section, I discuss the software development process used in developing CDM. A first subsection explains the choice of programming language and frameworks. Next follows a discussion of the quality assurance process. The section ends with a description of the software configuration management applied to the CDM project.

3.4.1 Programming language & frameworks

In order to simplify deployment across multiple centres, I chose to develop the primary user interface of CDM's EDC module as a web-based application. Desktop applications require installation and regular upgrading of software on individual participants' computers, leading to a relatively high IT cost and barrier to entry. By contrast, web-based applications only require administration of a single system. Participants only need to point a compatible web browser to the web application's Uniform Resource Locator (URL).

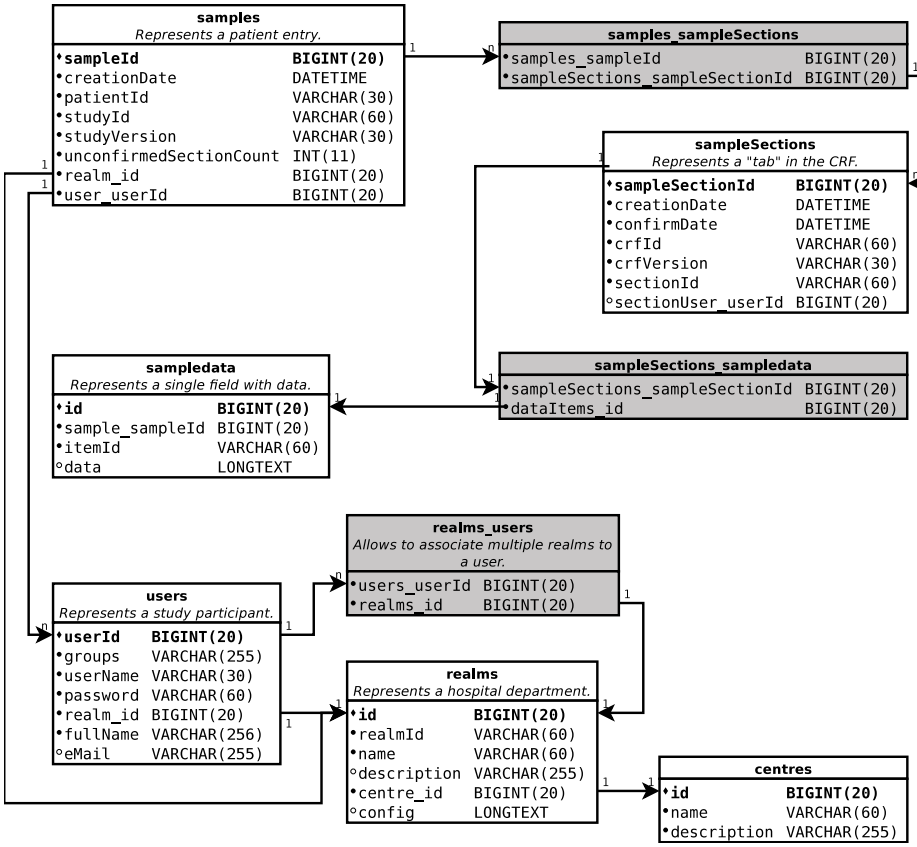


Figure 3.9 – CDM's database schema. Tables corresponding with business classes have a white background. Tables with a grey background are responsible for establishing a 1:n relationship between business classes.

Modern web applications, however, not only run server code on a centrally located web server, but also run client code on end users' web browsers. The aim of software running on the latter is to improve interactivity and response times. While web servers can be programmed in a variety of languages, such as *PHP*, *python*, *perl*, or *Java*[31], client code is programmed in *ECMAScript*, which is universally recognized by current web browsers. Using different programming languages for client and server, however, would complicate the software development process, maintenance, and testing. Software development would require different tool sets for different languages; code refactorings would not propagate changes to source code written in a different language; and verification of the software would require testing at a considerably higher level,

causing test development to become both more complex and less effective. More importantly, by using *ECMAScript* for client code, its usage would be restricted to web applications, preventing reuse in desktop applications.

The Google Window Toolkit (GWT)[30] provides a solution, allowing the use of the *Java* programming language for both client and server code. GWT combines a user interface widget library, for programming user interface elements, as well as a compiler transforming *Java* client code to *ECMAScript*, enabling client-side execution. Adopting GWT for developing CDM's EDC component allowed to write the entire source code in a single language, namely *Java*. By further separating the client code into presentation and business logic layers, as described in Section 3.5, the latter could potentially be reused for future user interfaces, such as desktop or mobile applications.

Apart from the GWT toolkit, CDM makes use of Inversion of Control (IoC) containers[24]. IoC is a programming technique in which objects are not coupled to specific implementations of other objects during compile time, but to the *interfaces* of those objects. At runtime, specific implementations for these interfaces are selected by an IoC container, which can be configured using XML files or annotations. Such a setup offers flexibility advantages, because the specific implementation of an interface can be changed by reconfiguring the IoC container. More importantly, it enables loose coupling between objects, thereby improving software design. The loose coupling of objects in turn facilitates unit testing, which will be discussed in the next section.

Initially, only CDM's server code made use of such an IoC container, provided by the *Spring* framework, because integration of the GWT and *Spring* frameworks is relatively difficult. Later, I modified the client code to use the *Gin & Guice* frameworks, which are intended specifically for projects using the GWT framework.

By relying on the *Hibernate* ORM library for database access, CDM can make abstraction of the specific database technology, instead relying on *Hibernate* for mapping the domain object model to SQL server commands. Finally, the project leverages the *Spring Security* framework for web server authentication and authorization, as already mentioned in Subsection 3.3.4.

3.4.2 Quality assurance

In order to ensure software quality, while avoiding spending excessive time on manual testing, I developed CDM using a Test-Driven Development (TDD)[6] approach, one of the elements of the eXtreme Programming (XP) methodology[7].

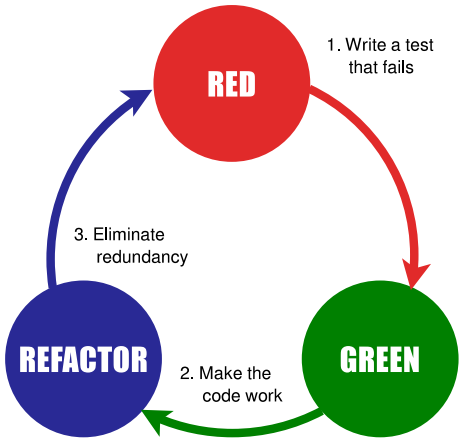


Figure 3.10 – The Test-Driven Development workflow.

The TDD workflow is depicted in Figure 3.10. Rather than starting the process with the writing of production code, first one of the requirements is selected and transcribed into an automated test. Requirements should be granular enough, so that their functionality can be verified by a single test. If they are not, they should be split into multiple requirements. Tests typically pertain to a single unit of code only, corresponding to a single class in an Object-Oriented Programming (OOP) environment. At this point, while the other tests should succeed, the new test should fail, since its associated functionality has not been implemented yet. Second, production code is modified until the new test succeeds as well. A third "refactoring" step improves the software's design and eliminates any redundancy. No functionality is altered, which is reflected by all tests remaining successful through this step. This process is repeated with the next requirement to be implemented.

Likewise, when software errors are discovered, a very similar three step process is applied to fixing them, with the requirement being to fix the error:

1. Implement a test verifying expected operation, which, at this point, should fail, since the software behaves incorrectly;
2. Fix the erratic code, by implementing the expected behaviour. Observe that the newly written test now does succeeed;
3. Improve software design by refactoring.

Adhering to this process will avoid the recurrence of failures that have been encountered before.

The TDD workflow provides many benefits. As Malaiya et al.[51] have shown,

higher test coverage is associated with lower software defect rate. Hence, projects using a TDD approach will have fewer defects than projects that do not. Indeed, TDD catches most errors during the development phase, rather than relying on a manual, post-development quality assurance phase. The latter does not enable the same fine-grained testing as TDD does, leaving more errors undetected. By contrast, the TDD approach to developing CDM has enabled mostly error-free upgrade deployments, despite very low degrees of prior manual testing.

Another, equally important, advantage of TDD is the fact that it encourages better design and loose coupling[5, 65]: writing tests first forces one to think about how code will be used by clients, hence about its interface, rather than how it should be implemented.

Also, development on a TDD project rarely requires the use of a debugger, since the sequences of steps executed by production code have already been verified by unit tests[49]. This observation is corroborated by my experience with the CDM project. Whenever errors do occur, they usually trigger a failing unit test, localizing the problem to the unit verified by the failed test.

Another important advantage, and in fact perhaps one of the most important ones, is that, by being backed by an entire test suite verifying correctness of the software project, developers can confidently effect changes, as errors introduced by these changes will be detected by the test suite. By contrast, in software projects lacking such a test suite, any errors introduced by modifications to the code must be captured in the quality assurance phase. As the latter typically fails to attain the test coverage of TDD processes, this will cause more errors to pass undetected. As a result, developers in such projects are more cautious to introduce changes, and avoid improving software design by refactoring. The end result of the lack of a comprehensive test suite in a project therefore is unmaintainable code in which much needed changes are avoided.

Finally, by translating requirements into unit tests, a project's test suite effectively provides documentation for the classes under test, which, contrary to comments or design documents, will never go stale, as in order to compile and pass, the test suite will have to evolve along with the production code.

CDM makes use of the combination of the `JUnit` testing framework and the `EasyMock` library for writing unit tests. Since higher test coverage has been associated with lower software defect rates[51, 46], the build process automatically tracks the project's test coverage using the `Cobertura` tool.

3.4.3 Software configuration management

Software configuration management is the process of tracking and managing changes in software throughout its lifecycle. This entails a broad array of processes, the most important of which are detailed in this subsection. First, Version Control Systems (VCSs) are discussed. This is followed by build automation, which is required to obtain reproducible builds. Finally, continuous integration is explained.

Version Control System

VCSs are software systems that track and manage changes in a project over time. This allows viewing and restoring the state of a software project at a certain time in the past, or comparing its current state with its state in the past. Such functionality is useful to revert to a known good state of the project, after a software failure is introduced. Or it can be used to perform root cause analysis for a problem reported for an earlier release of the software.

Many VCS systems exist. A primary distinguishing feature is whether atomic transactions occur on individual files, or on the entire project at once. If the former, problems may arise if the operations of different developers are interleaved, thus recording inconsistent states. This is solved by implementing a locking mechanism, preventing interleaved operation. Most modern VCS systems avoid this problem altogether by effecting transactions on an entire project at once.

Another important distinction should be made between traditional client-server systems and Distributed Version Control Systems (DVCSs). In the former, a centrally managed server stores the complete history of files in a project, called a repository. Client applications on developers' workstations then interrogate the server for requesting specific versions of files. Client workstations, in this model, only maintain a single version of the files in a project. By contrast, DVCSs do not operate according to a client-server model, but take a peer-to-peer approach instead. In this approach, every peer maintains a copy of the entire software repository. Thus, technically, no single "reference" repository exists, though projects may assign reference status to a repository by convention. Also, apart from synchronization operations between repositories, all operations are performed locally, resulting in fast operation. More importantly, it enables operation while disconnected from the network.

The CDM project makes use of `git`, an open-source DVCS which commits changes to an entire project tree, rather than on individual files.

Build automation

Build automation entails the scripting of the process of creating deliverables derived from source code. Deliverables can include binaries, shared libraries, documentation, test results, test coverage analysis, and possibly even deployment to production systems, amongst others. Build automation will prevent errors arising from the manual creation of deliverables.

If done correctly, the automatic build process starts with the installation of all required modules, such as compilers, linkers, and libraries, using versions specified by the build script. By tracking build scripts in a VCS system, this allows the creation of *reproducible builds*[25], which, in case errors are reported for earlier releases, allow to recreate the exact same environment used by that release.

Several tools exist for simplifying build automation. Most notable are `make`, `ant`, `ivy` and `maven`, all of which are open-source. CDM utilizes the latter for build automation.

Continuous integration

Continuous Integration (CI) is one of the techniques practised in XP[7]. It refers to the process of frequently committing small, incremental changes, which have been verified to pass all unit tests, to a reference repository. Typically, a build server automatically runs the build process to create deliverables and collect test results and source code metrics, either at regular intervals, or for every committed change, and presents the results in a very visible manner.

Through the visibility of up-to-date test results and source code metrics, CI offers the advantage of providing immediate feedback to developers on the quality, functionality and system-wide impact of their changes.

In the context of CDM, I use the open-source `Jenkins` CI server.

3.5 Architecture

At a high level, CDM's architecture consists of a number of modules, organized in layers, as shown in Figure 3.11.

At the lowest layer, `cdm-common` groups common functionality needed by both client and server code. Apart from some helper classes, this mainly includes

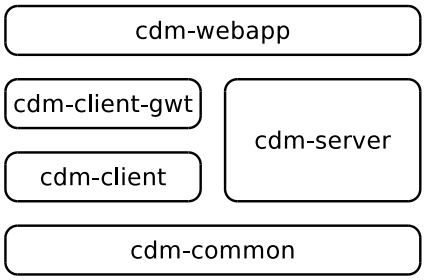


Figure 3.11 – CDM's high-level architecture consists of a set of separate modules.

the representation of CRF metadata and structure, which are transmitted in RPC calls between client and server, hence are required by both.

The `cdm-server` module, which depends on `cdm-common`, is responsible for handling server-side communications and database management.

The `cdm-client` module handles client-side communications, user interface and user interaction handling. It makes abstraction of the specific user interface widgets used, handling only user interface *logic*, with `cdm-client-gwt` attaching specific GWT widgets to `cdm-client`'s user interface logic. This decoupling between user interface *logic* and *widgets* is achieved through the use of the *AbstractFactory* design pattern[28], illustrated by Figure 3.12. In this design, the `NodeFactory` interface forms the *AbstractFactory* pattern, together with the interfaces derived from `EntryNode`. These reside in the `cdm-client` module. Their implementations, namely the `GwtNodeFactory` class and classes such as `GwtTextEntryNode`, `GwtDateEntryNode`, etc., are located in the `cdm-client-gwt` module. Thus, in this design, module `cdm-client-gwt` depends on module `cdm-client`. The organization of user interface code in this manner, facilitates potential future implementation of other user interfaces, such as desktop or tablet user interfaces, or integration in hospital IT systems, such as UZ-Leuven's Klinisch Werkstation (KWS). Indeed, a new user interface would only require the implementation of an additional module, parallel to `cdm-client-gwt`, using different widgets for implementing user interface logic.

Finally, module `cdm-webapp` provides an entrypoint to the application, and binds module `cdm-client`'s *AbstractFactory* interface to its specific implementation in `cdm-client-gwt`.

At an intermediate level, the software is designed around components, grouping classes that cooperate to provide conceptually related functionality. Components use other components' functionality through special classes exhibiting the *Facade* design pattern[28], and providing central access to the respective components'

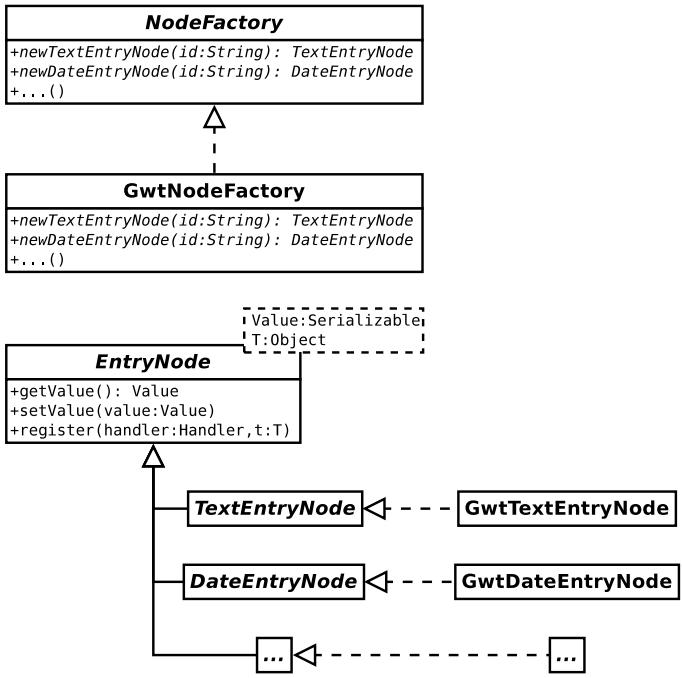


Figure 3.12 – CDM's user interface *logic* and *presentation* are decoupled by means of the *AbstractFactory* design pattern represented here. The `NodeFactory` interface is an *AbstractFactory*, implemented by the `GwtNodeFactory` class.

functionality. At the class level, design patterns from Gamma et al.[28] are used extensively to organize interaction between classes.

3.6 Server setup

The primary CDM server is hosted on a virtual private server operated by Infrastructure as a Service (IaaS) provider *Linode* (`http://www.linode.com`), who are responsible for monitoring hardware and handling related issues. The operating system installed is Debian GNU/Linux v6.0.8, which I configured to install security updates automatically. CDM runs within an open-source Apache Tomcat servlet container. An SSL certificate, issued by TERENA SSL Certificate Authority (CA), and expiring every three years, provides encrypted communication between clients and server.

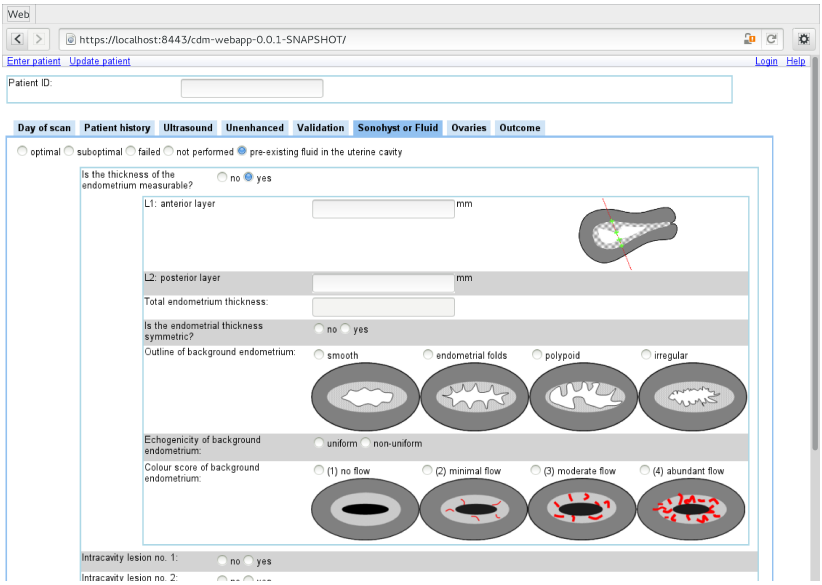


Figure 3.13 – CDM's user interface. (Screenshot of the "Sonohysterography or fluid" section, included in the IETA #1, #2, and #3 studies, as presented by the CDM user interface.)

The virtual server additionally runs the *MySQL* Relational Database Management System (RDBMS). Shell access is provided by *OpenSSH*, which provides both secure and private communications.

A simple shell script, leveraging both the *GNU Privacy Guard* encryption software and *OpenSSH*, creates daily backups and transfers them to a server in a different geographical location, hosted at the KU Leuven Department of Electrical Engineering (ESAT). Another shell script automates the deployment of new CDM releases, keeping backups of previous, known good, releases, for disaster recovery.

3.7 Results

I have developed a generic EDC software framework with a client-server architecture, including a user-friendly web-based interface. This user interface is available at `https://cdm.esat.kuleuven.be` and is shown in Figure 3.13.

J.L. Alcazar	B. Benacerraf	T. Bourne
V. Chiappa	M.E. Coccia	A. Czekierdowski
D. De Neubourg	A. Dilegge	F. Dorella
E. Epstein	D. Fischerova	R. Fruscio
S. Guerriero	L. Haakova	J. Heymans
A. Jakab	L. Jokubkiene	Y. Kacem
J. Kaijser	C. Lanzani	F.P.G. Leone
L. Manuela	F. Mascilini	G. Opolskiene
M.A. Pascual	N. Raine-Fenning	F. Rizzello
A. Rossi	P. Sladkevicius	A. Smith
A.C. Testa	D. Timmerman	L. Valentin
T. Van den Bosch	C. Van Holsbeke	C. Van Pachterbeke
D. Van Schoubroeck	K. Van Tornout	B. Virgilio
A. Votino	L. Zannoni	R. Zlotorowicz-Grochowska

Table 3.2 – Clinicians selected for participation in CDM survey. Of 42 participants, 28 responded, resulting in a 66.7% response rate.

Survey

In order to obtain user feedback about CDM's EDC interface, I sent a request for participation in an anonymous survey to the clinicians listed in Table 3.2. This list includes any clinician who contributed at least 10 patient entries to the IETA studies, or participated in an inter-rater agreement study using the CDM-based interface described in Subsection 4.2.2. These selection criteria should ensure participants have sufficient experience with CDM's user interface to be able to provide relevant feedback. Of these 42 clinicians, 28 responded, for a response ratio of 66.7%.

The survey consisted of three parts, arranged on as many pages. In the first part, survey participants indicated their level of agreement with a number of statements, to enable a quantitative assessment of user satisfaction. I copied these questions from an older survey I conducted among the participants of the first CDM-based inter-rater agreement study, listed in Table 4.1. The second part aimed to estimate how often and which problems users encounter. Finally, using open-ended questions, the third part sollicited users' feedback, about what they do or do not like about CDM, and what could be improved, providing useful information for prioritizing further CDM development.

Table 3.3 presents quantitative results of the multiple-choice questions present in the first and second parts of the survey. As these results show, CDM users are quite satisfied with CDM's user-friendliness, and its capability to include

VASs, as in Figure 3.3. Additionally, they were very enthusiastic about the possibility to include pictograms. All would consider using CDM for their own studies. A large majority of 78.6% of participants experienced problems for less than 5% of their interactions with CDM.

Table 3.4 enumerates the survey's open-ended questions, as well as some of their most relevant answers. As this table shows, most frequently requested is the ability to view and print information about patient entries that are complete. Comprehensive survey results are available at `https://www.surveymonkey.com/results/SM-QDYK9B7/`.

Software metrics

The size of the CDM project is examined in Table 3.5, for the different modules separately. Note that, apart from data collection functionality, module `cdm-server` includes data analysis functionality as well, adding to the module's size. The latter functionality will be discussed in detail in Chapter 6.

Analyzing test coverage results listed in Table 3.6 shows relatively low test coverage for the `cdm-client-gwt` module, responsible for binding widgets to the user interface logic, and the `cdm-webapp` module, which provides the application's entry point. The functionality of these modules cannot be verified in isolation, by unit tests, explaining the low test coverage. As these modules are of low complexity, and change very rarely, this does not negatively impact software quality.

Modules `cdm-common`, `cdm-server`, and `cdm-client`, which do change frequently, and do contain a lot of complexity, all have around 90% line and branch coverage, guaranteeing excellent software quality.

Average line and branch coverages for the entire project, weighted to the number of lines of production code, are 85% and 84%, respectively, providing good overall test coverage.

The framework's design and modularity ensure future extensibility. The separation between user interface *logic* and *widgets*, for example, enables the implementation of alternative user interfaces, such as desktop or tablet interfaces.

3.8 Conclusion

In this chapter, I have discussed several aspects of the EDC software component I developed. While its development occurred in the context of the IETA studies,

Question	Score
1. Evaluation	
1a. Please indicate your level of agreement with the following statements (0 = no agreement; 5 = neutral; 10 = complete agreement; N/A = unknown).	
• CDM is user-friendly.	8.6
• The layout of studies is clear.	8.6
• The VAS is user-friendly.	8.1
• CDM's VAS is a good alternative to a paper VAS.	8.2
• Pictograms help to clarify questions.	9.4
• Pictograms help to differentiate multiple-choice options.	9.2
• Pictograms next to multiple-choice options will improve reliability.	9.3

Question	Percentage
1b. Would you consider using CDM for your own studies?	
• no	0%
• yes	100%
2. Software problems	
2a. How frequently do you create a (new) patient entry with CDM?	
• < 1 per month	32.1%
• $1 - 2$ per month	35.7%
• $2 - 10$ per month	21.4%
• > 10 per month	10.7%
2b. Have you participated in an inter-rater agreement study?	
• no	60.7%
• yes	39.3%
2c. Please estimate how often you encounter software problems with CDM, expressed as a percentage of your interactions with CDM.	
• $< 5\%$ – less than 1 out of 20	78.6%
• $5 - 10\%$ – between 1 and 2 out of 20	14.3%
• $10 - 30\%$ – between 1 and 3 out of 10	7.1%
• $> 30\%$ – more than 3 out of 10	0%

Table 3.3 – Quantitative results from the multiple-choice questions in the CDM survey.

2. Software problems
2d. Please describe any problems you have encountered in CDM.
The program marks a question as unanswered when it has been answered. Sometimes very slow, but this is probably dependent on the internet browser
3. Good and bad
3a. Please describe what you like about CDM.
Data collection on an internet based database allow to use any computer in my institution and even at home. In particular CDM is very clear and simple
I can avoid use of paper
Most features, pictograms
3b. Please describe what you dislike about CDM.
Sometimes it is uncomfortable that we can not see the information about our patients we have filled.
Too slow for many browsers, marks answered questions as unanswere quite often
3c. Please describe three or more changes you would like to see in CDM. These can be both changes to existing functionality or new features.
1. To see how many patiens I have filled. 2. To have some feedback - is everything OK, maybe I am doing some mistakes when entering patients data, maybe the study is finished and it is late to enter new data.
[...] It should be possible to go back and retrieve the infroamtion about each patient at any time (but not to change data of course) it should be possible to print the full report including the histology I did not understand the questions about MCQ: is there an MCQ test in the clinical data miner?
More clarity about optional or not sections
Printing functionality. Intuitive screen for variable entry. Audit tools for commercial studies

Table 3.4 – Open-ended questions listed in CDM survey, along with a selection of the most relevant answers.

Module	Production code (SLOC)	Test code (SLOC)
`cdm-common`	5862	7023
`cdm-server`	15260	28109
`cdm-client`	3595	7607
`cdm-client-gwt`	4090	5123
`cdm-webapp`	321	177
Total	29128	48039

Table 3.5 – Source Lines of Code (SLOC) per module. Note that these numbers include the line counts from the APIs which will be elaborated on in Chapter 6.

Module	Line coverage (%)	Branch coverage (%)
`cdm-common`	91	94
`cdm-server`	92	90
`cdm-client`	88	91
`cdm-client-gwt`	53	42
`cdm-webapp`	34	100
Weighted average	85	84

Table 3.6 – Test coverage for the different modules, as ratios of lines and branches covered, respectively. Averages are weighted according to the modules' production code sizes from Table 3.5.

I designed it as a generic component for data collection in CRF-driven clinical studies, with questionnaires defined by means of spreadsheets.

It has been well received for its user friendliness, and its web-based interface considerably simplifies the organization of multi-centre studies, compared to traditional client-server approaches. The integration of pictograms in the user interface assists clinicians with the correct classification of patients.

Several factors contribute to CDM's excellent maintainability. Both the TDD approach and the extensive use of design patterns have promoted a loosely coupled design, simplifying maintenance. Additionally, the extensive test suite, resulting from the TDD approach, allowed to confidently apply changes. Without it, the time spent on manually testing upgrades would have been prohibitive for a project with a single developer.

CDM's architecture, built around a set of modules, provides future extensibility:

modules can be exchanged for other modules, allowing the implementation of other user interfaces, enabling integration in hospital systems such as UZ-Leuven's KWS, or allowing to repurpose CDM for other uses. An example of the latter will be presented in Chapter 4.

Chapter 4

Influence of pictograms on data quality

Clinical studies require reliable data for drawing correct conclusions. In the case of imaging-based modalities, however, data are subject to a sonologist's interpretation, leading to variability in the results. With Clinical Data Miner (CDM)'s ability to integrate pictograms in its user interface, this chapter analyzes if and how these pictograms influence data quality.

4.1 Introduction

In order for clinical studies to produce meaningful results, the collected data have to be reliable. For variables such as patient age, weight, concentration of substances in a blood sample, this can be easily attained. For other variables, this may not be as straightforward.

When analyzing imaging modalities such as ultrasound images, for instance, a sonologist will look for the absence or presence of certain characteristics, or attempt to measure certain features in the image. Here, the quality of the obtained data is influenced on four different levels:

- First, due to the nature of ultrasonography itself, the quality of the image, from which the data are obtained, depends on various parameters, such as the position of the ultrasound probe, its frequency, its angle, the pressure applied, whether or not sonohysterography is used, the Body Mass Index

(BMI) of the patient, the presence of acoustic shadows, etc. Finding the optimal parameters for visualizing an image depends both on experience of the sonologist as well as knowledge of the latest recommendations regarding technologies to be used.

- Second, correctly recognizing patterns in ultrasound images can be challenging. Often, widely differing pathologies may present with virtually identical ultrasound images. Invasive endometrial cancer in the myometrium and adenomyosis, for example, can produce very similar ultrasound images. A correct interpretation of such images depends on the experience and the ability of a sonologist.

- Third, a poorly designed Case Report Form (CRF) can prevent correct reporting of patient data. Examples are categorical variables for which categories are missing, or unclear instructions in the CRF. Some of these design mistakes can be fixed after the study started. However, an inadequate structure in the hierarchy of questions is very difficult to mend after the fact. Such problems can be avoided by careful consideration during the design stage of the study.

- Finally, a study participant may misinterpret a CRF's question, or the different categories of a categorical variable, due to lack of familiarity with its terminology. In exploratory research aiming to find a diagnostic model for a hitherto unmodelled disease, an important initial step is to identify variables that may influence a diagnosis, and, if needed, to define new terminology describing those variables and their possible values. Hence, many researchers may initially be unfamiliar with the project's terminology.

The publication of consensus papers, such as Timmerman et al.[69], Condous et al.[12], and Leone et al.[45], alleviates the latter problem to a certain extent. However, in developing the CDM software framework, we aimed to improve it even further, by enabling the integration of pictograms in CRFs. This allows to clarify questions and/or the differences between the possible answers to a question. In this chapter, I describe how we verified if the addition of pictograms indeed improves data quality. This discussion starts in Section 4.2 with a description of study set-up, the user interface developed for this study, and the method of analysis. Results are presented in Section 4.3, for variables from the "*unenhanced ultrasound*" and "*sonohysterography*" sections of the International Endometrial Tumour Analysis (IETA) studies, respectively. Finally, conclusions are listed in Section 4.4.

D. Van Schoubroeck
P. Sladkevicius
J. Heymans
L. Jokubkiene
L. Zannoni

Table 4.1 – List of study participants.

4.2 Methods

This section starts with an explanation of how this study was conceived. Then follows a discussion of the Electronic Data Capture (EDC) user interface that I developed for this study, as well as a more general one derived from it. It ends with a description of the method used to analyze the collected data.

4.2.1 Study design

In order to examine how adding pictograms to a CRF affects data quality, I organized a study that required the evaluation of a set of one hundred anonymized ultrasound images. This set comprised fifty images obtained by means of unenhanced ultrasound technology, and fifty using sonohysterography, all supplied by T. Van den Bosch. D. Timmerman contacted five clinicians, listed in Table 4.1, all specialized in gynaecological ultrasound, who agreed to participate in the study.

Unenhanced ultrasound and sonohysterographic images were evaluated according to different questionnaires, listed in subsections B.1.1 and B.1.2, respectively. These were derived from questionnaires used in the IETA studies. Both questionnaires were combined into a single CRF, with the top question allowing study participants to select between either.

During the first phase, for each ultrasound image, participants were requested to fill in a CRF without pictograms, while during the second phase, pictograms were added to some of the variables in the CRF.

In order to minimize potential learning effects, we observed a minimum interval of two weeks between both phases. Additionally, the user interface that I developed for this study randomizes the order in which images are presented to study participants.

Inter-rater agreement changes for variables that gained pictograms in the second phase of the study resulted not only from the addition of pictograms, but

also from a learning effect caused by seeing the same set of images twice. By randomizing the image order, this latter effect should be reduced. By comparing these changes with those experienced by variables with no pictograms in either phase of the study, it should be possible to estimate the magnitude of the two effects.

4.2.2 `ImgStudy` user interface

Rather than providing the raters with a set of ultrasound images and a questionnaire to fill in for each ultrasound image, I developed a custom user interface for conducting inter-rater agreement studies, starting from the EDC software component described in Chapter 3. While the normal EDC user interface displays a field for patient identification on top of a questionnaire, the custom user interface shows an image next to a questionnaire containing the variables to be evaluated. The image is selected randomly from a study specific directory containing the images to be evaluated. This random order in which images are shown aims to reduce the learning effect between different phases of a same study. Finally, for usability, the interface shows the remaining number of images to be evaluated. The resulting user interface is demonstrated in figures 4.1 and 4.2, for the study phases without and with pictograms, respectively.

Development of this user interface occurred in the `ImgStudy` software project, separate from the CDM software framework. Reusing the latter's `cdm-common`, `cdm-client`, `cdm-client-gwt`, and `cdm-server` components drastically reduced the extra development, leaving only the following to be implemented:

- A new RPC call, modelled in Figure 4.3, allowing clients to request a list of images that still need to be evaluated. This list contains the images from the study specific image directory, excluding images that were already evaluated by the rater;
- A different Model-View-Controller (MVC) software pattern implementation, including a different layout *view* and *controller*, adapted to the needs of inter-rater agreement studies;
- A new application entrypoint, calling into the new MVC pattern.

Through extensive reuse of the CDM software components, the implementation of this user interface only required 782 new lines of *Java* code.

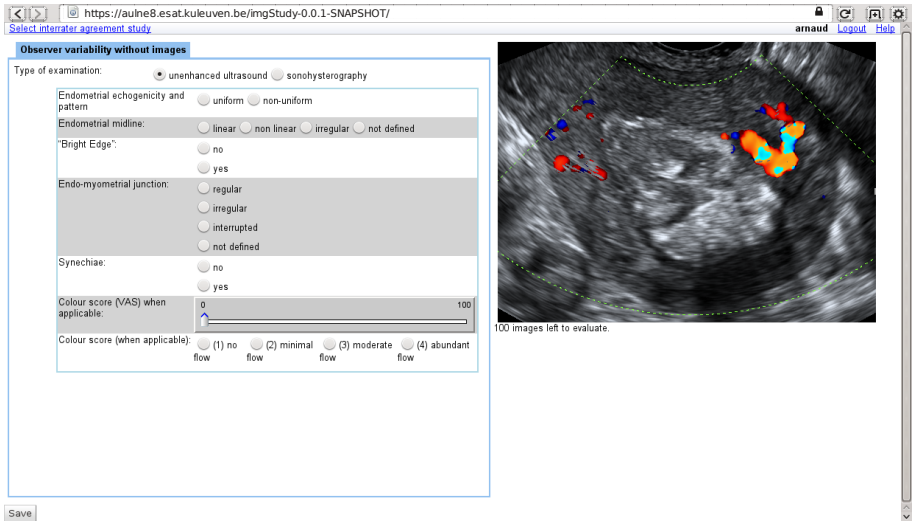


Figure 4.1 – CRF *without* pictograms for the first phase of the study. (Screenshot of the "Observer variability without images" section, included in the study investigating the influence of pictograms on inter-rater agreement, as presented by the CDM user interface.)

4.2.3 `MediaStudy` user interface

The `ImgStudy` user interface described in subsection 4.2.2 has proven very useful for determining inter-rater agreement of the subjective assessment of features in imaging modalities. It has enabled the organization of a number of such studies, which are listed in more detail in Appendix A. It is, however, limited to the evaluation of images, and does not support other media types, whereas the IETA #2 study required assessment not of ultrasound images, but of ultrasound video clips. I created the `MediaStudy` prototype, based on the user interface from the previous subsection, in order to support the latter.

Since the HTML5 standard for embedding video clips in web pages had only just been published[15], support by the major browsers for this standard was still very limited at the time. Some of the most recent browsers had some support, but no single video codec was supported by all of them. As CDM avoids imposing browser requirements, it was preferrable to avoid imposing any for this user interface, so another solution was needed.

Developing a custom solution for viewing video clips, based on Adobe® Flag® technology, would require a substantial time investment. Instead, the established solution at the time was to use a video website such as YouTube™. This site

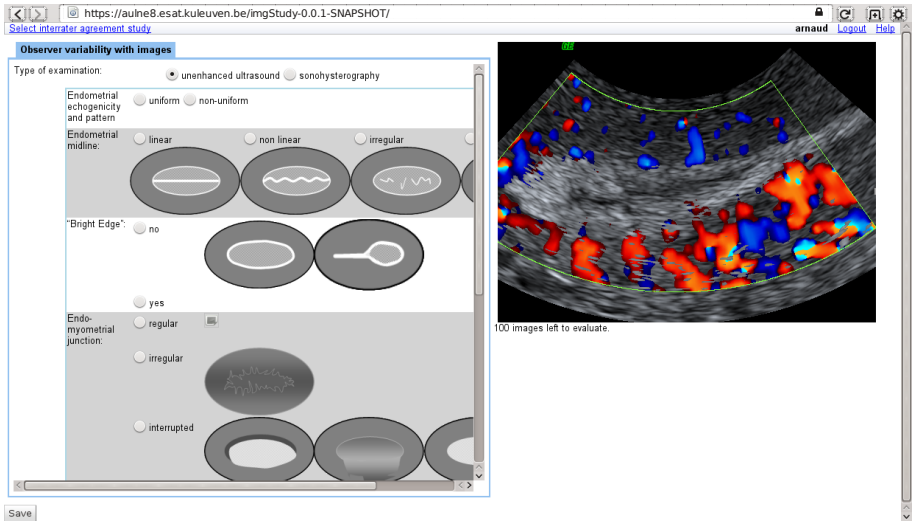


Figure 4.2 – CRF *with* pictograms for the second phase of the study. Except for the added pictograms, the questionnaire is identical to that of Figure 4.1. (Screenshot of the "Observer variability with images" section, included in the study investigating the influence of pictograms on inter-rater agreement, as presented by the CDM user interface.)

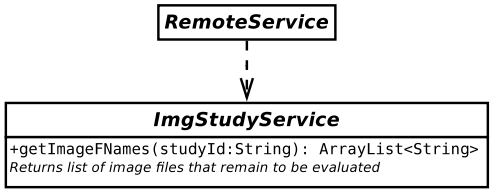


Figure 4.3 – Unified Modelling Language (UML) diagram of the RPC call introduced in the inter-rater agreement study user interface. With this call, a user's client can obtain a list of images that remain to be evaluated.

returns a Uniform Resource Identifier (URI) for each uploaded video clip, which can be embedded in other web pages. Since these URIs end with an opaque string, consisting of random sequences of letters, digits, and punctuation, and by keeping them unlisted on YouTube™, these videos are only accessible to study participants.

An extension to the CRF parsing code in the `cdm-server` module provided a mechanism to supply a list of URIs, through the parsing of an extra page in the study definition spreadsheet. Rather than creating separate software projects for handling the different supported media types, however, and with the eventual aim to reintegrate this specialized user interface in the main CDM software project, I generalized the server-side changes as indicated in Figure 4.4 to handle different types of study. The type of study is specified in the study definition file, and can currently be either "consult", "image", or "youtube". These respectively handle patient visits, inter-rater agreement studies based on imaging modalities, and inter-rater agreement studies using video clips. For each study type, a specific `StudyTypeServerSupport` object is registered at startup. These `StudyTypeServerSupport` objects are responsible for constructing study type specific `Study` objects. The `MediaTypeServerSupport` class, for example, will not only parse the study's CRF structure, but will additionally parse a list of media URIs, which is supplied to the `MediaStudy` object. The `StudyTypeServerSupport` objects are registered at runtime by automatically loaded plugin classes.

As was the case for the user interface from Subsection 4.2.2, an RPC protocol was needed to enable querying which media objects still need to be evaluated. This protocol is represented in Figure 4.5.

So far, only the server code has a plugin mechanism for supporting different study types. I intend to implement similar functionality on the client side though. When completed, this will enable the integration of the user interface for inter-rater agreement studies with that for regular patient data collection.

4.2.4 Analysis

Data quality requires high inter-rater agreement levels. To measure this inter-rater agreement, the analysis makes use of Fleiss' κ coefficient[23]. The jackknife sampling technique[19] provided sample distributions for these κ-coefficients. These allowed to calculate inter-rater agreement estimates and their variance.

Due to the CRF's hierarchical structure, participants could only enter certain combinations of variables for the evaluation of images, with other variables structurally missing. For calculating inter-rater agreement of a certain variable,

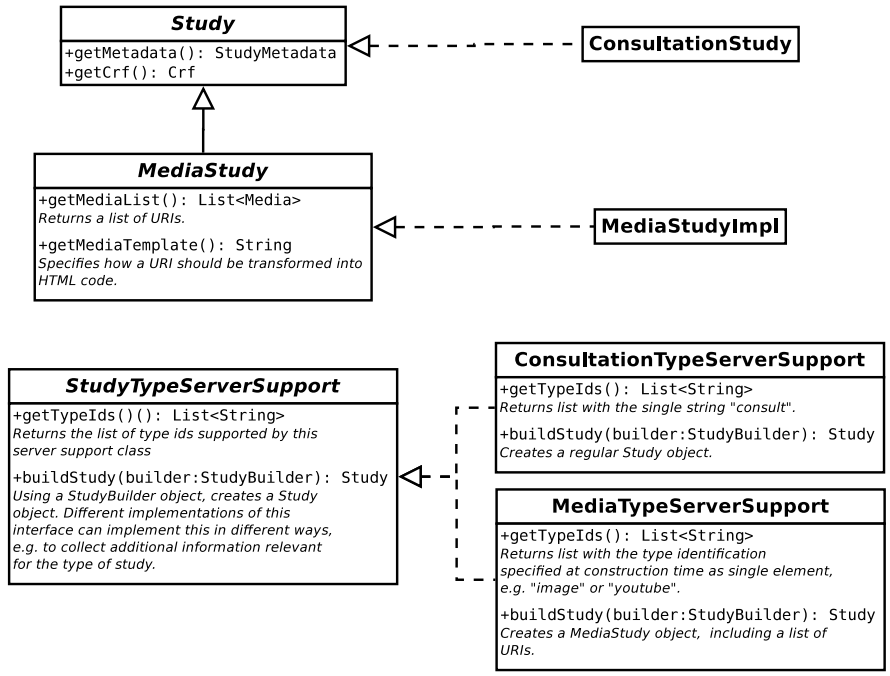


Figure 4.4 – Main server classes responsible for handling of generic study types.

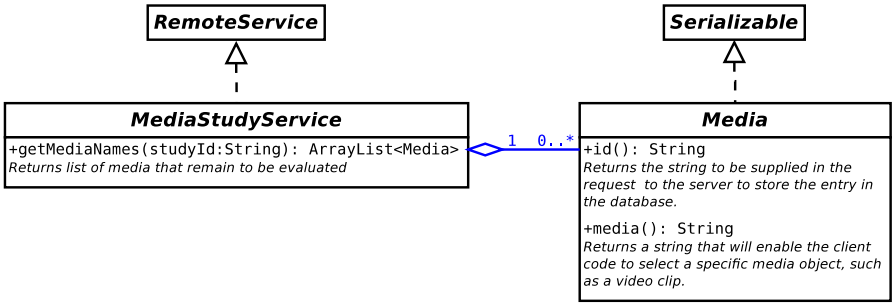


Figure 4.5 – Representation of the RPC call implemented by the generalized inter-rater agreement study user interface. This RPC mechanism allows to obtain a list of `Media` objects that a user still needs to evaluate.

I only considered those images for which all raters evaluated that variable. Notably, this excluded eight sonohysterography images that any of the raters misinterpreted as unenhanced ultrasound images. No inter-rater agreement was calculated for the variable "*colour score*" recorded as a continuous variable on a VAS, as most techniques for evaluating inter-rater agreement of continuous variables require conversion to an ordinal variable, which was, for "*colour score*", already present in the study's CRF as a separate variable.

4.3 Results

This section discusses results from our study for "*unenhanced ultrasound*" and "*sonohysterography*" variables in the following subsections, respectively.

4.3.1 Unenhanced ultrasound

Fleiss' κ coefficients for the unenhanced ultrasound variables are listed in Table 4.2 and depicted in Figure 4.6. Variable "*echogenicity*", indicated in dark grey, did not have pictograms in either phase. Contrary to what I had expected, the κ coefficient difference between the two phases of the study was not negligible, with an improvement of 4.7% in the second phase. This suggests a modest learning effect occurred, possibly due to the relatively small time interval of two weeks between phases.

The other variables, which did obtain pictograms in the second phase, also experienced substantial differences in inter-rater agreement between phases. Except for variable "*vascular pattern*", their inter-rater agreements improved considerably more than for the "*echogenicity*" variable, with improvements between 20.4% and 66.5%, suggesting that at least part of these improvements should be attributed to the addition of pictograms. Variable "*vascular pattern*" exhibited deteriorated inter-rater agreement, with a decrease of -12.4%, suggesting that its pictograms confused participants rather than helped them.

4.3.2 Sonohysterography

Results for sonohysterography are shown in Table 4.3 and Figure 4.7. As in Subsection 4.3.1, dark grey rows represent variables for which no pictograms were available in either phase of the study. The other variables did receive pictograms in the second phase.

	Variable	$\hat{\kappa_1}$	σ_{κ_1}	n_1	$\hat{\kappa_2}$	σ_{κ_2}	n_2	$\hat{\kappa_2} - \hat{\kappa_1}$	$\frac{\hat{\kappa_2} - \hat{\kappa_1}}{\hat{\kappa_1}}$
1	echogenicity	0.533	0.075	50	0.558	0.080	50	0.025	4.7%
1.1	uniform pattern	0.352	0.134	9	0.535	0.128	13	0.184	52.2%
1.2	non-uniform pattern	0.330	0.067	19	0.420	0.099	16	0.090	27.3%
2	midline	0.535	0.068	50	0.645	0.071	50	0.109	20.4%
3	bright edge	0.257	0.069	50	0.428	0.109	50	0.171	66.5%
4	endo-myometrial junction	0.166	0.047	50	0.259	0.057	50	0.093	56.2%
5	synechiae	−0.008	0.006	50	−0.012	0.007	50	−0.004	50.6%
7	colour score	0.228	0.076	17	0.314	0.082	20	0.087	38.0%
7.1	vascular pattern	0.551	0.117	11	0.482	0.110	12	−0.068	−12.4%

Table 4.2 – Jackknife estimates of Fleiss' κ coefficient for the different variables pertaining to unenhanced ultrasound, in the first ($\hat{\kappa_1}$) and second ($\hat{\kappa_2}$) phases of the study. Columns σ_{κ_1} and σ_{κ_2} show respective standard errors, while columns n_1 and n_2 are the respective number of subjects available for the calculation of these coefficients.

	Variable	$\hat{\kappa_1}$	σ_{κ_1}	n_1	$\hat{\kappa_2}$	σ_{κ_2}	n_2	$\hat{\kappa_2} - \hat{\kappa_1}$	$\frac{\hat{\kappa_2} - \hat{\kappa_1}}{\hat{\kappa_1}}$
1	endometrial outline	0.548	0.075	42	0.571	0.081	41	0.023	4.1%
2	echogenicity	0.258	0.170	42	0.101	0.066	41	−0.157	−60.8%
2.1	uniform pattern	0.182	0.182	29	0.211	0.083	35	0.029	16.2%
2.2	non-uniform pattern	–	–	–	–	–	–	–	–
3	colour score	0.101	0.081	24	0.070	0.053	23	−0.031	−31.0%
5	lesion presence	0.552	0.140	42	0.660	0.174	41	0.108	19.5%
5.1	lesion origin	0.785	0.165	33	0.750	0.209	34	−0.035	−4.5%
5.1.1.1	endometrial lesion extent	0.160	0.071	28	0.077	0.080	28	−0.084	−52.2%
5.1.1.2	endometrial lesion type	0.215	0.075	28	0.073	0.061	28	−0.142	−65.9%
5.1.1.3	endometrial lesion echogenicity	0.596	0.103	28	0.708	0.101	28	0.112	18.7%
5.1.1.3.1	endometrial lesion uniform pattern	0.472	0.127	12	0.610	0.252	12	0.138	29.3%
5.1.1.3.2	endometrial lesion non-uniform pattern	0.067	0.142	9	0.173	0.140	8	0.107	160.0%
5.1.1.6	endometrial lesion outline	0.508	0.175	28	0.331	0.192	28	−0.178	−35.0%
5.1.2.1	myometrial lesion echogenicity	–	–	2	0.650	0.000	2	–	–
5.1.2.2	myometrial lesion grading	−0.131	0.000	2	–	–	2	–	–
5.3	lesion colour score	0.391	0.094	24	0.319	0.095	25	−0.072	−18.5%
5.3.1	lesion vascular pattern	0.506	0.075	23	0.503	0.084	24	−0.002	−0.5%

Table 4.3 – Jackknife estimates of Fleiss' κ coefficient for the different variables pertaining to sonohysterography, in the first ($\hat{\kappa_1}$) and second ($\hat{\kappa_2}$) phases of the study. Columns σ_{κ_1} and σ_{κ_2} show respective standard errors, while columns n_1 and n_2 are the respective number of subjects available for the calculation of these coefficients.

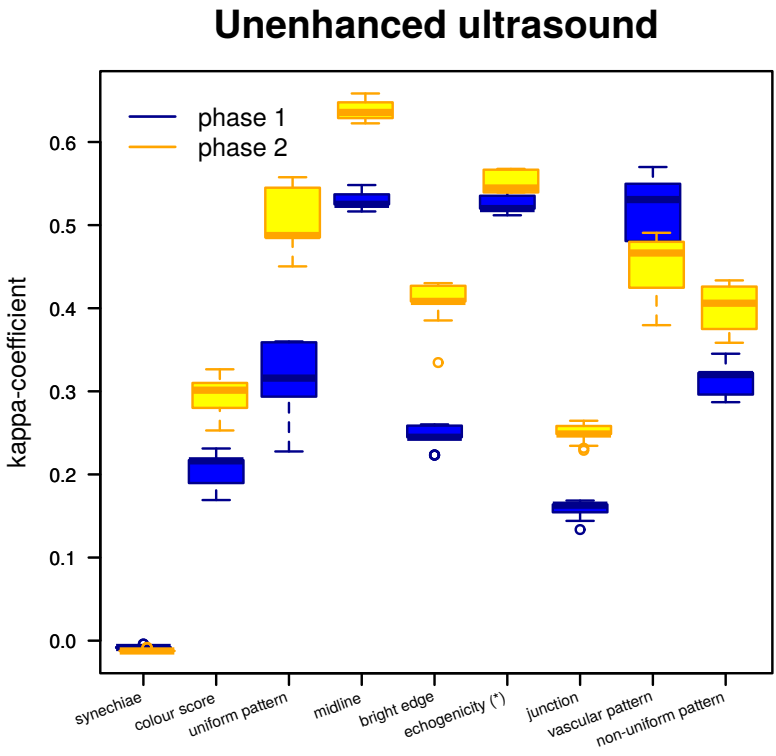


Figure 4.6 – Boxplot comparing inter-rater agreement between phases 1 and 2 for the different unenhanced ultrasound variables. Variables marked with (*) did not have pictograms in either phase of the study.

Inter-rater agreement for the variables without additional pictograms in the second phase differed considerably between phases, except for variable "*lesion vascular pattern*". Moreover, percentage differences varied widely between these variables, ranging from a deterioration of -60.8% to an improvement of $+19.5\%$. Thus, contrary to what I had expected, the learning effect for these variables did not result in comparable improvements for these variables.

For the variables that did obtain pictograms, percentage differences exhibited a similar range, between -65.9% and $+16.2\%$. Hence, no clear conclusions can be drawn from these results.

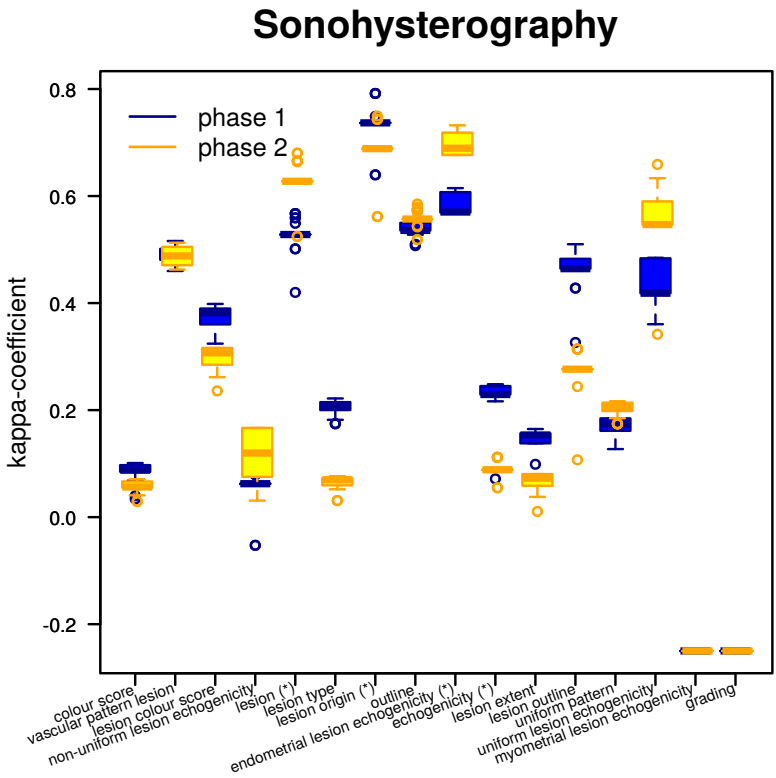


Figure 4.7 – Boxplot comparing inter-rater agreement between phases 1 and 2 for the different sonohysterography variables. Variables marked with (*) did not have pictograms in either phase of the study.

4.4 Conclusion

From the results for the sonohysterography variables clearly follows that the assumption that variables with no pictograms in either phase of the study would not experience a learning effect, or even that the learning effect would be comparable for these variables, was invalid. Consequently, using these variables as controls for the variables that did obtain pictograms in the second phase, did not provide an optimal study set-up for analyzing the effect of pictograms on inter-rater agreement. In hindsight, a better set-up would have provided a control value for each variable separately, by including more participants in the study, and having half of them evaluate exactly the same questionnaire twice,

while the other half would be shown pictograms during the second phase.

For the unenhanced ultrasound variables, results do seem to suggest that the addition of pictograms does influence Fleiss' κ coefficient, in all cases but one leading to improved inter-rater agreement.

The user interface I implemented for this study, and the later variant for IETA #2 have proven very effective for organizing inter-rater agreement studies. They have since been used in the context of several other such studies, discussed in further detail in Appendix A. Though the integration of ultrasound images in a web browser, rather than displaying them on the high-resolution screens of ultrasound machines, decreases the image quality, likely biasing the results somewhat, it has proven to deliver useful results, especially in comparing data quality provided by different technologies. Moreover, its ease of use considerably simplifies both the organization of and participation in inter-rater agreement studies. This has often resulted in very positive feedback about the user interface.

With the important role the reliability of clinical variables plays in the quality of diagnostic models, this software can be an invaluable tool for clinical research. Using it to conduct an inter-rater agreement study, prior to starting diagnostic model research, can provide an interesting insight in the reliability of the recorded variables and the effectiveness of added pictograms.

Chapter 5

Feasibility of automating machine-learning

Logistic regression is a commonly used machine-learning algorithm in diagnostic modelling. In order for it to produce models with good predictive performance, however, it requires a complex, time-consuming process, holding back automation.

This chapter examines if more sophisticated algorithms, such as Least-Squares Support Vector Machines (LS-SVM), necessitate the same preprocessing steps, or if they are able to produce good models directly from raw data. The latter would enable more extensive automation of the machine-learning workflow, which could considerably accelerate diagnostic model research.

5.1 Introduction

Deriving diagnostic models from patient data using machine-learning techniques typically involves a complex, time-consuming process of linearization of variables, analysis of second-order effects, and removal of redundant variables, in order to optimize the data format for subsequent processing by machine-learning algorithms. Using logistic regression, a commonly used classification algorithm for medical diagnostics, such preprocessing steps are required to obtain models that are able to make predictions sufficiently accurately. The complexity of these preprocessing steps limits the options for automation of the machine-

learning workflow, however, with time-consuming, largely manual preprocessing procedures as a result.

The results from this chapter show that, for the case of the International Ovarian Tumour Analysis (IOTA) data set, sophisticated, non-linear classifiers perform as well on raw data as more traditional classifiers on preprocessed data. This observation paves the way for more integral automation of the machine-learning workflow, which could therefore be integrated in Electronic Data Capture (EDC) software components.

This chapter starts with a description of the International Ovarian Tumour Analysis (IOTA) data set, which is used for the present analysis. Following that, the applied machine-learning algorithms are briefly introduced. Next, I define some common predictive performance measures, and explain the workflow used for calculating learning curves. The chapter concludes with an analysis of the results, and the conclusions drawn from this analysis.

5.2 Data set

I based the following analysis on data collected by the multi-centric IOTA consortium – an international collaboration of gynaecologists aiming to improve ovarian tumour diagnosis, in order to positively affect patient survival. Their consensus paper[69] standardizes terms and definitions for describing sonographic features of ovarian tumours, including ovarian cancer, which is the most important gynaecologic cause of cancer deaths among women[22, 41]. The IOTA consortium based the Case Report Form (CRF) utilized during patient data collection on these terms and definitions, resulting in a data set from which several diagnostic models[70, 72, 73] have been derived since. As shown in Lu et al.[50] and Van Holsbeke et al.[77], the predictive performances of these models outperform the Risk of Malignancy Index (RMI)[39], the current standard model used for diagnosing ovarian tumours in UK hospitals.

The patient data used in the present analysis, consist of information from phases 1, 1b and 2 of the IOTA database, for a total of $N = 3511$ data points. These were collected between the years 1999 and 2007 by gynaecological sonologists from 21 different centres. The data were curated and collected in a spreadsheet containing 70 features, two of which describe multi-class (`Outcome`) and binary (`outcome1`) target variables. The latter, distinguishing between malignancy and benignity, was withheld as output variable. Features not relevant to the learning problem (`patientid`, `Set`), containing text (`Diagnosis`, `Presumed diagnosis`), or with missing values (`FIGO stage`, `FIGO a`, `CA125`, `PMB`, `prev_oopherectomy`, `ratiopaples`, `papbase1`, `papbase2`,

Phase	Center	centertype1	centertype2
Age	famhistovca	famhistbrca	pershistovca
pershistbrca	menoyn	Parity	nullipara
hysterectomy	hormtherapy	bilateral	Side
lesiond1	lesiond2	lesiond3	ovaryd1
ovaryd2	ovaryd3	pain	locularity
nrloculescat	solidd1	solidd2	solidd3
papillation	papnr	papheight	papflow
papsmooth	wallreg	incomplseptum	Shadows
Echogenicity	colscore	venous	PSV
TAMXV	RI	PI	Ascites
	Free_fluid	fluid	

Table 5.1 – Input features of the "raw" IOTA data set.

`acoustic_streaming`, `septum`, `Metastases`, `Crescent_sign`) were discarded, as well as clinicians' subjective assessments regarding patient outcomes (`subjass`, `subjprob`, `Origin1`).

The data further include "derived" features, some of which were introduced because they make intuitive sense, while others were obtained through a complex, manual preprocessing process[3], involving feature linearization, and introduction of second-order effects. The variables derived intuitively include `lesdmax`, `soldmaxorig`, `lesvol`, `solvol`, while the variables obtained through complex preprocessing comprise `soldmax`, and `ratiosolles`. Table 5.3 shows their respective definitions. Neither the addition of intuitively derived variables, nor the complex preprocessing are easily automated.

In the present discussion, I consider two data sets: one that does not include these derived variables, called the "raw data set", consisting of 46 variables, and another that does include them, the "preprocessed data set", with 52 variables. The features included in both sets are listed in Table 5.1 and Table 5.2, respectively.

5.3 Classification

This section introduces the classification algorithms used in this chapter. They include both logistic regression and LS-SVM classifiers. For better understanding, a description of Support Vector Machines (SVMs) precedes that of LS-SVM.

Phase	Center	centertype1	centertype2
Age	famhistovca	famhistbrca	pershistovca
pershistbrca	menoyn	Parity	nullipara
hysterectomy	hormtherapy	bilateral	Side
lesiond1	lesiond2	lesiond3	lesvol
lesdmax	ovaryd1	ovaryd2	ovaryd3
pain	locularity	nrloculescat	solidd1
solidd2	solidd3	solvol	soldmax
soldmaxorig	ratiosolles	papillation	papnr
papheight	papflow	papsmooth	wallreg
incomplseptum	Shadows	Echogenicity	colscore
venous	PSV	TAMXV	RI
PI	Ascites	Free_fluid	fluid

Table 5.2 – Input features of the "preprocessed" IOTA data set. Derived features, obtained intuitively or by preprocessing, are enclosed in a black box.

Variable	Definition
lesdmax	$\max(lesiond1, lesiond2, lesiond3)$
soldmaxorig	$\max(solidd1, solidd2, solidd3)$
lesvol	$\frac{\pi}{6000} \cdot lesiond1 \cdot lesiond2 \cdot lesiond3$
solvol	$\frac{\pi}{6000} \cdot sold1 \cdot sold2 \cdot sold3$
soldmax	$\min(soldmaxorig, 50)$
ratiosolles	$\frac{solvol}{lesvol}$

Table 5.3 – Definition of "derived" variables used in the generation of the IOTA model. While the introduction of the first four makes intuitive sense, the latter two require an elaborate, time-consuming analysis, described in Ameye[3].

5.3.1 Logistic regression

Logistic regression[36, 2] belongs to the family of Generalized Linear Models (GLMs). These model an outcome variable as the output of a link function applied to a linear combination of input variables. In the case of logistic regression, this link function is the logistic function $P(t) = \frac{1}{1+\exp(-t)}$. Hence, logistic regression models have the following general form, with $\mathbf{x}$ an n-

dimensional input vector:

$$\begin{aligned} \pi(\mathbf{x}) &= \frac{1}{1+\exp\left(-\boldsymbol{\beta}^T\mathbf{x'}\right)} \\ \text{with: } \mathbf{x} &= \left[x_1\ x_2\ \ldots\ x_n\right]^T \\ \mathbf{x'} &= \left[1\ \mathbf{x^T}\right]^T. \end{aligned}$$

The outcome variable y is dichotomous, with values encoded as 0 or 1. The result of $\pi(\mathbf{x})$ lies between 0 and 1, and should be interpreted as the probability that the outcome has value 1, given input vector $\mathbf{x}$:

$$\Pr(y = 1|\mathbf{x}) = \pi(\mathbf{x}).$$

Logistic regression models aim to maximize the logarithm of the model's likelihood function, which is solved using gradient descent methods. Given input vectors $\mathbf{x_i} \in \mathbb{R}^n$ and corresponding binary outcomes y_i, encoded as 0 or 1, the optimization problem becomes:

$$\max_{\beta} \sum_{i=1}^{N} y_i \ln[\pi(\mathbf{x_i})] + (1 - y_i) \ln[1 - \pi(\mathbf{x_i})].$$

Their popularity in medical diagnostic models can partly be attributed to tradition. Additionally, however, their intuitive interpretability can be an important advantage in medical contexts, as they can provide an insight into which input parameters primarily influence outcome. Moreover, since they produce probability estimates of the outcome, they lead to more informed risk assessments than would be possible with models delivering only binary predictions.

5.3.2 Support Vector Machines

Though SVM are not used in this chapter's analysis, LS-SVM are. In order to put the latter in their historical perspective, I briefly describe the evolution of SVM here.

Original Support Vector Machines formulation

While they have been extended to more complex learning problems since, in their original formulation by Vapnik and Lerner[80], SVM aim to find a hyperplane

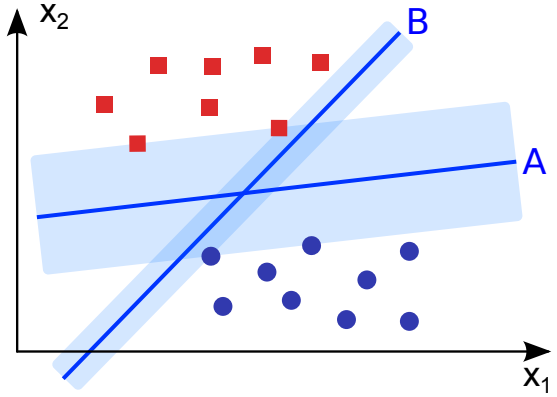


Figure 5.1 – In their most basic form, SVM find a hyperplane separating the two classes of a linearly separable data set. As this figure intuitively shows, hyperplanes with a wider margin devoid of data points around them, will tend to exhibit better generalization. (© Fabian Buërger / Wikimedia commons / CC-BY-SA-3.0 / GFDL)

separating the data points belonging to two distinct, strictly linearly separable classes in a training set.

More formally, given a training set $\{(\mathbf{x_i}, y_i) \in \mathbb{R}^n \times \{-1, +1\}\}_{i=1}^{N}$ of N data points, with n-dimensional input vectors $\mathbf{x_i}$, and binary outcome variables $y_i \in \{-1, +1\}$, strict linear separability implies that a function exists of the form:

$$f(\mathbf{x}) = \mathbf{w^T x} + b,$$

such that $f(\mathbf{x_i}) > 0$ for $y_i = +1$, and $f(\mathbf{x_i}) < 0$ for $y_i = -1$, or, more generally:

$$y_i \, f(\mathbf{x_i}) > 0, \quad i = 1, \ldots, N.$$

The hyperplane defined by $f(\mathbf{x}) = 0$ thus separates the data points from both classes perfectly. This strict separation implies that a value $\varepsilon > 0$ exists, such that $y_i f(\mathbf{x_i}) \geq \varepsilon, i = 1, \ldots, N$, defining a margin around the separating hyperplane devoid of data points. Thus, without loss of generality, we can rescale $\mathbf{w}$ and b by $\frac{1}{\varepsilon}$, such that:

$$y_i \, f(\mathbf{x_i}) = y_i \left(\mathbf{w^T x_i} + b\right) \geq 1, \quad i = 1, \ldots, N.$$

As demonstrated in Figure 5.1, intuitively, good generalization requires maximizing the width of this margin around the separating hyperplane. This

corresponds to minimizing $\|\mathbf{w}\|$. Mathematically, these conditions are expressed as a Quadratic Programming (QP) optimization problem as follows:

$$\begin{aligned} &\min_{w,b} \ \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} \\ &\text{subject to} \quad y_i\left(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b\right) \geq 1, \quad i = 1, \ldots, N. \end{aligned} \tag{5.1}$$

This corresponds to the original SVM formulation as described in Vapnik and Lerner[80]. Because they aim to maximize the width of the margin around the separating hyperplane, SVM are called "maximal margin classifiers".

Extension to non-separable data sets

Since few data sets are linearly separable, Cortes and Vapnik[14] generalized the formulation of (5.1), with the introduction of a "soft margin", to:

$$\begin{aligned} &\min_{w,b} \ \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to} \quad y_i\left(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b\right) \geq 1 - \xi_i \\ &\qquad\qquad \xi_i \geq 0, \quad i = 1, \ldots, N. \end{aligned}$$

This introduces "slack variables" ξ_i into the optimization problem, allowing a limited amount of overlap between data classes. As a result, it is able to provide solutions in situations for which the original formulation from (5.1) would be unable to provide one.

This formulation introduces a hyperparameter C, which allows to make the following trade-off: If C is low, the optimization problem's objective function will be dominated by its first term, $\frac{1}{2}\ \mathbf{w}^{\mathbf{T}}\mathbf{w}$, and will therefore favour solutions with a large margin, potentially at the expense of misclassifications in the training set. If, on the other hand, C is high, solutions will tend to avoid misclassification of training data, sacrificing margin width. A good model will strike a balance between maximizing margin width on the one hand, and minimizing training data misclassifications on the other, by tuning the hyperparameter C.

Extension to non-linear models

The aforementioned SVM formulations are only capable of modelling data if their underlying model behaves in a linear way. This can be alleviated by introducing a feature map $\varphi : \mathbb{R}^n \to \mathbb{R}^m : x_i \mapsto \varphi(x_i)$, mapping the input space

to a feature space, in which the transformed data can be separated linearly. The SVM formulation will therefore be applied to the data mapped to the feature space, resulting in the following problem:

$$\begin{aligned} \min_{w,b} \ & \frac{1}{2}\mathbf{w^T w} + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i\left(\mathbf{w^T}\boldsymbol{\varphi}(\mathbf{x_i}) + b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{5.2}$$

The problem is that usually this $\varphi(\mathbf{x_i})$ mapping between input space and feature space is unknown, and may map to an infinite-dimensional feature space, precluding use of the above formulation. To solve this, one formulates the Lagrangian dual problem[8] of (5.2) as follows:

$$\begin{aligned} \max_{\alpha} \ & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \boldsymbol{\varphi}(\mathbf{x_i})^{\mathbf{T}}\boldsymbol{\varphi}(\mathbf{x_j})\alpha_i\alpha_j \ + \ \sum_{i=1}^{N}\alpha_i \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{N}\alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{5.3}$$

Since in this equation, the mapping function $\varphi(\mathbf{x_i})$ only appears in the inner product $\boldsymbol{\varphi}(\mathbf{x_i})^{\mathbf{T}}\boldsymbol{\varphi}(\mathbf{x_j})$ of pairs of data points mapped into feature space, the *kernel trick* may be applied, replacing this inner product by the kernel function defined as follows:

$$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} : (\mathbf{x_i}, \mathbf{x_j}) \mapsto \boldsymbol{\varphi}(\mathbf{x_i})^{\mathbf{T}}\boldsymbol{\varphi}(\mathbf{x_j}), \tag{5.4}$$

so that, by substituting (5.4) into (5.3), the dual of the SVM formulation becomes the following QP optimization problem:

$$\begin{aligned} \max_{\alpha} \ & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j K(\mathbf{x_i}, \mathbf{x_j})\,\alpha_i\alpha_j \ + \ \sum_{i=1}^{N}\alpha_j \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{N}\alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{5.5}$$

Thus, the *kernel trick* enables reformulating the dual SVM problem, without using the mapping function $\varphi(\mathbf{x_i})$ explicitly, utilizing the kernel function $K(\mathbf{x_i}, \mathbf{x_j})$ instead, thereby solving the aforementioned issues.

5.3.3 Kernel functions

Kernel functions are defined as the inner product of pairs of mapped data points, which can be interpreted as a measure of similarity between these mapped points. Thus, when using the kernel trick to design kernel functions $K(x_i, x_j)$ directly, avoiding the definition of a mapping function $\varphi(x_i, x_j)$, they should express a measure of similarity between data points in feature space. Constructing a kernel function thus is usually easier than it is to define a corresponding, potentially infinite-dimensional, mapping function $\varphi(\mathbf{x_i})$. Care needs to be taken, however, to construct kernel functions that can be decomposed as an inner product of mapping functions, as specified in (5.4). Mercer's theorem[52] provides a sufficient condition for this to be the case, namely that $K(\mathbf{x_i}, \mathbf{x_j})$ be positive semi-definite. By requiring the kernel function to be positive semi-definite, (5.5) additionally becomes a convex optimization problem, for which efficient solving techniques exist.

The kernel functions used in the remainder of this chapter are:

- The linear kernel, which is defined as:

$$K_{linear}(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i^T x_j} = \mathbf{x_i} \cdot \mathbf{x_j},$$

  which is the inner product of its inputs. The corresponding mapping $\varphi(\mathbf{x_i})$ is the identity function. Using this kernel, the SVM problem reduces to the linearly separable case. This kernel will only provide good results if the process generating the data behaves linearly.

- The formula for the Radial Basis Function (RBF) kernel is provided by:

$$K_{rbf}(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|}{2\sigma^2}\right).$$

  RBF kernels are universal kernels[53], meaning that, with proper regularization and sufficient training data, they can be made to approximate any arbitrary function. Thus, for data sets generated by an unknown non-linear process, the RBF kernel constitutes an appropriate choice.

5.3.4 Least-Squares Support Vector Machines

LS-SVM are a relatively recent development, derived from SVM. They were developed by Suykens and Vandewalle[67], and saw their initial publication in 1999. By modifying the formulation of the error terms, instead of the dual

problem being a QP problem, as is the case for standard SVM, it becomes an elegant matrix equation, for which efficient solvers are readily available.

Assume a training set $\{(\mathbf{x_i}, y_i) \in \mathbb{R}^n \times \{-1, +1\}\}_{i=1}^N$. Starting from the non-linear SVM formulation of (5.2), the error terms are squared, and the inequalities are replaced by equalities, resulting in the following primal problem:

$$\begin{aligned} &\min_{\mathbf{w},b,\mathbf{e}} \ \frac{1}{2}\mathbf{w^T w} + \gamma \sum_{i=1}^{N} e_i^2 \\ &\text{subject to} \quad y_i \left(\mathbf{w^T} \boldsymbol{\varphi}(\mathbf{x_i}) + b\right) = 1 - e_i, i = 1, \ldots, N. \end{aligned} \tag{5.6}$$

As for regular SVM, this can be converted to a dual problem, which, in the case of LS-SVM, reduces to the following:

$$\begin{aligned} &\left[\begin{array}{c|c} 0 & \mathbf{y^T} \\ \hline \mathbf{y} & \Omega + I/\gamma \end{array}\right] \left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{1_v} \end{array}\right] \\ &\text{with} \qquad \Omega = [y_i y_j K(\mathbf{x_i}, \mathbf{x_j})]. \end{aligned} \tag{5.7}$$

Thus, while regular SVM requires solving a QP optimization problem, LS-SVM's dual problem produces a matrix equation, which is computationally considerably less intensive, and for which efficient solvers exist. This results in shorter training times for LS-SVM compared to SVM. On the other hand, LS-SVM models lack the sparsity of SVM models. Using LS-SVM models instead of SVM models may therefore slightly increase the time required for making predictions.

5.4 Model evaluation

Classification models can only be used as diagnostic models if they have sufficient predictive performance. In other words, they should be sufficiently able to correctly predict the outcomes for previously unseen input data. Given a test data set, separate from the training data, for which both input data and outcomes are known, several measures exist to evaluate a model's predictive performance[56]. This subsection defines some of these.

In case of a binary classifier, both the actual and predicted outcomes take on two possible values, which, in the case of diagnostic models, usually correspond with presence or absence of disease. To keep the discussion general, These will be labelled "positive" and "negative", respectively. A classifier may predict outcomes correctly or incorrectly, corresponding with "true" and "false" outcomes. For any particular input data point, four combinations are thus

		Actual outcome	
		positive	negative
Predicted outcome	positive	true positive (TP)	false positive (FP)
	negative	false negative (FN)	true negative (TN)

Table 5.4 – Contingency table of actual versus predicted outcome.

possible, as indicated in the contingency table from Table 5.4: a positive outcome predicted as positive (TP), a positive outcome predicted as negative (FN), a negative outcome predicted as positive (FP), or a negative outcome predicted as negative (TN). By counting the number of occurrences of each such combination in the predictions, made by a model for a test data set, the following performance measures can be defined:

$$\begin{aligned}
\text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \\
\text{sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\
\text{false negative rate (FNR)} &= \frac{\text{FN}}{\text{TP} + \text{FN}} \\
\text{specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}} \\
\text{false positive rate (FPR)} &= \frac{\text{FP}}{\text{TN} + \text{FP}}
\end{aligned}$$

Of these measures, accuracy is the most intuitive one, as it simply represents the ratio of correctly predicted data points. This metric has several drawbacks, however[60, 37, 44]. For highly unbalanced data sets, for example, it is not very informative. Sensitivity expresses the ratio of positive data points correctly identified as such, whereas specificity constitutes the ratio of negative data points correctly predicted. Thus, diagnostic tests, which are used in cases of suspected, life-threatening disease, require high sensitivity to avoid missing affected patients (type II error). By contrast, for a screening test, to be applied to the general population, high specificity is of prime importance, to avoid too many interventions based on false alarms (type I error). For a specific application, ideally, the relative risks associated with type I and type II errors are known, enabling the optimization of models based on measures that assign different costs to these risks[35, 82, 18, 56].

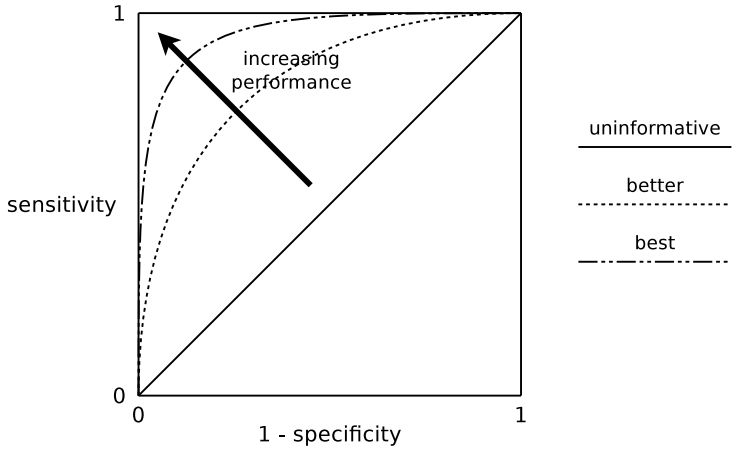


Figure 5.2 – General form of ROC curves. The closer they reach 100% sensitivity and specificity, corresponding with the $(0, 1)$ coordinate in the figure, the better the models.

For a general performance evaluation of classification algorithms, no assumptions about the values of these relative risks are possible, thus requiring different performance measures. Neither sensitivity nor specificity are sufficient by themselves. Moreover, they suffer from the fact that their value depends on the threshold used to distinguish positive from negative predictions. This can be seen by plotting sensitivity and specificity as the model's classification threshold is varied. Graphed in a plane formed by axes representing $\text{FPR} = 1 - \text{specificity}$ and sensitivity, respectively, this produces a Receiver Operating Characteristic (ROC) curve, such as in Figure 5.2. ROC curves of better models will approach the $(0, 1)$ coordinate closer. The latter corresponds with higher Area under the ROC curve (AUC), which will therefore be used as a measure of predictive performance. Since it is calculated from various sensitivity-specificity trade-offs, resulting from different classification thresholds, the Area under the ROC curve (AUC) will provide a summary measure of a model's behaviour, independent of the classification threshold used, and is therefore commonly used for comparing predictive performance of classification algorithms.

5.5 Learning curves

Apart from the classification algorithm used, sample size is one of the primary factors influencing a model's predictive performance. In order to obtain a better

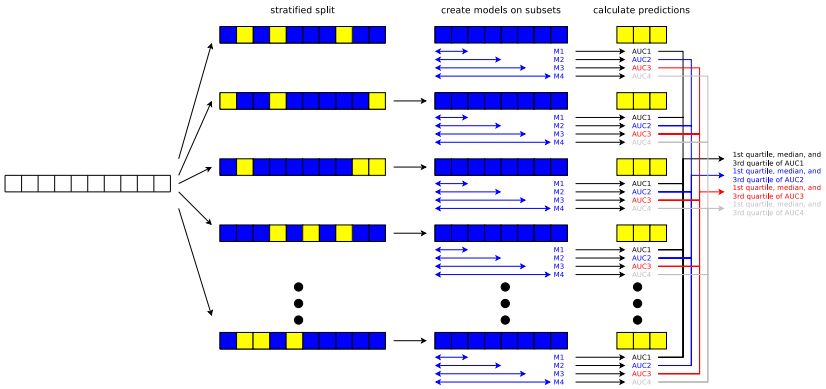


Figure 5.3 – Workflow used for obtaining learning curves, graphing AUC, sensitivity, specificity, and accuracy with regard to sample size.

understanding of the relative behaviours of different classifiers, it is interesting to measure their predictive performance with respect to sample size, using learning curves. Additionally, since classifiers exhibit a certain variability in their results, for a comparison of classifiers to be valid, one should take this variability into account.

These two considerations led to the machine-learning workflow from Figure 5.3. Starting from the full set of $N = 3511$ data points, I generated fifty splits, each consisting of 70% training and 30% test data, randomly stratified to the binary outcome variable. For each such split, I drew samples from the training set, of sizes increasing by 50 data points, which served to generate models. Testing these models against the corresponding test data set then allowed to produce learning curves for accuracy, sensitivity, specificity, and AUC. Combining results from the 50 training-test splits allowed to obtain median values and interquartile range (IQR) for these respective learning curves, providing a more complete insight in the relative performances of different algorithms.

5.6 Analysis

I generated learning curves, as described in the previous section, for different combinations of classification algorithms, applied to the data set described in Section 5.2, both using raw features, as well as derived features, obtained intuitively or through preprocessing. The classification algorithms were logistic regression and LS-SVM, the latter both with linear and RBF kernels. For each combination, I calculated accuracy, sensitivity, specificity, and AUC.

For brevity, I only present results for AUC in this chapter, which, as stated in Section 5.4, provide a useful summary result of a classifier's predictive performance, independent of the classification threshold used. I list the main results in this section.

First, comparing performance of the three different classifiers, as applied to the raw IOTA data, Figure 5.4 clearly shows LS-SVM's considerable superiority at low sample sizes, using either linear or RBF kernels, compared to logistic regression. At these sample sizes, both linear and RBF kernels perform similarly. LS-SVM's superiority is likely due to its built-in regularization mechanism, which counteracts overfitting when samples are small with respect to the number of variables. As sample sizes grow, and performance improvements for growing sample sizes become more modest, the performance gap between logistic regression and LS-SVM with linear kernel diminishes, as both essentially behave linearly. Under these conditions, LS-SVM with RBF kernel outperforms the other two classifiers, since, given sufficient data points, it is able to capture the data's non-linearity, which linear models cannot. The learning curves from Figure 5.4 thus convey a comprehensive picture of the relative performances of LS-SVM and logistic regression for a range of sample sizes, explaining reports both of similar performance for large samples[76, 72, 74, 77], as well as of LS-SVM outperforming logistic regression for low sample sizes[59, 50]. In either case, without additional knowledge about which part of the learning curve a certain sample size belongs to, LS-SVM generally provides performance as good as, or better than, logistic regression, if used in conjunction with an RBF kernel.

The second plot, from Figure 5.5, clearly answers the main question of this chapter. It compares performance of logistic regression, applied to the preprocessed data set, with that of LS-SVM, applied to the raw data, with both linear and RBF kernels. Here, at low sample sizes, LS-SVM with linear kernel, applied to the raw IOTA data without derived features, exhibits similar performance as logistic regression, applied to the preprocessed data with the derived features. LS-SVM with RBF kernel, applied to the raw data, meanwhile, outperforms both. At low sample sizes, the use of LS-SVM thus obviates the need for the time-consuming, manual analysis for defining the derived variables from Table 5.3. At high sample sizes, however, logistic regression is able to take advantage of these derived variables, surpassing performance of LS-SVM with linear kernel, applied to a data set without these extra variables. However, it is not able to surpass LS-SVM's performance on raw data when an RBF kernel is used.

Figure 5.6 shows some of these same comparisons for training sample sizes of 250 and 2450 patient entries. This shows that, when applied to 250 raw data points, both LS-SVM with linear and RBF kernels outperform logistic regression, whether the latter be applied to the raw or preprocessed data sets.

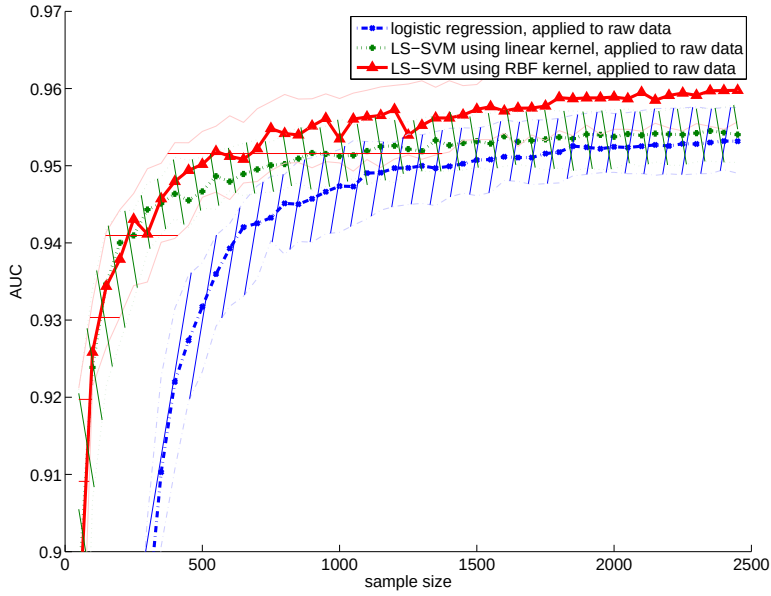


Figure 5.4 – AUC learning curves obtained from applying logistic regression, LS-SVM using a linear kernel, and LS-SVM using an RBF kernel, to the raw IOTA variables listed in Table 5.1. Thick lines indicate median values, with the region around them showing IQR.

At a sample size of 2450 data points, when applied to the raw IOTA data, both generalized linear models – logistic regression and LS-SVM with a linear kernel – get similar performance, with both outperformed by LS-SVM with RBF kernel. By contrast, when applying logistic regression to the preprocessed data, it takes advantage of the derived variables to reach similar performance as LS-SVM with RBF kernel applied to the raw data set.

Finally, the similarity of the two curves in Figure 5.7, of LS-SVM with an RBF kernel applied to the raw and preprocessed data, respectively, shows that LS-SVM is well capable of modelling the data's non-linearity, requiring neither the addition of intuitively derived variables, nor complex prior preprocessing involving linearization and introduction of second-order effects.

Thus, when aiming for maximum AUC, without additional prior knowledge, LS-SVM with an RBF kernel constitutes a safe choice in all situations. It provides the added advantage of not requiring any preprocessing to obtain this performance, which can otherwise be a complex, time-consuming, and in large

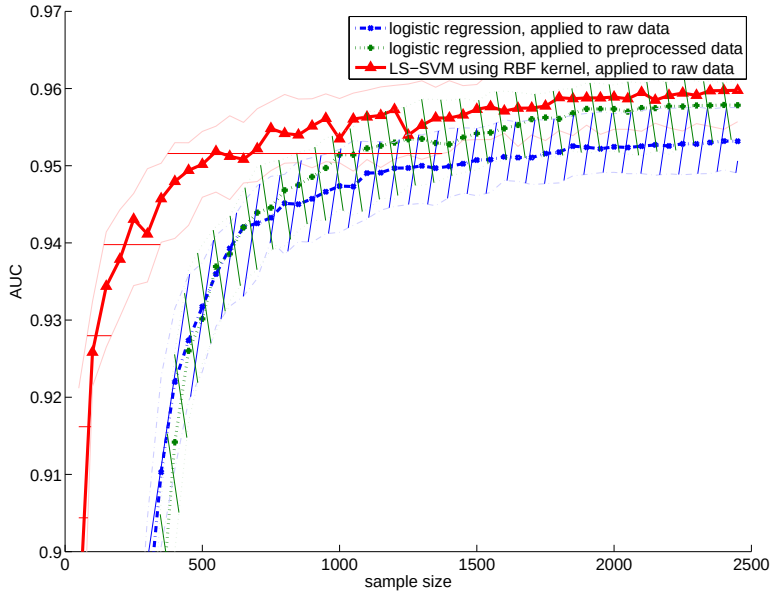


Figure 5.5 – AUC learning curves obtained from logistic regression applied to the preprocessed IOTA variables from Table 5.2, and from LS-SVM using linear and RBF kernels applied to the raw IOTA variables listed in Table 5.1. Thick lines indicate median values, with the region around them showing IQR.

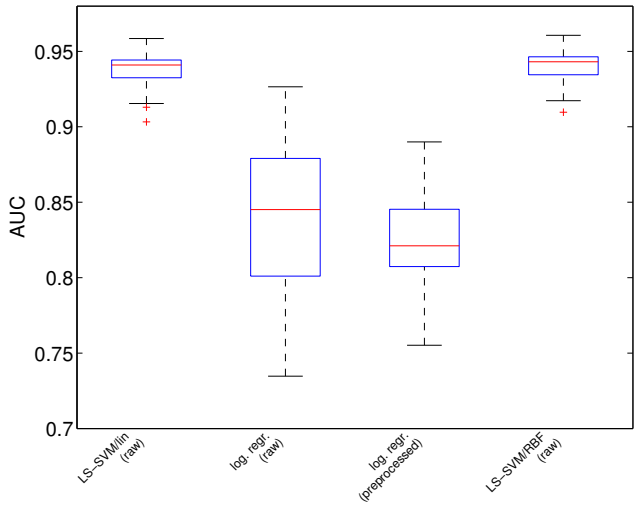
part manual process.

5.7 Conclusion

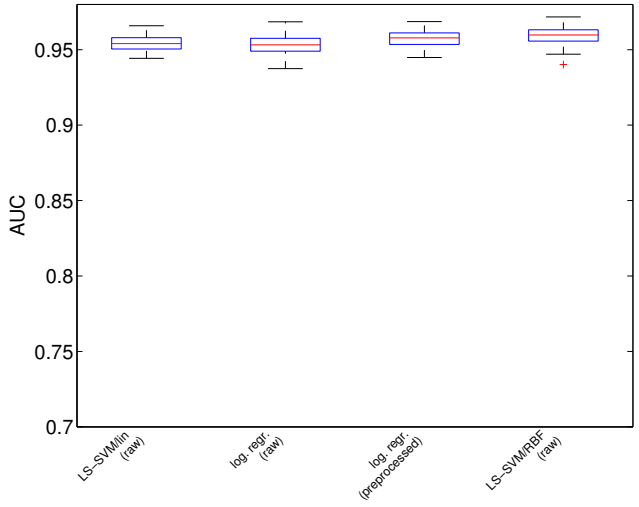
The addition of derived variables, obtained either intuitively or by complex data preprocessing, are difficult to automate, but they are necessary steps for obtaining high-quality predictive models when using logistic regression.

The results of this chapter, however, show that, for the IOTA data set, more sophisticated classification algorithms, such as LS-SVM, which is capable of modelling non-linear effects, can produce high-quality models directly from raw data, without the need for complex preprocessing or addition of intuitively derived variables. This considerably simplifies the machine-learning workflow.

More research is needed to confirm that these results generalize to other data sets



(a) @ 250 patient entries



(b) @ 2450 patient entries

Figure 5.6 – AUC values for several combinations of classifiers and feature sets, for different training set sizes.

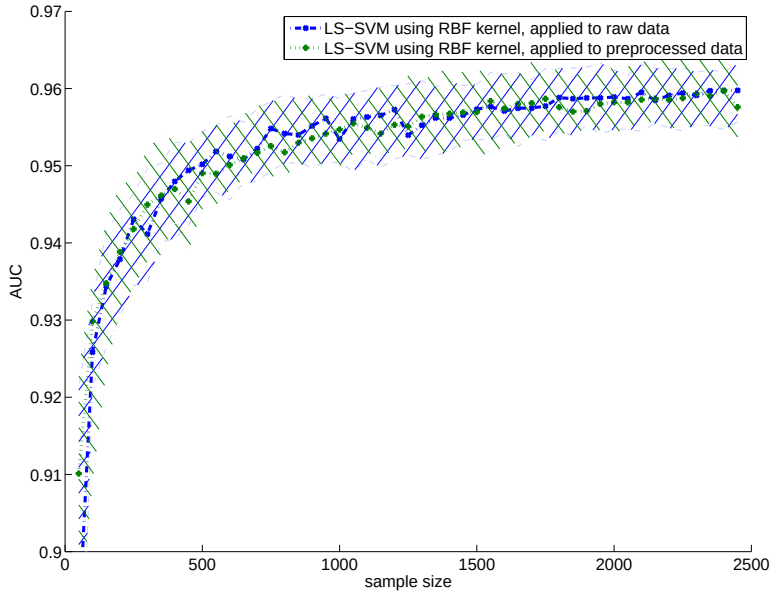


Figure 5.7 – The two learning curves in this plot show the AUC attained by LS-SVM with an RBF kernel applied to the raw and preprocessed data sets, respectively. The similarity of the results indicates that LS-SVM with an RBF kernel is sufficiently capable of modelling any non-linearities, so that it does not require any additional preprocessing step.

and to other machine-learning algorithms capable of modelling non-linearities. Therefore, I plan similar analyses using the International Endometrial Tumour Analysis (IETA) data set, and other sophisticated classifiers. However, given the nature of the machine-learning algorithms involved, I expect these results to generalize to other data sets.

Benchmarks comparing performance of classification algorithms typically focus on a single sample size, thereby leading to conflicting reports caused by different sample size choices. By contrast, learning curves and their interquartile range (IQR) provide a more comprehensive overview of the relative strengths and weaknesses of algorithms. Moreover, they enable study coordinators to assess when data collection should be terminated. For these reasons, the components developed in Chapter 6 include methods that simplify the generation of such learning curves.

I have omitted feature selection from this discussion. In order to be usable in

clinical practice, however, diagnostic models need to use as few variables as possible. Full automation of the machine-learning workflow used in medical diagnostic modelling thus requires the automation of feature selection as well. Appendix C lists the results of a few initial experiments with common feature selection algorithms, which a priori assign equal cost to all variables. For diagnostic models, however, this automated feature selection requires feature selection techniques that take into account variables' costs, whether they be objective, financial costs, or subjective, relating for example to the level of discomfort for the patient. While I plan more research into such techniques in the future, I concentrated solely on the predictive performance attained by the classification algorithms themselves for this analysis.

Obviously, classifiers capable of modelling non-linear effects produce models that cannot be interpreted as readily as logistic regression models. Therefore, and in order to avoid costly preprocessing, I propose to always generate two models, derived directly from raw data: one with lower predictive performance, generated by an algorithm such as logistic regression, providing insight into the primary features influencing diagnosis; and another, generated by a sophisticated algorithm producing complex non-linear models, such as LS-SVM combined with an RBF kernel, to be used for predicting diagnoses for incoming patients.

In conclusion, the elimination of the need for complex preprocessing and addition of intuitively derived variables thus opens up the possibility of extensive automation of the machine-learning workflow, paving the way for highly integrated software frameworks that include components both for data collection and machine-learning modelling. I will discuss the building blocks for such a tightly integrated software framework in the next chapter.

Chapter 6

Data analysis integration

Electronic Data Capture (EDC) has considerably improved efficiency of data collection in the context of clinical studies, with fewer errors, shorter delay between start and end of studies, and lower costs as a result. I believe clinical studies will benefit further from the integration of EDC and data analysis into a single software framework, such as Clinical Data Miner (CDM).

The previous chapter showed that sophisticated machine-learning algorithms can be applied without a prior complex preprocessing step, simplifying automation of the machine-learning workflow. Therefore, the CDM framework integrates several APIs for implementing this machine-learning workflow. This chapter describes the design of these APIs, and how they can be used interactively from within the *Jython* programming language interpreter to provide a powerful experimentation platform.

6.1 Introduction

Figure 6.1 lists the steps of a typical machine-learning workflow, as used in clinical diagnostic modelling. Currently, most steps are performed manually, resulting in an error-prone, time-consuming process.

The identification and removal of redundant or non-informative variables, as well as the linearization of non-linear features, form the biggest hurdles to automation of this workflow. An example of this can be observed from [3], which describes this process as applied to data from the International Ovarian Tumour Analysis (IOTA) consortium. Such steps contribute considerably to the complexity of

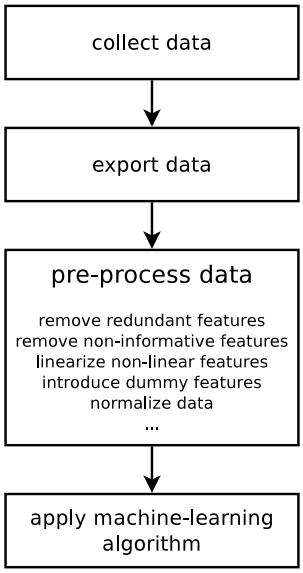


Figure 6.1 – Typical machine-learning workflow for clinical diagnostic modelling.

the machine-learning workflow, which can therefore not readily be automated. They are nevertheless required for obtaining acceptable predictive performance from traditional machine-learning algorithms such as logistic regression.

Chapter 5, however, has demonstrated that more sophisticated classification algorithms, such as Least-Squares Support Vector Machines (LS-SVM), are capable of attaining comparable performance directly from raw, unpreprocessed data. Thus, these more sophisticated algorithms enable the implementation of drastically simplified workflows, opening up the possibility of automation.

The goal of the CDM project therefore is to integrate patient data collection and machine-learning into a single software framework, in order to simplify data analysis. This chapter describes the software components that I have developed towards furthering this goal.

The organization of this chapter is as follows: The first sections describe CDM's APIs for data access, data preprocessing, and machine-learning, enabling the steps listed in Figure 6.1. This is followed by a chapter describing an API implementing some basic statistical functionality, for use in inter-rater agreement studies. As for the development of the EDC component, these APIs were developed in a TDD fashion, ensuring software quality. The chapter continues with a description of a few *Jython* modules, which enable interactive use of

CDM's APIs. This interactivity facilitates very flexible experimentation. The chapter ends with a few conclusions.

6.2 Data access

Background

Typical machine-learning workflows use separate software packages for data collection and data analysis. This requires a means to extract data from the data collection software and feed them into the data analysis software. To that end, data are exported in an intermediary format. Often used are Microsoft® Excel® or Comma-Separated Values (CSV) formats. This conversion, however, can introduce errors. Dates may be incorrectly converted to numbers, text containing three separate numbers may be interpreted as dates, etc. This leads to the need for manual verification of the exported data, making this step potentially very time-consuming.

By contrast, CDM's integration of EDC and machine-learning components obviates the need for exporting data to files, enabling direct communication between these two components instead, using a data access API accepting data queries.

Design

The classes involved in this representation are shown in the UML diagram from Figure 6.2. A call to `DataManager.newDescriptor`(), shown in Figure 6.3, produces a `DataDescriptor` object, which, instead of containing the data itself, holds the necessary information for loading it at a later time, by means of the `DataDescriptor.load`() method. As will be elaborated in the next section, by allowing a number of manipulations to this data loading strategy, many preprocessing steps can operate on such `DataDescriptor` objects without requiring any database access or data copying.

When their `DataDescriptor.load`() method is called, a `Data` object is returned. This object provides a means of iterating over `DataPoint` objects, which represent individual patient entries. These entries can be queried for the values of various fields, with `Data.getFields`() providing a collection of fields contained in the `DataPoint` objects.

The interfaces `DataDescriptor`, `Data`, and `DataPoint`, representing unlabelled data, are further extended by interfaces `LabelledDataDescriptor`,

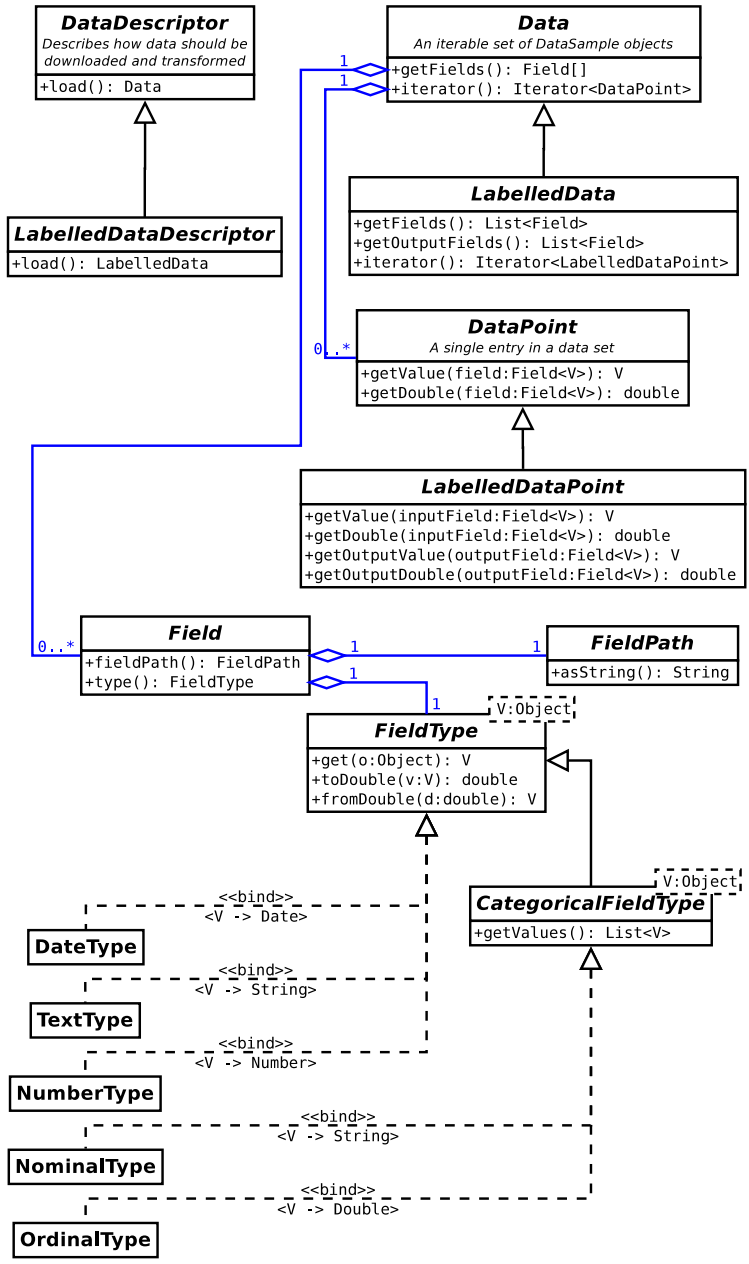


Figure 6.2 – Simplified `uml` representation of the classes involved in CDM's internal representation of study data.

`LabelledData`, and `LabelledDataPoint`, respectively. The latter represent labelled data and are intended for use by supervised learning methods.

6.3 Data preprocessing

Background

Generally, data have to undergo some basic preparatory steps prior to their analysis by machine-learning algorithms. This can include removal of fields, such as optional or free text fields, which only provide useful information for large samples, and after specific preprocessing. Or it can include the conversion of nominal and/or ordinal variables to dummy variables.

Many preprocessing steps apply to all variables of a particular type. This preprocessing can be performed either manually, which is time-consuming, or automatically, based on heuristics. The *R* language, for example, when loading a CSV-formatted file, will analyze the values of a column to determine their type. Both manual and heuristics-based approaches, however, are error-prone: the former may lead to human error, while in the second, heuristics may lead to incorrect guesses, interpreting numeric variables as ordinal ones or vice-versa, for example.

Moreover, without close integration between the EDC and data analysis components, the data analysis cannot possibly take into account the data's structure. As has been mentioned in Subsection 3.3.1, CDM enables conditional inclusion of questions in Case Report Forms (CRFs). The inclusion of such questions depends on the answer to the questions they depend on, leading to a hierarchical questionnaire structure, such as in Figure 6.8. Such a structure will inevitably produce data points with structurally missing variables. Without integration between data collection and data analysis components, the latter will have no information about a CRF's hierarchical nature, hence will not be able to treat structurally missing variables specially.

By contrast, CDM's machine-learning component, thanks to its integration with the EDC component, has full access to a CRF's metadata. It therefore can avoid error-prone manual or heuristics-based preprocessing, and instead rely on accurate knowledge of both the types of variables, as well as the questionnaire's structure.

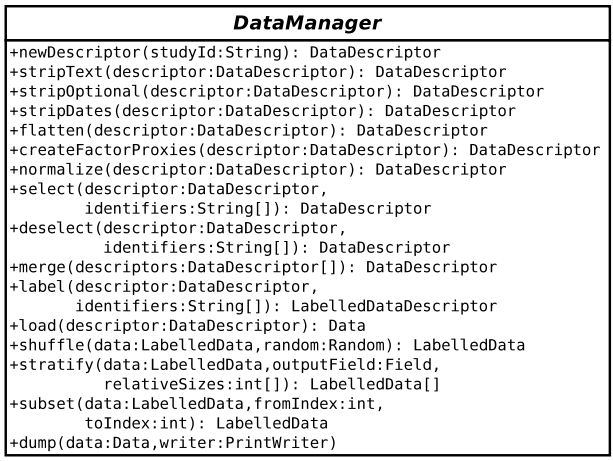


Figure 6.3 – The `DataManager` interface provides access to data and enables preprocessing, by including methods for the creation and manipulation of `DataDescriptor` and `Data` objects.

Data presentation transformation

Most of CDM's preprocessors operate on `DataDescriptor` objects. They are applied by invoking methods of the `DataManager` interface, which exhibits a *Facade* design pattern[28]. This interface is presented in Figure 6.3.

Several preprocessors remove features that would otherwise complicate subsequent application of machine-learning algorithms. Method `stripText`(), for example, removes any feature that consists of free form text. While machine-learning techniques exist for processing free text[26, 11, 17], they are likely of little use for the clinical studies handled by CDM, in which sample sizes are typically limited, and in which free text is generally only used to describe patient situations which cannot adequately be captured by other questions present in the CRF.

Features that are optional likely include missing values, which many machine-learning algorithms cannot readily handle. This can be handled by *case deletion*, in which data points with missing values are deleted list-wise. An alternative is to use *imputation*[62], in which missing values are replaced with substituted values. The approach of CDM's preprocessor is to delete the optional variable, column-wise, and is implemented by `stripOptional`().

The method `stripDates`() removes date features. Date values of themselves are not informative: only values calculated from dates, such as time durations

may provide insight. Since CDM does not currently support date calculus, it provides a `stripDates`() method, removing any date fields.

Another class of preprocessors allow to select or deselect variables, which can either target specific variables, or use patterns to respectively include or exclude entire sections of variables. The `select`() method keeps the variable(s) specified by the supplied parameter, and discards the others, while `deselect`() discards the specified features and keeps the rest. The `label`() method combines the functionality of both, by discarding the specified variables from the input data, and using them as output data instead.

In order to be able to combine the data from several studies that contain some overlapping sections, `merge`() will combine data points from these studies, only keeping fields from sections present in all the provided studies. This can be used, for example, to combine International Endometrial Tumour Analysis (IETA) #1a, #1b and #1c, which are essentially the same studies, but for which some participants completed a more elaborate questionnaire than others.

Since numeric machine-learning algorithms cannot handle categorical variables directly, CDM's preprocessing API provides the `createFactorProxies`() method to convert a categorical variable into a set of dummy variables.

Further, since most classifiers use some distance measure to calculate (dis)similarity between data points, having features with a larger range than the others would cause them to dominate a classifier's objective function. In order to avoid this, data are typically normalized prior to the application of machine-learning algorithms. In CDM, the `normalize`() implements this functionality.

Finally, `flatten`() deals with the hierarchical structure of CRFs and associated structurally missing data. Its approach is to provide default values for structurally missing values. These default values depend on the variable type, as indicated in Table 6.1. In the future, I will implement similar methods that use more sophisticated approaches for determining these default values.

Data point reordering & selection

Apart from preprocessors operating on `DataDescriptor` objects, Figure 6.3 additionally lists preprocessors operating on `Data` objects. The former are used for transformations that modify the data's presentation, and may be part of a chain of transformations, the intermediate results of which will not be used. Therefore, they avoid loading data into memory. By contrast, the preprocessors mentioned in this section are used for transformations that do not modify how the data are presented, but re-arrange the data instead.

Feature type	Default value
free text	“”
number	−1
date	Unix epoch
ordinal variable	−1
nominal variable	“_NA_”

Table 6.1 – Default values used by `DataManager.flatten`() for the different feature types.

They consist of the methods `subset`(), `shuffle`(), and `stratify`(). As the name implies, `subset`() returns a subset of the input data starting and ending at the specified indices. The second method, `shuffle`(), returns a random permutation of the original data. Its second parameter enables supplying a random number generator, which may be initialized with a specific number for reproducibility purposes. Finally, `stratify`() is trivially implemented by combining the functionality of the first two methods, and split the input data set in several distinct sets, the relative sizes of which are supplied by the `relativeSizes` array input parameter. If the parameter `relativeSizes` is set to $[70, 30]$, the method produces training and validation sets, with 70% and 30% of data points, respectively. For relativeSizes = $[10, 10, 10, 10, 10, 10, 10, 10, 10, 10]$, data sets for 10-fold cross-validation can be created.

Design

One of the design considerations was to avoid unnecessary memory copies for chained data transformations. To that end, transformations do not operate on objects containing the data themselves, but instead they operate on objects keeping track of which data should be loaded and how.

These objects are of class `DataDescriptor`, and are presented in Figure 6.4. They keep track of *which* data to load by means of a `SampleIterable` object, while a `SectionInfoMap` registers *how* data should be transformed. Additionally, a `DataPointFilter` object determines which data points should be filtered out.

In this design, preprocessors correspond with `DataTransformStrategy` objects. These objects transform *how* data are formatted, by modifying a `DataDescriptor`'s `SectionInfoMap`. The latter contains a `FieldInfo` object for each question, of each of a study's sections. These `FieldInfo` objects determine how a CRF question maps to fields, through their `toFields`() method. When a descriptor is created using `DataManager.newDescriptor`(), the `FieldInfo`

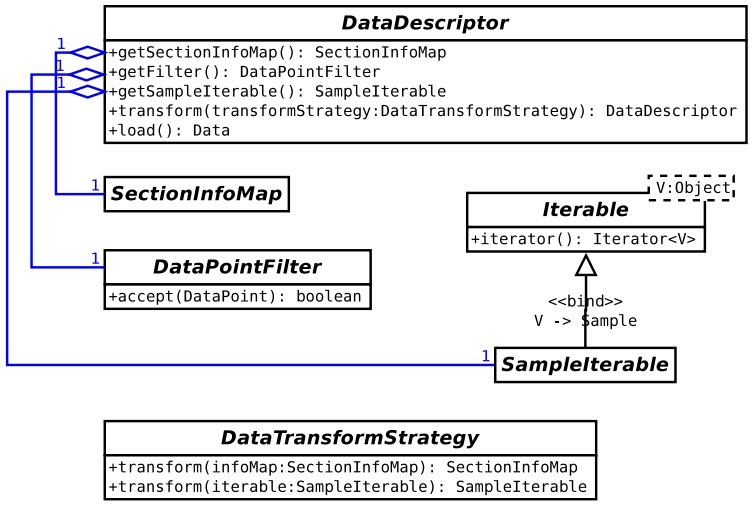


Figure 6.4 – Most preprocessors operate on `DataDescriptor` objects, which do not contain the data themselves, but instead hold `SampleIterable`, `DataPointFilter`, and `SectionInfoMap` objects, which respectively describe which data should be loaded, which should be filtered out, and how they should be presented.

objects initially created are `FieldInfoImpl` objects, initially establishing a one-to-one mapping between CRF questions and fields. The relationship between `SectionInfoMap` and the different possible implementations of the `FieldInfo` interface is shown in Figure 6.5.

Preprocessors first clone the `SectionInfoMap`, to ensure that the original `DataDescriptor` object remains unchanged and can still be used for other purposes. They then remove individual `FieldInfo` objects, to discard certain fields, or replace them with other implementations of the `FieldInfo` interface, to modify the mapping between CRF questions and data fields. These `FieldInfo` objects use a *Decorator* design pattern[28], keeping a reference to the original `FieldInfo` object. When their `toFields`() method is invoked, they delegate this call to the original object, and manipulate the result before returning it. As an example, `DesignVariablesFieldInfo.toFields`(), calls its `delegate`'s `toFields`() method, and replaces any ordinal fields in the return value with dummy fields. This design allows preprocessors to be applied sequentially, progressively modifying how data are represented, without the need for actually loading data.

`DataPointFilter` objects specify which data points should be accepted, and which should be filtered out. The design of these objects follows the *Composite*

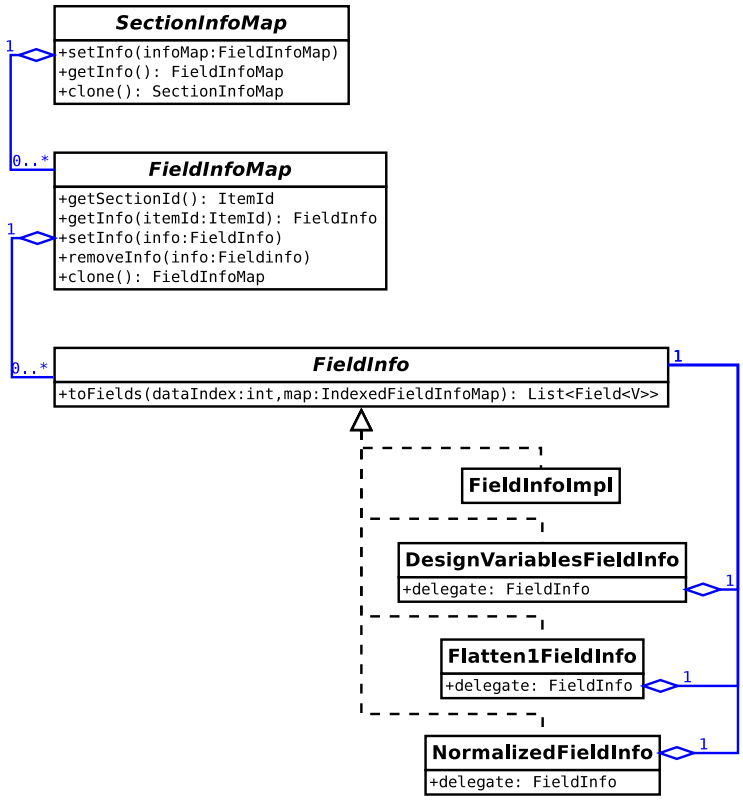


Figure 6.5 – Representation of the relationship between `SectionInfoMap` and the different implementations of the `FieldInfo` interface. For clarity, `FieldInfo`'s template arguments, as well as some methods, are omitted.

design pattern[28] presented in Figure 6.6, allowing to combine filters using "and", "or", and "not" operators.

Finally, the `SampleIterable` interface exhibits a *Composite* design pattern[28] as well, demonstrated in Figure 6.7. `DataManager.newDescriptor`() creates a `StudySampleIterable` object, which, using lazy initialization, defers the loading of the study data sample to when it is needed. `DataManager.merge`() constructs a `CombinedSampleIterable` object, which delegates as necessary to the `SampleIterable` objects from the `DataDescriptor` objects that need to be merged.

When their `load`() method is called, `DataDescriptor` objects produce `Data` objects. The `SampleIterable` queries the database; the `DataPointFilter`

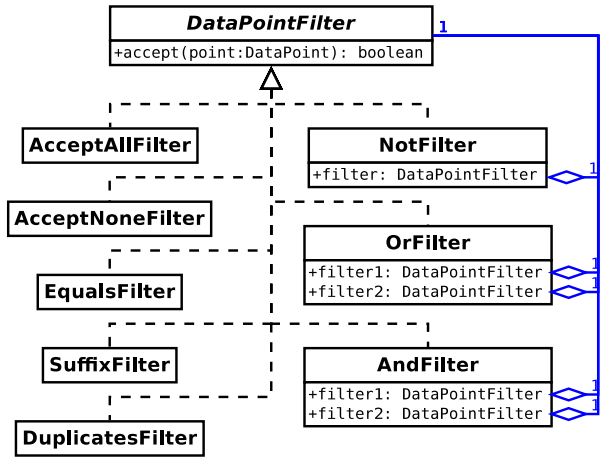


Figure 6.6 – The `DataPointFilter` leverages the *Composite* design pattern[28] to enable combining basic filters to more complex filters using "and", "or", and "not" operators.

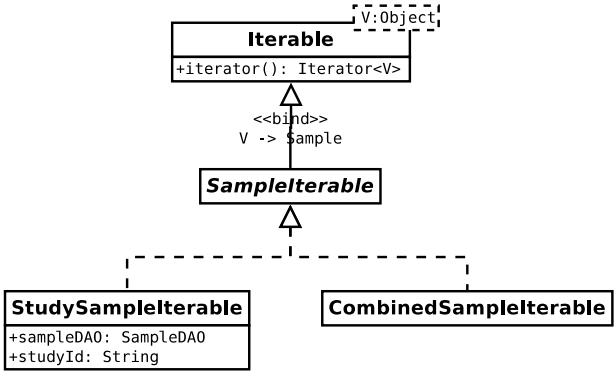


Figure 6.7 – The `SampleIterable` interface provides a means to iterate over patient entries stored in the database. Its default implementation, `StudySampleIterable`, only loads data upon invocation of its `iterator`() method, using a handle to the database and a study identifier. Using the *Composite* design pattern, `CombinedSampleIterable` accepts a list of `StudySampleIterable` objects, enabling iteration over several different studies at once.

decides which data points are included. The `SectionInfoMap` determines which CRF fields to load in the `Data` object, and how they are transformed to `Field` objects.

Advantages

Architecturally, the described design, involving `DataDescriptor` objects, and preprocessors modifying the `DataDescriptor`'s components, provides several advantages. First, it can be easily extended with new preprocessors, possibly implemented by third parties, making the design flexible. Second, preprocessors can be applied sequentially, allowing to combine them to construct complex transformations. Further, by effecting data presentation changes on a cloned version of the supplied `DataDescriptor`'s `SectionInfoMap`, rather than on the original one itself, the original `DataDescriptor` object can still be used. Finally, the design around `DataDescriptor` objects, that do not hold data themselves, enables their manipulation, avoiding the creation of modified copies of the data for each successively applied preprocessor, which could otherwise potentially become very memory-intensive.

Further, the integration of EDC and data preprocessing provides the advantage of enabling automation. Without this integration, the selection of which variables a particular preprocessor should be applied to, either occurs manually, which is cumbersome, or automatically, using a heuristic to determine the types of variables. The type of variable then determines which preprocessors to apply. Both the manual and heuristic-based approaches are prone to errors. By integrating EDC and data preprocessing, on the other hand, CRF metadata, which includes information about variables' types, becomes available during preprocessing, avoiding the need for guessing them.

The availability of CRF metadata during preprocessing additionally enables transformations that would otherwise be impossible, such as that implemented by `DataManager.flatten`(), which converts the hierarchical CRF structure into vectors. For categorical variables that are "children" of a "parent" categorical variable, knowledge about the CRF structure, enables the sequential conversion from hierarchical data into vector data, by means of the `flatten`() method, followed by creation of dummy variables, using `createFactorProxies`(), without the creation of redundant dummy variables. Consider, for example, the questionnaire structure from Figure 6.8. In the absence of information about a questionnaire's structure, automatic, heuristic-based creation of dummy variables will generate six dummy variables, as demonstrated in Table 6.2, two of which are redundant, whereas CDM's preprocessors will correctly identify four dummy variables.

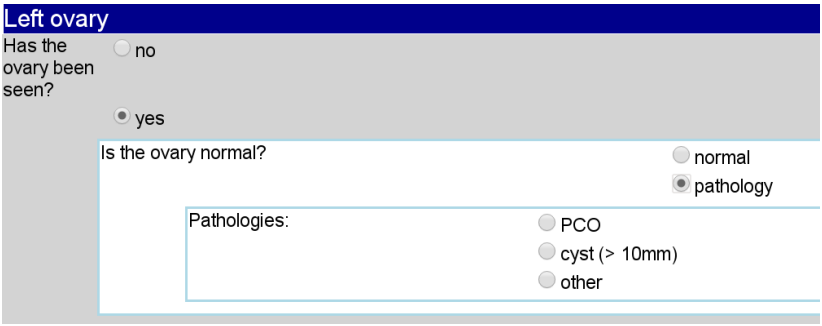


Figure 6.8 – Example of the hierarchical structure of questionnaires. Except for questions at the top of the hierarchy, questions only apply for certain values of their "parent" questions, and will be structurally missing otherwise. (Screenshot of the "Ovaries" section, included in the IETA #1, #3, #4 studies, as presented by the CDM user interface.)

Question	Possible values	Dummy variables	
		no structure	structure
Was ovary seen?	no, yes	yes	yes
Was ovary normal?	s.m., normal, pathology	normal, pathology	pathology
Pathologies	s.m., PCO, cyst, other	PCO, cyst, other	cyst, other
Specify	s.m., *free text*	*free text*	*free text*
# dummy variables		6	4

Table 6.2 – The questions from the hierarchical structure depicted in Figure 6.8 have several possible values. Except for the top question, they can be structurally missing, depending on the value of their parent question. Using automated methods for generating dummy variables, this table shows how many such dummy variables would be introduced, respectively in the absence or presence of structural information about the CRF.

For the example presented here, the amount of redundant variables introduced by a heuristic-based approach is relatively limited. For real-world CRFs, such as from the IETA studies, this number would be much larger, and would not be identified so quickly manually. Obviously, using heuristics, the creation of redundant dummy variables can be avoided by using a non-hierarchical questionnaire instead. However, this would result in very long questionnaires, with many non-applicable questions, considerably complicating data collection.

Apart from enabling the transformation of hierarchical data into vector data, the availability of information about a CRF's structure could enable the definition of kernel functions taking this structure into account, allowing the use of kernel

methods.

In conclusion, the integration of EDC with data access and preprocessing provides many advantages, thereby streamlining the machine-learning process.

6.4 Machine-learning

Background

Several excellent software libraries exist that implement machine-learning algorithms. Some of these implement a wealth of machine-learning algorithms, such as the Waikato Environment for Knowledge Analysis (WEKA) workbench[89, 33] and SHOGUN library[66], while others specialize in a particular algorithm or class of algorithms, such as the LSSVMLab toolbox[16] or LibSVM[9]. The data produced by CDM's data preprocessing API could therefore be perfectly fed to any of these software packages to obtain prediction models, as long as care is taken not to introduce any conversion issues.

However, further integrating machine-learning functionality into CDM opens up new possibilities. Machine-learning models could automatically be updated as new patient entries are registered. Learning curves could be integrated in the framework's user interface, enabling study coordinators to monitor a study's progress. The latter would allow them to make an informed decision when to terminate data collection.

As an initial step towards this goal, I integrated a machine-learning API into CDM.

Design

CDM's machine-learning API consists of a set of interfaces, provided by CDM to interact with classification algorithms and the models they produce, as well as some convenience methods for producing learning curves from a patient data set.

The interfaces for interacting with classifiers are presented in Figure 6.9. For the moment, CDM takes advantage of the abundance of existing machine-learning libraries to avoid the need for implementing its own. More specifically, CDM leverages the existing WEKA library to gain access to a wealth of classification algorithms.

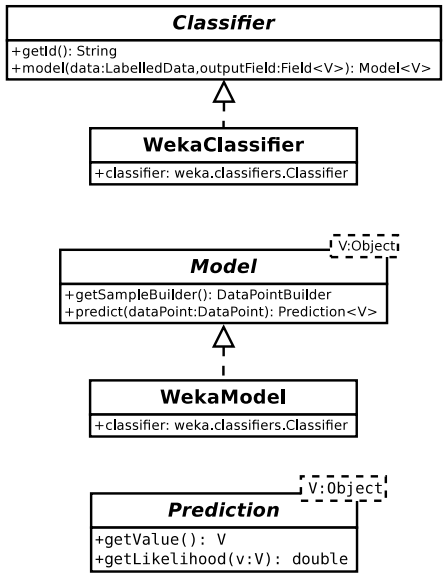


Figure 6.9 – Set of interfaces defined by the CDM software framework for interaction with machine-learning algorithms, and their current only implementation, utilizing the WEKA library. The machine-learning algorithm used can be chosen by supplying `WekaClassifier`'s constructor with a subclass of WEKA's `Classifier` class.

Using the *Adapter* design pattern[28], classes `WekaClassifier` and `WekaModel` implement the CDM `Classifier` and `Model` interfaces respectively, by delegating calls to a particular instance of the WEKA class `Classifier`, as shown in Figure 6.9. Other implementations of the `Classifier` and `Model` interfaces may be added in the future.

CDM's machine-learning API is accessed through the `ClassifierFacade` interface. This interface exposes CDM's machine-learning capabilities using the *Facade* design pattern[28], and is displayed in Figure 6.10. It includes the `newWekaClassifier`() method, for constructing a `Classifier`, using the WEKA machine-learning algorithm implementation specified by the `classifier` parameter. Its `sweep`() method calculates predictive performance on a test set, for each of the data set sizes provided, for a number of different training-test splits, using the same workflow as that depicted in Figure 5.3. Performance measures currently calculated are accuracy, sensitivity, specificity, as well as AUC, but could easily be extended with performance measures that optimize the tradeoff of risks associated with type I and type II errors[35, 82, 18, 56]. The `sweep`() method returns performance data that can be supplied to method

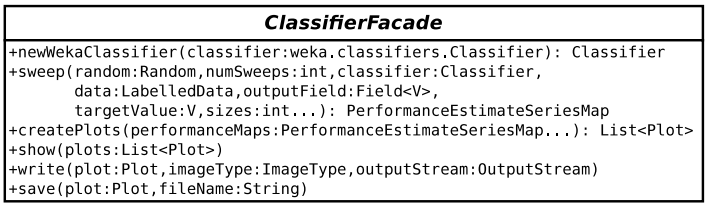


Figure 6.10 – *Facade* interface exposing CDM's machine-learning capabilities.

`createPlots`() for generating learning curves, which can be displayed on screen using `show`(), or written to disk using the `write`() or `save`() methods.

While Monte Carlo simulations, or rules of thumb advising a minimum of ten events per variable[58], provide a guess of the amount of patient data required for obtaining meaningful models, both rely on assumptions: the former assume the model used in the simulation resembles the actual model, while the latter assume all collected variables are relevant. By offering the `sweep`() method to generate learning curves, CDM enables study coordinators to visualize how predictive performance evolves with respect to patient set size. Such plots show if sample size is such that a small increase results in relatively large performance gains, encouraging additional collection of patient data, or, on the contrary, whether large sample size increases are required for barely obtaining modest performance gains, providing an indication for terminating data collection. Due to the difficulty of a priori estimation of required sample size, CDM's functionality for generating learning curves thus provides an important advantage for study coordinators.

6.5 Statistical analysis

The user interface, described in Subsection 4.2.2, which I derived from the CDM framework for supporting inter-rater agreement studies, has been used for six such studies so far. Therefore, I implemented an API for facilitating the analysis of such studies.

As for the machine-learning API, data are prepared using CDM's data access and preprocessing APIs before they are processed by CDM's statistical analysis API. This API enables the calculation of κ coefficients of inter- and intra-rater agreement. For comparing measurements from two raters, or for comparing two measurements from a single rater, Cohen's κ coefficient[10] can be used, while Fleiss' κ coefficient[23] is available for studies with more than two raters. Apart from κ coefficients, percentage agreements can be calculated as well.

CDM's API further provides the jackknife sampling technique[20] for generating samples from the application of a measure to a data set. This enables the calculation of estimates and their variance, as well as the use of statistical tests for comparing inter-rater agreement samples. The latter can for example be used to compare the inter-rater agreement of two sets of images obtained through different ultrasound technologies over the same patient sets.

CDM's API for statistical analysis has been used in the context of several studies, amongst which the IETA #2 study, primarily for comparing the effect of different technologies on data quality.

6.6 Jython interface

Background

CDM's machine-learning capabilities are not accessible through its user interface yet. I therefore developed a set of *Jython* modules to enable the use of CDM's data analysis capabilities, for interactive rapid prototyping and experimentation.

Jython[42] is an interpreter for the *Python*[54] language, running on the *Java Virtual Machine* (JVM)[81]. By running on the JVM, it can provide direct access to *Java* libraries from within the *Jython* language. As with most interpreters, *Jython* cannot only be used for running scripts, but it can also be used interactively. This combination of direct access to *Java* libraries and interactivity make the language ideally suited for experimenting with CDM's data preprocessing and machine-learning APIs.

Design

The infrastructure to provide *Jython* scripts access to CDM's APIs consists of three elements, depicted in Figure 6.11:

- The *Java* class `JythonSupport`, contained in the `cdm-server` module, which uses Spring[87] framework functionality to obtain handles to CDM's different *Facades*, such as `DataManager` or `ClassifierFacade`;
- The *Jython* `cdm.py` module, which creates an instance of the `JythonSupport` class, and uses it to implement a function that enables retrieving CDM's *Facades* from within *Jython*, facilitating access to CDM's API from within *Jython*.

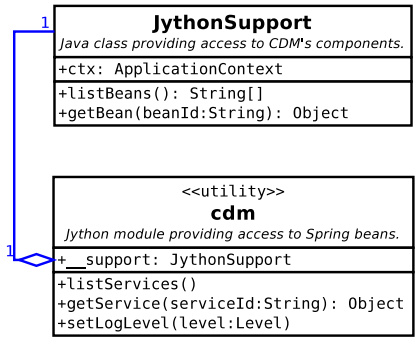


Figure 6.11 – The classes from this UML diagram provide access to CDM's data analysis APIs, from within the *Jython* interpreter. To this end, they leverage Spring's IoC container to provide access to Spring components.

- The *Bash* shell script `startJython.sh`, which initializes the `CLASSPATH` environment variable to ensure *Jython* loads all relevant *Java* libraries.

Jython modules further provide convenience interfaces for interacting with CDM's different *Facade* objects, making CDM's data preprocessing and machine-learning APIs more easily accessible from within *Jython*. These modules can be leveraged from within *Jython* scripts, or they can be used interactively, for rapid prototyping and experimentation. The latter can be very useful for quickly gaining an insight into the effect that certain choices of pre-processors and machine-learning algorithms have on predictive performance of the resulting models.

6.7 Development methodology

As for CDM's EDC component discussed in Chapter 3, I developed CDM's data preprocessing and machine-learning APIs following a TDD development methodology. Since these APIs are part of the `cdm-server` module, which also includes CDM's web server code, test coverage cannot be reported separately from the latter. As reported in Table 3.6 though, the automated test suite for web server code and data analysis APIs combined covers around 92% of source lines, and 90% of branches. These numbers ensure high software quality.

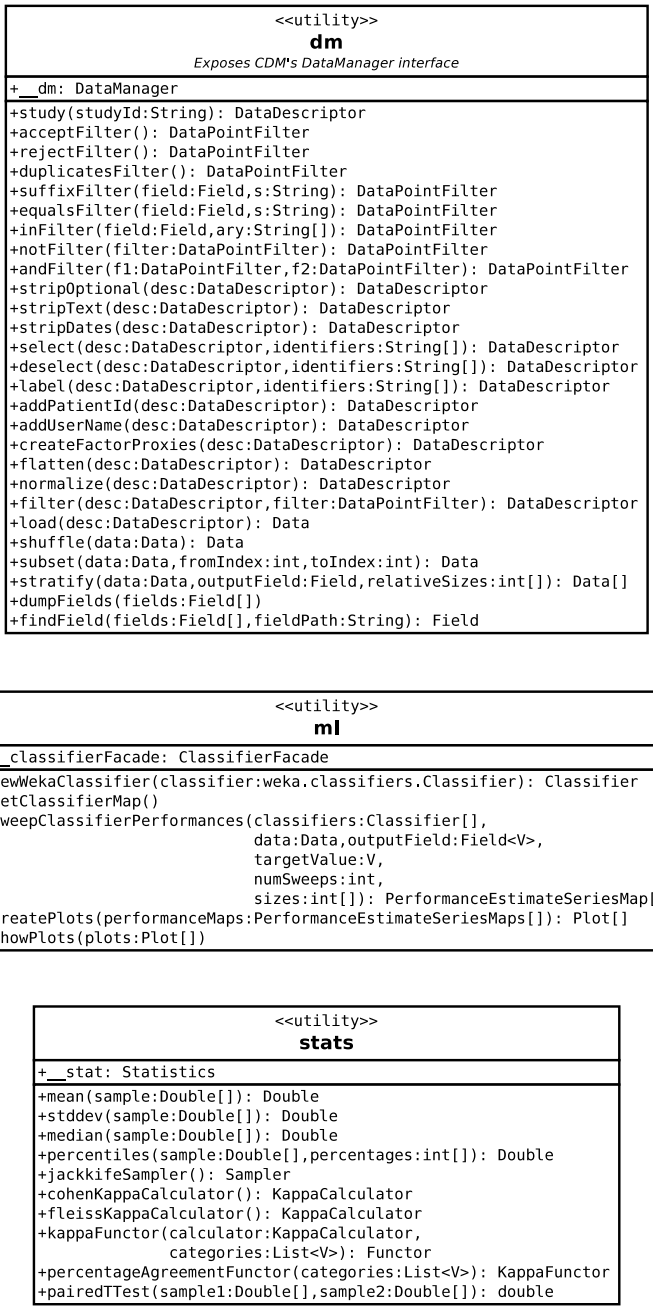


Figure 6.12 – The UML diagrams above list the *Jython* modules providing access to CDM's data analysis facilities. The `dm` module enables the use of data access and data preprocessing capabilities; `ml` provides access to CDM's machine-learning API; while `stats` can be used for calculating some common statistical measures.

6.8 Conclusion

Today, machine-learning workflows in clinical diagnostic research employ separate software packages for data collection and data analysis processes. While such set-ups have well served the medical research community, the separation between these processes requires the use of error-prone processes, requiring manual, time-consuming verification. They include the transfer of data between EDC and data analysis software packages, potentially using incompatible data file formats, and manual or heuristics-based data preprocessing.

In order to provide an answer to these drawbacks, I developed CDM, integrating EDC and data analysis APIs into a single software framework. To the best of my knowledge, CDM is the first framework providing such integration, enabling the machine-learning workflow to be much more automated, using CRF metadata instead of error-prone heuristics. Moreover, the availability of information about a CRF's hierarchical structure, enables the implementation of preprocessors that take this structure into account, for which no heuristics-based alternative exists.

Through its *Jython* interface, CDM enables interactive use of its data analysis APIs, greatly facilitating experimenting with the effect of decisions made in the machine-learning workflow, about choices of preprocessors and machine-learning algorithms. Its built-in functionality for generating learning curves empowers study coordinators to easily monitor their study's progress, enabling them to make an informed decision about when to terminate data collection. This helps prevent the creation of underpowered diagnostic models, as well as the collection of unnecessary data, which is incapable of further improving predictive performance.

Hence, CDM's integration of EDC and machine-learning facilitates streamlining the machine-learning workflow. Equally important though, is that its extensive test suite and sound architecture form a solid basis for further development.

In this further development, these data analysis APIs will be integrated into CDM's user interface, enabling study coordinators to monitor study performance as patient data accumulate. To this end, background processes will update machine-learning models, and store the results, as patient data come in. This will empower clinicians to handle more of the machine-learning workflow themselves, decreasing their reliance on consultancy from machine-learning experts.

Further future work will enable derived machine-learning models to be integrated in CRFs, for use in a prediction component. This will allow CDM to be used for the entire machine-learning worfklow, from data collection, machine-learning, to prediction. While today, published diagnostic models often fail to be used in

clinical practice, due to model complexity and the unavailability of prediction tools, the integration of such functionality into CDM would considerably simplify dissemination of derived diagnostic models. Whereas currently the medical community prefers the use of simple models, based on logistic regression, the availability of a prediction tool would enable the use of more sophisticated, better performing machine-learning algorithms.

Summarizing, CDM's current capabilities considerably simplify the workflow from data to model for the machine-learning expert, and the convenience method for easily generating learning curves provides study coordinators an invaluable tool for determining when data collection can be terminated. Future work should see the development of additional functionality for increasingly assisting clinicians in the management of the clinical diagnostic model research workflow. It should further allow the creation of a user interface using the derived models for diagnosing new patients, simplifying the dissemination of models.

Chapter 7

Clinical Data Miner results

In this chapter, I present the progress made on patient data collection with Clinical Data Miner (CDM), as well as a few examples of the use of CDM's data analysis APIs.

7.1 Introduction

A software framework is only useful if it is used. In this chapter, I show that since May 2011, when it was first put into production, CDM has been used extensively for collecting patient data in the context of clinical diagnostic model research, as well as for inter-rater agreement studies. I further demonstrate how CDM's APIs can be used to perform data analysis.

7.2 International Endometrial Tumour Analysis

As mentioned in Section 2.5, the study organized by the International Endometrial Tumour Analysis (IETA) consortium is further subdivided in a number of studies. One of these, namely International Endometrial Tumour Analysis (IETA) #2, evaluates inter-rater agreement of some of the variables used in the IETA studies. The other phases, #1, #3, and #4, involve the collection of data gathered during patient consultations, and aim to model the diagnosis of endometrial pathology based on the evaluation of ultrasound imaging modalities, described using the IETA terminology.

T. Van den Bosch	M.A. Pascual	L.P.G. Francesco
R. Fruscio	E. Epstein	D. Fischerova
J.L. Alcazar	D. Franchi	C. Lanzani
A. Rossi	L. Haakova	S. Guerriero
C. Van Holsbeke	L. Valentin	A. Votino
F. Mascilini	A.C. Testa	A. Dilegge
G. Opolskiene	A. Jakab	P. Sladkevicius
R. Zlotorowicz	B. Virgilio	M. Ludovisi
C. Van Pachterbeke	V. Chiappa	C. Penati
D. Dordoni	I. Tsikhanenka	M. Signorelli
P. Capmas	F. Rizzello	M. Szajnik
K. Van Tornout	M. Kudla	M. Baumgarten
D. Rysak-Luberowicz	R. Di Pace	J. Kaijser

Table 7.1 – List of active participants in the studies organized by the IETA consortium.

In this section, I present the progress that the IETA consortium have made with respect to data collection, since the start of the studies. The subsections list active participants and inclusion numbers, respectively.

7.2.1 Participants

Since the start of the studies, the IETA consortium has recruited the 39 participants listed in Table 7.1. They are geographically distributed across diverse centres in Europe, as indicated on the map of Figure 7.1. Most patient entries originated from Western, Northern, and Southern Europe, with a relatively small contribution from countries in Eastern Europe. The centres include regional hospitals as well as referral centres. They include both general hospitals and centres specialized in oncology. Different countries use different referral approaches, leading to different patient mixes. All these factors should ensure the studies form a relatively heterogeneous group, resulting in patient data that can be used to model the relationship between the IETA terminology and endometrial pathology, not just of a single country, or of a particular composition, but instead should be sufficiently general to derive a universal model.

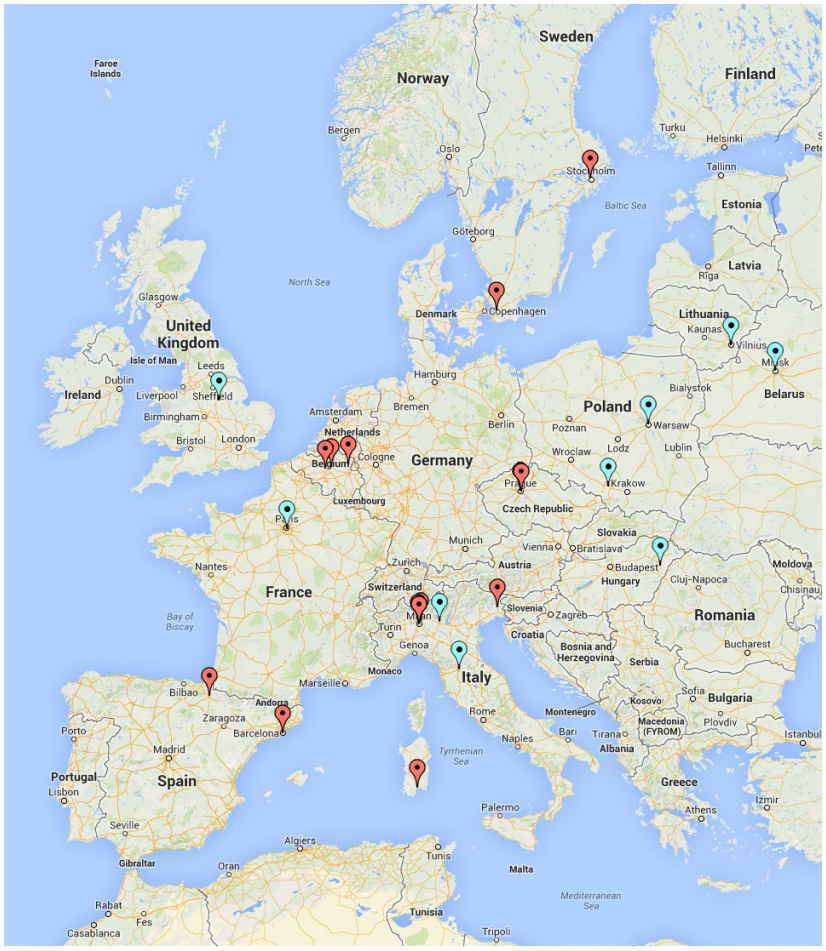


Figure 7.1 – Geographical distribution of centres participating in the IETA studies. Centres that contributed fewer than 50 total patient entries are marked in blue, while the others are marked in red.

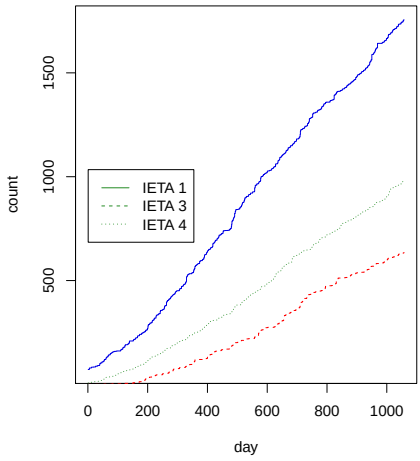


Figure 7.2 – Evolution of patient inclusions over time, for the different phases of the IETA study.

Study	Complete	Incomplete	Total
IETA #1a	590	139	729
IETA #1b	661	182	843
IETA #1c	154	66	220
IETA #3	510	128	638
IETA #4	591	421	1012
Total	2506	936	3442

Table 7.2 – Number of inclusions for the different IETA studies, as of April 10, 2014.

7.2.2 Inclusions

Patient inclusions for the different IETA studies have mostly increased linearly since the CDM server was taken into production, in May 2011. Figure 7.2 illustrates this for each of the different IETA studies. On April 10, 2014, patient inclusion levels had attained the levels from Table 7.2. In total, 3442 patient entries were collected, 2506 of which include a gold standard diagnosis.

7.3 Other studies

In CDM, the Case Report Forms (CRFs) used to compose a study are conveniently defined by means of spreadsheets. CDM's Electronic Data Capture (EDC) component can thus be straightforwardly applied to other studies than those of the IETA consortium.

Several such studies involving the collection of patient consultation data are planned. Apart from the IETA studies, the International Pregnancy of Unknown Location Analysis (IPULA) study, led by T. Bourne and G. Condous[12], is now online. Other studies about robotic surgery, in collaboration with I. Vergote, and about optimal cytoreduction, in collaboration with A.C. Testa et al., are currently in the design stage.

Additionally, several inter-rater agreement studies have been conducted leveraging the specialized user interface created for organizing this type of study. The first examined the influence of pictograms on data quality, and is described in more detail in Chapter 4. Another study examined the inter-rater agreement of some of the variables included in the IETA consensus paper[45] (IETA #2, led by L. Valentin). Yet another sought technological recommendations for ultrasound settings during examination of the endomyometrial junction (in collaboration with A. Votino et al.[85, 84, 86]). For an extensive list of the inter-rater agreement studies organized using the CDM software framework, refer to Appendix A.

7.4 Data analysis example scripts

In this section, I demonstrate how CDM's data preprocessing and machine-learning APIs can be leveraged from within a *Jython* console for data analysis. I do this using a few example scripts, along with the output they produce.

The first example shows how the distribution of the different categories of a certain variable can be computed. A second example shows how to determine contingency tables. The third example demonstrates the use of the machine-learning API for calculating learning curves. The section concludes with an example of how to generate a model and use it for calculating predictions.

7.4.1 Class distribution

The class distribution of a variable can be trivially calculated using the `dm` *Jython* module. Textbox 7.1 shows an example script for listing the distribution of the possible outcomes, for the data of all the IETA studies combined. The first lines specify which data should be loaded. The next few lines select the field of interest, and obtain its list of possible categories. Then, the occurrence of each of these categories is counted, discarding patient entries for which this variable was not filled in, and finally the results are printed as the ratio of the number of occurrences of each category divided by the amount of times the field was filled in. Results of this script are listed in Textbox 7.2.

```
import dm

# 1. Load combined data of all IETA studies.
study_ids = [ 'ieta_1a', 'ieta_1b', 'ieta_1c',
              'ieta_3', 'ieta_4' ]
descriptor = dm.merge(*[ dm.study(id) for id in study_ids ])
data = descriptor.load()

# 2. Determine field and its associated categories.
field = dm.findField(data.getFields(), 'ieta_outcome.endometrium')
categories = field.type().getValues()

# 3. Count occurrence of each category.
results = dict([ (category, 0) for category in categories ])

for p in data:
    value = p.getValue(field)
    if value != None:
        results[value] += 1

# 4. Print distribution
total = float(sum(results.values()))
for category in categories:
    print category, results[category] / total
```

Textbox 7.1 – This script loads the data from all IETA studies, merges them, and prints out a class distribution of the variable `ieta_outcome.endometrium`.

The distribution for the outcome and menopausal status variables can be obtained similarly, producing the results from Table 7.3 and Table 7.4, respectively. They present class distribution for each phase of the IETA study separately, as well as for all phases combined.

```
atrophy 0.112632508834
proliferative_endometrium 0.108215547703
secretory_endometrium 0.083480565371
endometrial_hyperplasia_without_atypia 0.0446113074205
atypical_hyperplasia 0.00706713780919
malignancy 0.278268551237
endometrial_polyp 0.27871024735
intracavitary_myoma 0.0521201413428
endometritis 0.00353356890459
other 0.0313604240283
```

Textbox 7.2 – Results generated by the script listed in Textbox 7.1.

Outcome	IETA 1	IETA 3	IETA 4	Overall
atrophy	11.4	26.1	0.2	11.3
proliferative endometrium	16.0	9.4	0.5	10.8
secretory endometrium	13.9	2.6	0.2	8.3
hyperplasia without atypia	6.6	2.9	0.9	4.5
atypical hyperplasia	0.7	0.0	1.2	0.7
malignancy	5.6	2.2	95.3	27.8
endometrial polyp	32.6	50.6	1.0	27.9
intracavitary myoma	8.2	3.4	0.0	5.2
endometritis	0.6	0.2	0.0	0.4
other	4.4	2.6	0.7	3.1

Table 7.3 – Outcome distribution for the different IETA studies.

7.4.2 Contingency tables

For contingency tables as well, the `dm` module provides all necessary functionality. An example script for determining such tables is demonstrated in Textbox 7.3. Again, first, the data are loaded. Second, the fields of interest, `ieta_hist.menopausal_status` and `ieta_outcome.endometrium` are queried, and their possible categories are obtained. Third, occurrences for each possible combination of the two fields are counted. And finally, the results are printed

Menopausal status	IETA 1	IETA 3	IETA 4	Overall
pre-menopausal	63.9	46.3	11.4	48.0
post-menopausal	36.1	53.7	88.6	52.0

Table 7.4 – Distribution of menopausal status for the different IETA studies.

out as percentages. The output of this script is presented in Table 7.5.

```
import copy
import dm

# 1. Load combined data of all IETA studies.
study_ids = [ 'ieta_1a', 'ieta_1b', 'ieta_1c',
              'ieta_3', 'ieta_4' ]
descs = [ dm.study(id) for id in study_ids ]
descriptor = dm.merge(*[ dm.study(id) for id in study_ids ])
data = descriptor.load()

# 2. Determine fields and their associated categories.
field1 = dm.findField(data.getFields(),
                      'ieta_hist.menopausal_status')
field2 = dm.findField(data.getFields(),
                      'ieta_outcome.endometrium')
categories1 = field1.type().getValues()
categories2 = field2.type().getValues() + [ 'N/A' ]

# 3. Count occurrence of each category.
inner = dict([ (c, 0) for c in categories1 ])
outer = dict([ (c, copy.copy(inner)) for c in categories2 ])

for p in data:
    v1 = p.getValue(field1)
    v2 = p.getValue(field2)

    if v2 != None:
        outer[v2][v1] += 1
    else:
        outer['N/A'][v1] += 1

# 4. Print distribution
total = float(data.size())
print 'outcome,', ', '.join([ d for d in categories1 ])

print
for c in categories2:
    print c,
    for d in categories1:
        print ', %.1f' % (outer[c][d] * 100 / total),
    print
```

Textbox 7.3 – This script loads the IETA data, counts the number of occurrences for each possible combination of the variables `menopausal_status` and `outcome`, and prints the frequency for each.

Outcome	Pre-menopausal (%)	Post-menopausal (%)
atrophy	1.3	9.1
proliferative endometrium	8.0	1.9
secretory endometrium	7.0	0.6
hyperplasia without atypia	2.8	1.3
atypical hyperplasia	0.3	0.3
malignancy	2.6	22.9
endometrial polyp	14.1	11.5
intracavitary myoma	4.1	0.6
endometritis	0.3	0.0
other	1.8	1.1
N/A	5.7	2.6

Table 7.5 – Contingency table tabulating the frequency distribution of menopausal status versus outcome. This table's contents are produced by the script from Textbox 7.3.

7.4.3 Learning curves

Generating learning curves requires both the `dm` and `ml` modules. Textbox 7.4 lists the steps involved in generating such learning curves. The first step specifies which data should be loaded. Next, the data are preprocessed in order to prepare them for the application of a classification algorithm. Then, the classification algorithm is selected, and the output field specified. The method `ml.sweep`() splits the data into training and test sets containing 70% and 30% of data points, respectively, then generates models for subsets of the training data sets, with sizes specified by the `ml.sweep`() method's corresponding parameter, and calculates predictive performance measures for each model. These include accuracy, sensitivity, specificity, and AUC. The resulting performance data are converted to learning curves, plotting performance with respect to sample size, by the method `ml.create_plots`(). These are then finally saved to files by means of the `ml.savePlot`() method. Examples of learning curves generated by CDM are visualized in Figure 7.3, showing AUC of the different IETA studies separately, and in Figure 7.4, which shows AUC for the combined IETA data set.

```
import dm

# 1. Specify data to load
study_ids = [ 'ieta_1a', 'ieta_1b', 'ieta_1c',
              'ieta_3', 'ieta_4' ]

desc_1 = dm.merge(*[ dm.study(id) for id in study_ids ])

# 2. Preprocess data
desc_2 = dm.stripOptional(dm.stripText(dm.stripDates(desc_1)))
desc_3 = dm.label(desc_2, 'ieta_outcome.*')
desc_4 = dm.createFactorProxies(dm.flatten(dm.normalize(desc_3)))
data = desc_4.load()

# 3. Calculation of learning curves
import weka.classifiers.functions.Logistic as Logistic
import ml

classifier = ml.newWekaClassifier(Logistic())
output_field = dm.findField(data.getOutputFields(),
    'ieta_outcome.endometrium.malignancy')
perf_map = ml.sweep(classifier, data, output_field,
    'yes', 50, range(50, data.size(), 50))

# 4. Generate and save plots
plots = ml.create_plots([ perf_map ], [ 'IETA' ])

for plot in plots:
    ml.savePlot(plot, plot.getTitle() + '.pdf')
```

Textbox 7.4 – Script for generating and saving learning curves, leveraging CDM's `dm` and `ml` modules.

7.4.4 Model predictions

Using CDM's machine-learning interfaces to generate models, predictions are readily obtained. The script from Textbox 7.5 first defines which data to load, and prepares them for logistic regression analysis. Next, the data are split into training and test data sets. Using WEKA's implementation of logistic regression, a model based on the training data is created. Finally, the model is applied to the test data to print out the expected probability of malignant disease. Since the outcome is known for the test data, these probabilities are compared to the actual outcome.

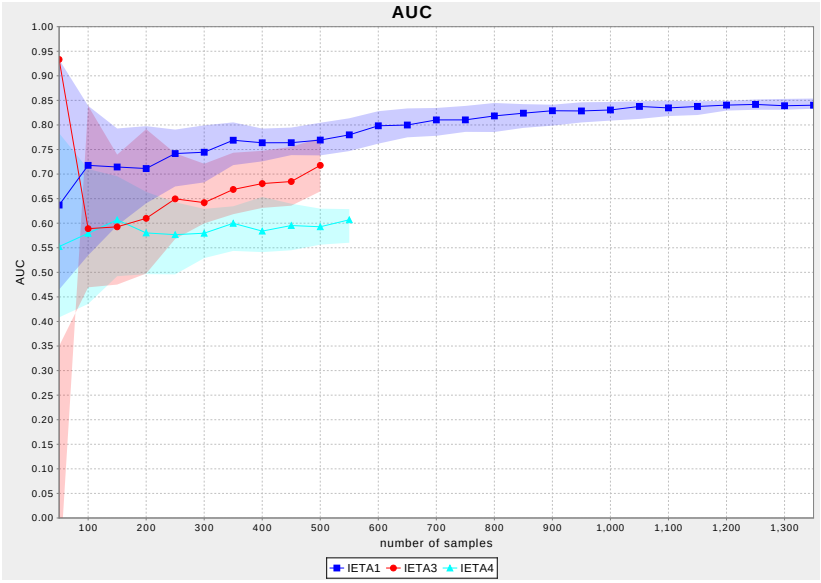


Figure 7.3 – Learning curves showing AUC with respect to sample size for models predicting endometrial malignancy derived from the data sets of the different IETA studies. Variability of the results is assessed by creating several training-test splits, and generating learning curves for each. The thick lines indicate median values, while the region around these lines represent the interquartile range (IQR) of the results.

7.5 Conclusion

Since CDM went into production in May 2011, it has been collecting data for the IETA studies, as well as for several inter-rater agreement studies. Moreover, several other studies are currently in the design stage.

In this chapter, I have further shown a few examples of how CDM's data analysis APIs can be applied to perform certain analyses. While such analyses currently require local access to the CDM server, in future work I will enable secure remote data access, enabling others to use CDM's data analysis capabilities, which, compared to traditional workflows, considerably simplify data analysis.

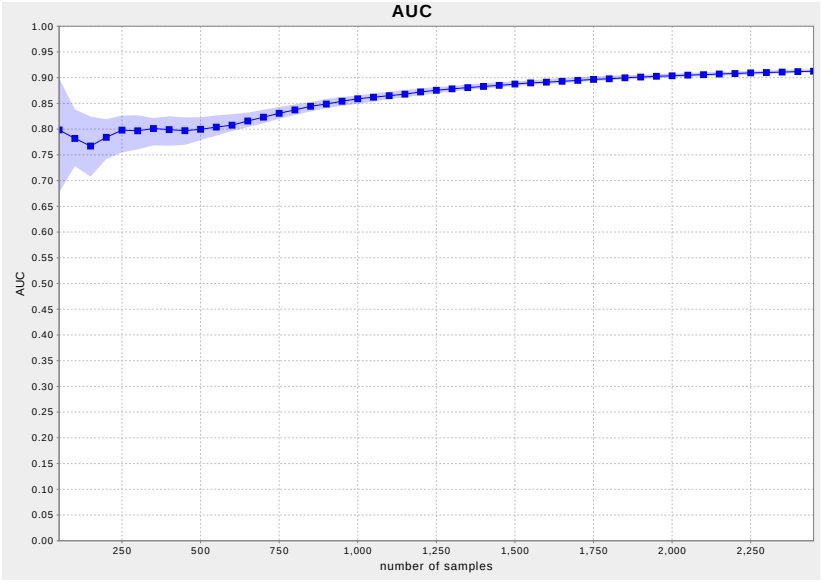


Figure 7.4 – AUC with respect to sample size, for models predicting endometrial malignancy, obtained from the merged data of all IETA studies combined.

```
import dm

# 1. Specify data to load
study_ids = [ 'ieta_1a', 'ieta_1b', 'ieta_1c',
              'ieta_3', 'ieta_4' ]

desc_1 = dm.merge(*[ dm.study(id) for id in study_ids ])

# 2. Preprocess data
desc_2 = dm.stripOptional(dm.stripText(dm.stripDates(desc_1)))
desc_3 = dm.label(desc_2, 'ieta_outcome.*')
desc_4 = dm.createFactorProxies(dm.flatten(dm.normalize(desc_3)))
data = dm.shuffle(desc_4.load())

# 3. Create training and test set
output_field = dm.findField(data.getOutputFields(),
    'ieta_outcome.endometrium.malignancy')
data_sets = dm.stratify(data, output_field, 70, 30)
training_data = data_sets[0]
test_data = data_sets[1]

# 4. Create model for training data
import weka.classifiers.functions.Logistic as Logistic
import ml

classifier = ml.newWekaClassifier(Logistic())
model = classifier.model(training_data, output_field)

# 5. Print predictions for test data
for p in test_data:
    prob_cancer = model.predict(p).getLikelihood('yes')
    print p.getPatientId(), ':', '%.1f%%' % (100 * prob_cancer),
    print '- actual outcome:', p.getOutputValue(output_field)
```

Textbox 7.5 – Script for calculating a model on training data, subsequently used for computing predictions on test data.

```
XXXXXXXX : 20.2% - actual outcome: yes
XXXXXXXX : 10.8% - actual outcome: no
XXXXXXXX : 81.4% - actual outcome: yes
XXXXXXXX : 0.3% - actual outcome: no
XXXXXXXX : 20.2% - actual outcome: no
XXXXXXXX : 1.1% - actual outcome: no
XXXXXXXX : 1.6% - actual outcome: no
XXXXXXXX : 5.4% - actual outcome: no
XXXXXXXX : 73.5% - actual outcome: yes
XXXXXXXX : 95.3% - actual outcome: yes
XXXXXXXX : 80.7% - actual outcome: yes
XXXXXXXX : 0.4% - actual outcome: no
XXXXXXXX : 46.6% - actual outcome: yes
```

Textbox 7.6 – Excerpt of output generated by the script from Textbox 7.5. In order to respect patient privacy, I substituted patient identifiers with fixed text.

Chapter 8

Conclusions and future research

8.1 Achievements

As the pace of medical research is ever accelerating, it drives not only an increasing need for data collection, but also for more efficient data analysis. The Clinical Data Miner (CDM) project discussed in this thesis therefore aims to support medical research by simplifying the workflow from data to model. To that end, it integrates data collection and data analysis in a single software framework.

I have implemented user interfaces both for the collection of data in the context of patient consultations, as well as for organizing inter-rater agreement studies. In order to accomodate the collection of variables obtained through assessment of imaging-based modalities, these user interfaces enable the integration of visual cues, by means of pictograms, for clarifying questions. A survey about their use has shown high user satisfaction levels, and users are especially pleased about the user interface integration of pictograms.

For Clinical Data Miner (CDM)'s data analysis capabilities, the focus has so far mostly been on diagnostic model research. The availability of good diagnostic models can have a considerable impact on disease management. For many diseases, treatment and/or management options are available, as long as the disease is diagnosed sufficiently early. Endometrial cancer, for example, causes relatively fewer deaths than many other types of cancer, thanks to

many being diagnosed at an early stage, clearly demonstrating the impact diagnostic procedures can have on patient survival. On the other hand, many diagnostic techniques require invasive procedures, potentially causing patients to avert diagnosis. These two factors combined explain the rising interest in diagnostic model research, which, using machine-learning techniques, aims to create diagnostic models that rely as much as possible solely on features that can be obtained using non-invasive means. Apart from leading to less risk and discomfort for the patient, such non-invasive diagnostic models could lead to higher compliance with diagnostic follow-up, improving patient survival.

Traditional machine-learning algorithms applied in clinical diagnostic model research, such as logistic regression, require complex, time-consuming preprocessing, in order to obtain sufficient predictive performance, impeding automation of the machine-learning workflow. For the data set of the International Ovarian Tumour Analysis (IOTA) consortium, I have shown that this performance can be exceeded or matched by more sophisticated machine-learning algorithms, such as Least-Squares Support Vector Machines (LS-SVM), applied directly to raw data. This provides the possibility of automating clinical diagnostic model research.

As an initial step towards the goal of full automation, CDM considerably simplifies the clinical diagnostic model workflow by integrating data collection, data preprocessing, and machine-learning capabilities in a single software framework. Currently, CDM provides a convenient experimentation platform for interactively exploring predictive performance of various machine-learning workflows.

I have developed all components of the CDM software framework, including Electronic Data Capture (EDC), preprocessing and machine-learning components, using a Test-Driven Development approach, simplifying development, deployment and maintenance. This should further facilitate potential future expansion of the team working on CDM's development.

In future work, to further support the growing interest in diagnostic model research, CDM's data analysis capabilities will be integrated in its user interface, enabling straightforward follow-up by study coordinators.

8.2 Future work

While I have made considerable progress on CDM, it is by no means finished. More work is needed, on several levels. This involves additional methodological research, software development, and diagnostic model research.

Methodologically, I have shown that, for the International Ovarian Tumour Analysis (IOTA) data set, LS-SVM applied directly to the raw data provides at least the same performance as logistic regression after elaborate preprocessing of the data. I plan similar analyses on other data sets, such as that of the International Endometrial Tumour Analysis (IETA) consortium, in order to further support the generality of this conclusion.

Methods such as logistic regression provide some important advantages over LS-SVM, especially in a clinical context. One advantage is that logistic regression provides probability estimates rather than binary outcomes. This allows to better assess the risk associated with a certain outcome. Another advantage is that logistic regression models are better interpretable than LS-SVM models, especially if the latter use non-linear kernel functions. Additional investigation is required to adapt LS-SVM to provide these same benefits.

More importantly, since high-dimensional diagnostic models are of little practical use, I will research to what extent feature selection can be automated. One aspect that cannot be automated, and will require the input of medical experts, is the estimation of the relative "costs" of variables. These include both the objective, financial costs of obtaining certain features, as well as subjective costs, relating to the level of discomfort to the patient of obtaining the variable, the health risk involved, or the need for surgery. Given a labelled data set, and given these relative feature costs, can feature selection be fully automated? This will be an important factor determining the feasibility of the aims of the CDM project, which I intend to investigate.

Further work is also needed to design efficient methods for handling the skip pattern commonly present in Case Report Forms (CRFs). While CDM already implements such a method, other, more sophisticated methods would likely lead to improved predictive performance.

Considering software development challenges, I plan to leverage CDM's data preprocessing and machine-learning Application Programming Interfaces (APIs) to provide study coordinators with a user interface that allows them to manage the machine-learning workflow without assistance from Information Technology (IT) or machine-learning experts. This requires the implementation of several new features in CDM. First, it requires automation of the entire machine-learning workflow, including feature selection. Second, study coordinators will need a user interface for monitoring the evolution of a study's predictive performance. When they deem predictive performance to be sufficient, they should be able to "freeze" a model and store it. And finally, I will extend CDM with a user interface for predicting a patient's diagnosis, based on the value of a number of input variables provided by the clinician.

Another potentially useful software development avenue would involve extending CDM's user interface to serve as a teaching tool, for assessing a clinician's ability to correctly measure relevant clinical variables. A correct assessment of study variables is a prerequisite for the creation of valid diagnostic models. Correct assessment by clinicians is equally required for reliable prediction of patient outcomes. Such a teaching tool could require a minimum score, in order to guarantee study participants, or clinicians using a diagnostic model, have a sufficient level of experience.

Apart from these methodological and software challenges, the objectives stated in the project entitled "Endometrial cancer diagnosis based on predictive computer models within an International Endometrial Tumour Analysis (IETA) collaboration" will have to be pursued. This project, in which I participate, is led by D. Timmerman and B. De Moor, and is funded by Agenschap voor Innovatie door Wetenschap & Technologie (IWT) - Toegepast Biomedisch Onderzoek met een primair maatschappelijke finaliteit (TBM) (IWT-TBM). Its primary goal involves the design of a clinical diagnostic model for endometrial cancer. As its secondary goal, this model will be integrated in CDM's user interface, enabling the calculation of predictions for patients presenting with symptoms indicative of endometrial cancer. Finally, the project's stretch goal entails the creation of a model that can be used as a screening test, which could be deployed to a wide population of asymptomatic women.

Clearly, many of the IWT-TBM's project's objectives overlap to a large extent the methodoligical and software engineering challenges previously discussed. Results of analyses carried out while researching the optimal diagnostic model for endometrial cancer will guide the implementation of an automated machine-learning workflow. Work on a general user interface for calculating predictions based on a stored model can be used directly for the IWT-TBM project's secondary goal.

8.3 Dissemination

Several avenues are possible with regard to dissemination of this work. Some of these could be implemented now, while others will be enabled by future work.

The organization of more studies with CDM is one possible approach to further dissemination. These can be diagnostic model studies or inter-rater agreement studies. As already stated, several studies are currently in the design stage, and this dissemination path will certainly be explored further.

Also, as a diagnostic model is derived from the IETA data collected by CDM, it

will be published by the IETA consortium, for wider adoption by the gynaecologic community. A user interface using this model to predict risk of endometrial cancer for new patients should support this adoption.

Concerning dissemination of the CDM software framework itself, several options are available. The software framework as a whole, or its data collection component separately, could be licensed to a third party, who could then commercialize it. Since in this scenario, the intellectual property (IP) rights remain with KU Leuven, this would enable further research, especially on CDM's data analysis components.

Another possibility would be to start a new company for commercializing CDM. Several business models are possible. Charging a fixed price for the entire software or its components is one possibility. Alternatively, one could collect micro-payments for each use of CDM's individual components, namely its EDC, machine-learning, and prediction components. Care should be taken, however, not to discourage the collection of large data sets, as doing so might be detrimental to predictive performance of calculated models.

Using a different business model, one could provide open-source access to the software, while offering support contracts as well as certified releases to paying customers. While this would lead to a substantial amount of non-paying customers, it would lead to higher visibility, which could eventually lead to more paying customers.

Finally, one could opt for societal instead of financial valorization, by further developing CDM within KU Leuven. This could also serve as a driver for international research collaboration on various medical studies.

8.4 Conclusion

Since diagnostic models can improve patient comfort and survival, many other such studies can be expected.

Observing examples from the IOTA and IETA studies, such research starts with identification of (potentially) relevant variables, and definition of terms used to describe them. Inter-rater agreement studies, facilitated by CDM, help assess the reliability of these variables. CDM's EDC software component enables the collection of data. Its machine-learning capabilities simplify the determination of diagnostic models. Future work will promote dissemination of models by providing a user interface enabling the calculation of predictions. A future teaching tool could examine users' proficiency with certain terminology prior to accepting their patient submissions.

In short, CDM presently enables automating many steps in the workflow of clinical diagnostic model research, and will expand this automation further in the future, to simplify this research. As EDC has done for collection of patient data before, this will deliver efficiency gains, which ultimately could accelerate clinical diagnostic model research.

Appendix A

Inter-rater agreement studies

This appendix details some aspects of the inter-rater agreement studies organized using Clinical Data Miner (CDM)'s Electronic Data Capture (EDC) user interface, introduced in Subsection 4.2.2, which was modified for facilitating the organization of such studies.

A.1 Influence of pictograms

This inter-rater agreement study is described in detail in Chapter 4. Results were presented at the 21st World Congress on Ultrasound in Obstetrics and Gynaecology in 2011[38].

A.2 Polycystic Ovaries (PCOs)

These studies, organized in collaboration with D. Van Schoubroeck, investigated inter-rater agreement for the diagnosis of PCO, as well as for the evaluation of certain of their aspects. Five study participants evaluated 40 ultrasound images for this study. Results were presented at the 22nd World Congress on Ultrasound in Obstetrics and Gynaecology in 2012[79].

A.3 Uterine anomalies

Also organized in collaboration with D. Van Schoubroeck, this study examined inter-rater agreement for the diagnosis of uterine anomalies, and their expected influence on fertility. It involved the evaluation of 60 ultrasound images by five study participants. Results were presented at the 22nd World Congress on Ultrasound in Obstetrics and Gynaecology in 2012[78].

A.4 Endomyometrial junction

Organized in collaboration with A. Votino, these studies compared inter-rater agreement of images obtained by different ultrasound technologies. Five clinicians participated in this study, one of whom evaluated the first study twice, enabling evaluation not only of inter-rater agreement, but also of intra-rater agreement.

In the first study, technologies included were Volume Contrast Imaging (VCI) 2mm, VCI 4mm, and 2D. Inter-rater agreement calculation was based on 80 ultrasound images for each technology. The second study compared render and 2D technologies, with and without hystero-salpingography. Each technology was again evaluated using 80 images.

Results of these studies were presented at the 22nd World Congress on Ultrasound in Obstetrics and Gynaecology in 2012[85, 84].

A.5 International Endometrial Tumour Analysis #2

This study by the International Endometrial Tumour Analysis (IETA) consortium evaluates some of the terminology introduced by the IETA consensus paper[45]. It involved the evaluation of 112 grayscale and 112 Doppler ultrasound images. Eight clinicians participated in this study, stratified to experience level into two groups, to enable evaluation of the images in two phases, with a two month time lapse. One group evaluated grayscale images during the first phase, and Doppler during the second; the other inverted this order, first evaluating Doppler images, with grayscale images evaluated in the second phase.

This study was performed in collaboration with L. Valentin. Analysis of the results is ongoing.

A.6 Image enhancement

For this study, I collaborated with T. Bourne. Its aim was to examine if adding a contrast enhanced version to a regular ultrasound image improves inter-rater agreement. This study involved the evaluation of 100 images without, and 100 images with contrast enhanced version. The ten participants were split in two groups, stratified to experience level, evaluating the two image sets in opposite order, with a time interval of two months between image sets. Analysis of the results is ongoing.

Appendix B

Case Report Forms

This appendix shows the structure of some Case Report Forms (CRFs) that have been used in studies organized using Clinical Data Miner (CDM)'s Electronic Data Capture (EDC) component.

B.1 Effect of pictograms on data quality

The CRFs described in this section were used for the analysis of the effect of pictograms on data quality, described in Chapter 4.

In the study, both these CRFs were combined in a single CRF, with the top question allowing participants to choose between either one. In very few cases, this led to erroneous identification of image type, and thus CRF selection, which were therefore excluded from analysis.

B.1.1 Unenhanced ultrasound

The CRF used for the evaluation of unenhanced ultrasound images includes many of the same variables as the "Unenhanced ultrasound" section included in the CRFs of International Endometrial Tumour Analysis (IETA) #1, #3, and #4.

- Endometrial echogenicity and pattern
  - uniform
    - 3-layer pattern
    - hyper-echoic

        - hypo-echoic
        - iso-echoic
    - non-uniform
        - regular cystic areas
        - irregular cystic areas
        - heterogeneous without cysts
        - heterogeneous with regular cysts
        - heterogeneous with irregular cysts
- Endometrial midline
    - linear
    - non-linear
    - irregular
    - not defined
- Presence of bright edge
- Endo-myometrial junction
    - regular
    - irregular
    - interrupted
    - not defined
- Colour score, as a value between 0 and 100 (when applicable)
- Colour score, as an ordinal variable (when applicable)
    1. no flow
    2. minimal flow
    3. moderate flow
    4. abundant flow
        - Vascular pattern, *in case of minimal, moderate, or abundant flow:*
            (a) single "dominant" vessel without branching
            (b) single "dominant" vessel with branching
            (c) multiple "dominant" vessels – focal origin
            (d) multiple "dominant" vessels – multifocal origin
            (e) scattered vessels
            (f) circular flow

B.1.2 Sonohysterography

The CRF used for the evaluation of sonohysterographies concurs in large part with the "Sonohysterography" section of the CRFs used for IETA #1 and #3.

- Outline of background endometrium
    - smooth
    - endometrial folds
    - polypoids
    - irregular
- Echogenicity of background endometrium
    - uniform
        - hyper-echoic
        - hypo-echoic
        - iso-echoic
    - non-uniform
        - regular cystic areas
        - irregular cystic areas

- heterogeneous without cysts
- heterogeneous with regular cysts
- heterogeneous with irregular cysts

- Colour score, as a value between 0 and 100 (when applicable)
- Colour score, as an ordinal variable (when applicable)
  1. no flow
  2. minimal flow
  3. moderate flow
  4. abundant flow
- Presence of intracavity lesion
  - no
  - yes
    - Lesion type
      - endometrial lesion
        - extent
          - localized ($< 25\%$)
          - extended ($\geq 25\%$)
          - not assessable
        - type of localized lesion
          - pedunculated
          - sessile
          - not applicable
          - not assessable
        - echogenicity
          - uniform
            - hyper-echoic
            - hypo-echoic
            - iso-echoic
          - non-uniform
            - without cystic areas
            - with regular cystic areas
            - with irregular cystic areas
        - outline
          - regular
          - irregular
      - lesion arising from the myometrium
        - echogenicity
          - uniform
          - non-uniform
        - grading
          - G0 (within the cavity)
          - G1 (endocavitary part $\geq 50\%$
          - G2 (endocavitary part $< 50\%$
    - colour score, as a value between 0 and 100
    - colour score, as an ordinal variable
      1. no flow
      2. minimal flow
      3. moderate flow
      4. abundant flow
         - Vascular pattern, *in case of minimal, moderate, or abundant flow:*
           (a) single “dominant” vessel without branching
           (b) single “dominant” vessel with branching
           (c) multiple “dominant” vessels – focal origin

(d) multiple “dominant” vessels – multifocal origin
(e) scattered vessels
(f) circular flow

Appendix C

Feature selection experiments

This appendix shows results of a number of experiments obtained from applying both feature selection and classification algorithms to data sets obtained from the International Ovarian Tumour Analysis (IOTA) data described in Section 5.2.

C.1 Algorithms

This analysis used two classification algorithms, namely logistic regression and Least-Squares Support Vector Machines (LS-SVM). The first, logistic regression[36, 2], is described in detail in Subsection 5.3.1, while LS-SVM[67, 68], as well as how it evolved from Support Vector Machines (SVM)[80, 14], are explained in Subsection 5.3.2. LS-SVM was combined both with linear and Radial Basis Function (RBF) kernels.

Feature selection algorithms used were Stepwise Logistic Regression (SLR)[36] and Automatic Relevance Determination (ARD)[75, 68], the latter using LS-SVM within a Bayesian evidence framework.

C.2 Learning curves

This first analysis compares learning curves obtained from different data sets, processed by either logistic regression or LS-SVM. All data sets are derived from the IOTA data described in Section 5.2. Using the workflow depicted

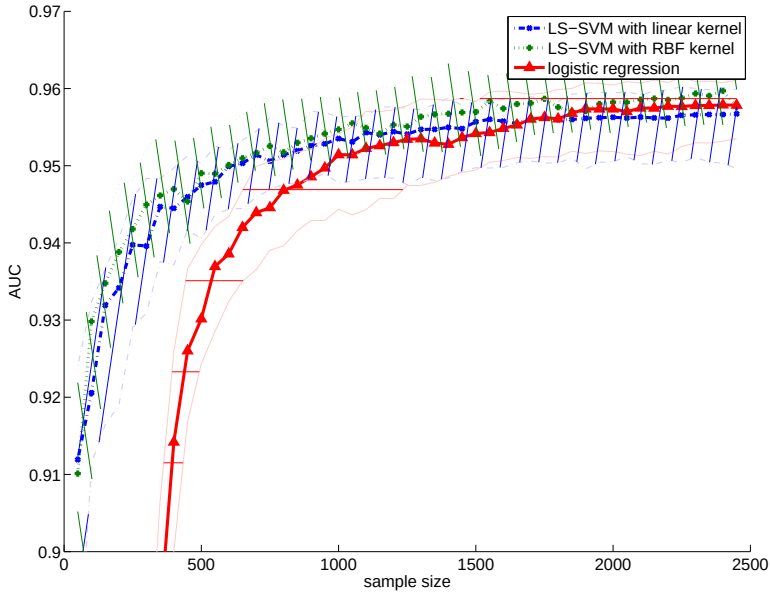


Figure C.1 – Learning curves showing evolution of AUC with respect to sample size, for logistic regression and LS-SVM, applied to the variables from the IOTA data set listed in Table 5.2.

in Figure 5.3, and described in Section 5.5, allowed the generation of learning curves and their interquartile range (IQR).

C.2.1 Performance on the full data set

The learning curves from Figure C.1 plot performance results obtained on the IOTA data in the absence of feature selection. They show Area under the Receiver Operating Characteristic (ROC) curve (AUC) values with respect to sample size, of logistic regression and LS-SVM, with linear and RBF kernels, applied to the variables listed in Table 5.2. As explained in Section 5.2, apart from variables collected directly by IOTA study participants, these include a few variables derived either intuitively, or by preprocessing. As in Chapter 5, the learning curves from Figure C.1 show LS-SVM to outperform logistic regression considerably at low sample sizes, while for large samples performance differences become insignificant.

Age	Ascites	papflow
soldmax	wallreg	Shadows

Table C.1 – Input features of the *[LR2]* data set. Derived features are enclosed in a black box.

soldmax	colscore	Ascites
wallreg	Shadows	ovaryd2

Table C.2 – Input features of the *[SLR6]* data set. Derived features are enclosed in a black box.

C.2.2 Performance on reduced data sets

Figure C.2 graphs the learning curves obtained from applying logistic regression and LS-SVM to three data sets derived from the IOTA data by feature selection.

The first of these data sets was obtained by a complex process involving the first 754 data points collected by the IOTA consortium. This process included the introduction of derived variables, followed by the application of SLR[36] for feature selection, This process is described in detail by Ameye[3], and resulted in a model based on twelve variables[70]. In order to obtain a more concise diagnostic model, which can be more easily deployed in clinical practice, this model was further restricted to the six most important variables, listed in Table C.1. This model was published by Timmerman et al.[70], and will be referred to as *[LR2]*.

The second data set consists of the variables obtained by applying SLR to a subsample of the IOTA data set, and choosing the six most important of the variables obtained. In order to create circumstances comparable to those used for determining the variables from the *[LR2]* data set, the subsample from which these variables were determined, contained 754 data points randomly drawn from the IOTA data without replacement. Table C.2 lists the variables retained by this procedure, referred to as the variables from the *[SLR6]* data set.

Finally, the third data set includes the variables obtained by applying ARD, using LS-SVM within the Bayesian evidence framework, to a subsample of the IOTA data set, constructed similarly as that used in determining the *[SLR6]* data set. Table C.3 lists the selected variables, forming the *[ARD6]* data set.

Note that the *[SLR6]* data set includes the `ovaryd2` variable, while the *[ARD6]* data set includes `solidd3`, which are the second diameter of the ovary and

lesdmax	solidd3	Shadows
venous	TAMXV	Ascites

Table C.3 – Input features of the *[ARD6]* data set. Derived features are enclosed in a black box.

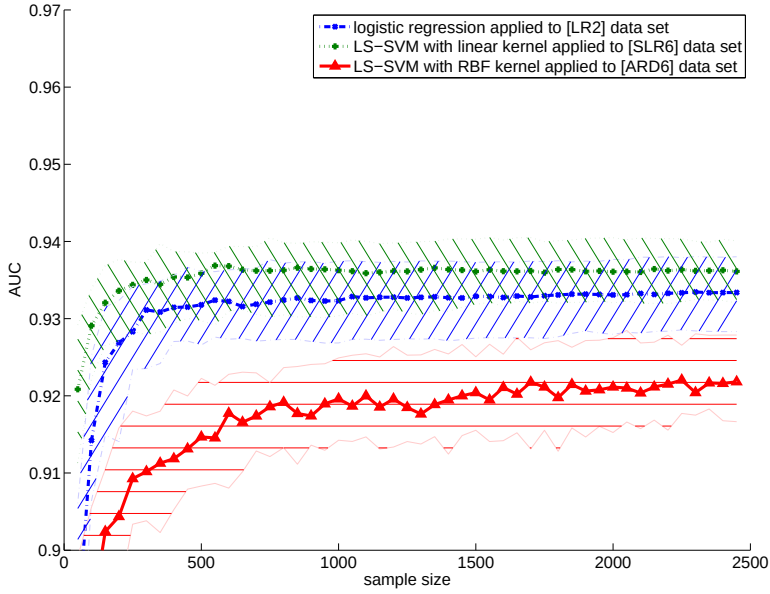


Figure C.2 – AUC learning curves for logistic regression and LS-SVM, applied to different subsets of variables from the IOTA data, obtained by feature selection.

the third diameter of the solid mass, respectively. While these variables have no physical meaning, in practice, clinicians typically enter length dimensions starting with the largest, and ending with the smallest. These variables can thus be interpreted as the intermediate ovarian diameter and the minimum diameter of the solid component, respectively. This demonstrates the importance of careful interpretation of feature sets obtained by automatic feature selection.

The curves from Figure C.2 present performance evolution of logistic regression applied to the *[LR2]* data set, of LS-SVM with a linear kernel applied to the *[SLR6]* data set, and of LS-SVM in conjunction with an RBF kernel applied to the *[ARD6]* data set. They show very similar performance for the two former workflows, while ARD offers lower performance.

C.3 Feature selection performance robustness

The experiment in this section evaluates robustness of predictive performance obtained from applying SLR and ARD to the IOTA data, followed by an appropriate classifier. Due to its linear behaviour, SLR is combined with either logistic regression or LS-SVM with a linear kernel, while ARD's non-linear properties are matched to LS-SVM with an RBF kernel.

For each combination, fifty samples, containing 754 data points each, were randomly drawn from the IOTA data, without replacement. Feature selection and classification were subsequently applied to each of the fifty samples.

Figure C.3 shows boxplots for the AUC values obtained. SLR combined with either logistic regression or LS-SVM with a linear kernel exhibit very similar results, with little variance. ARD followed by LS-SVM with an RBF kernel has lower median performance, and larger variance. Thus, for the data from the IOTA studies, feature sets obtained by SLR resulted in both higher and more robust performance than ARD. This is in contrast to LS-SVM with an RBF kernel performing equally or better than logistic regression on the full IOTA data set, as shown in Figure C.1. A possible reason for this is the specific approaches adopted by the used feature selection algorithms: while the SLR implementation used a combination of forward selection and backward elimination, LS-SVMLab's[16] ARD used backward elimination.

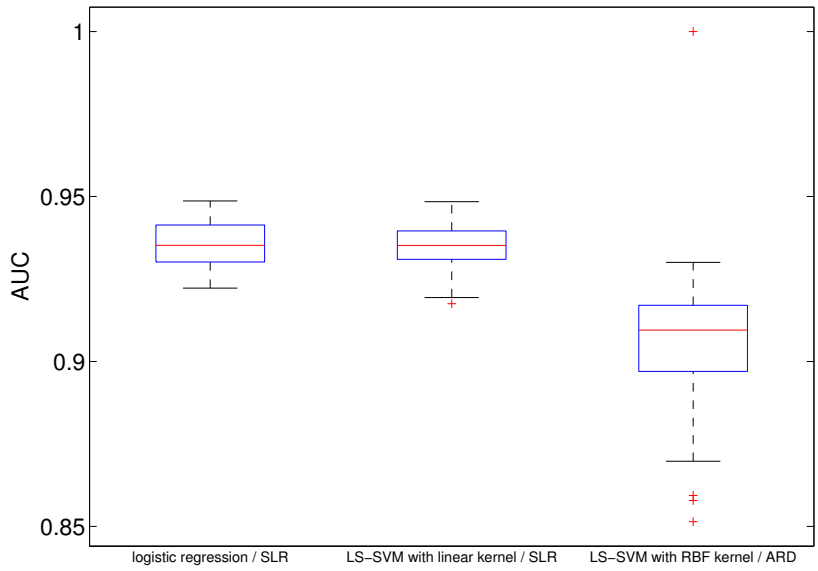


Figure C.3 – Performance comparison of SLR followed by either logistic regression or LS-SVM with a linear kernel, and ARD followed by LS-SVM with an RBF kernel.

Bibliography

[1] Cancer Incidence in Belgium, 2008. URL `http://www.kankerregister.be/media/docs/StK_publicatie.pdf`.

[2] A. Agresti. *Categorical Data Analysis*. John Wiley & Sons, Inc., second edition, 2002. ISBN 0471360937.

[3] L. Ameye. *Predictive models for classification problems in gynecology*. PhD thesis, Katholieke Universiteit Leuven, 2005.

[4] M. J. Barrett, E. G. Brown, and A. E. Twist. Web Clinical Trials Break Through. Technical report, Forrester Research, Cambridge, 2001.

[5] K. Beck. Aim, fire. *IEEE Software*, 18(5):87–89, 2001.

[6] K. Beck. *Test Driven Development: By Example*. Addison-Wesley Professional, Boston, MA, USA, 2002.

[7] K. Beck and C. Andres. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, Boston, MA, USA, 2004. ISBN 0321278658.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Stanford, second edition, 2004. ISBN 9780521833783.

[9] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3): 1–27, Apr. 2011. doi: 10.1145/1961189.1961199.

[10] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, Apr. 1960. doi: 10.1177/001316446002000104.

[11] K. B. Cohen and L. Hunter. Getting started in text mining. *PLoS computational biology*, 4(1):e20, Jan. 2008. doi: 10.1371/journal.pcbi.0040020.

[12] G. Condous, D. Timmerman, S. Goldstein, L. Valentin, D. Jurkovic, and T. Bourne. Pregnancies of unknown location: consensus statement. *Ultrasound in Obstetrics and Gynecology*, 28(2):121–122, 2006. doi: 10.1002/uog.2838.

[13] B. D. Connor. 2007 will be tipping point for edc. `http://www.bio-itworld.com/newsitems/2007/may/05-22-07-edc-forecast`, 2007. Accessed: January 12, 2014; archived by WebCite® at `http://www.webcitation.org/6Ma2bK6fz`.

[14] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, Sept. 1995. doi: 10.1007/BF00994018.

[15] F. Daoust, P. Hoschka, C. Z. Patrikakis, R. S. Cruz, M. S. Nunes, and D. S. Osborne. Towards Video on the Web with HTML5. In *Position paper, NEM Summit*, 2011.

[16] K. De Brabanter, P. Karsmakers, F. Ojeda, C. Alzate, J. De Brabanter, K. Pelckmans, B. De Moor, J. Vandewalle, and J. A. K. Suykens. LS-SVMlab Toolbox User's Guide version 1.8. 2010.

[17] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald. What can natural language processing do for clinical decision support? *Journal of biomedical informatics*, 42(5):760–72, Oct. 2009. doi: 10.1016/j.jbi.2009.08.007.

[18] C. Drummond and R. C. Holte. Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65(1):95–130, May 2006. doi: 10.1007/s10994-006-8199-5.

[19] B. Efron and G. Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48, 1983. doi: 10.2307/2685844.

[20] B. Efron and C. Stein. The Jackknife estimate of variance. *The Annals of Statistics*, 9(3):586–596, May 1981.

[21] K. El Emam, E. Jonker, M. Sampson, K. Krleža-Jerić, and A. Neisa. The use of electronic data capture tools in clinical trials: web-survey of 259 Canadian trials. *Journal of Medical Internet Research*, 11(1):e8, Jan. 2009.

[22] J. Ferlay, P. Autier, M. Boniol, M. Heanue, M. Colombet, and P. Boyle. Estimates of the cancer incidence and mortality in Europe in 2006. *Annals of Oncology*, 18(3):581–592, Mar. 2007. doi: 10.1093/annonc/mdl498.

[23] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[24] M. Fowler. Inversion of control containers and the dependency injection pattern. http://martinfowler.com/articles/injection.html, 2004. Accessed: January 12, 2014; archived by WebCite® at http://www.webcitation.org/6Ma33FsrU.

[25] M. Fowler. ReproducibleBuild. http://martinfowler.com/bliki/ReproducibleBuild.html, 2010. Accessed: February 3, 2014; archived by WebCite® at http://www.webcitation.org/6N6nFBpa5.

[26] C. Friedman and G. Hripcsak. Natural Language Processing and its future in medicine. *Academic Medicine*, 74(8):890–895, 1999.

[27] F. Funke and U.-D. Reips. Visual analogue scales in online surveys: Nonlinear data categorization by transformation with reduced extremes. In *General Online Research (GOR) conference*, Bielefeld, Germany, 2006.

[28] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[29] O. Gevaert, A. Daemen, B. De Moor, and L. Libbrecht. A taxonomy of epithelial human cancer and their metastases. *BMC medical genomics*, 2: 69, Jan. 2009. doi: 10.1186/1755-8794-2-69.

[30] Google Web Toolkit. Google web toolkit. http://www.gwtproject.org, 2006. Accessed: June 2, 2014; archived by WebCite® at http://www.webcitation.org/6Q22ICQGB.

[31] J. Gosling, B. Joy, G. L. Steele, G. Bracha, and A. Buckley. *The Java language specification - Java SE 7 edition*. Addison-Wesley Professional, 2013. ISBN 978-0133260229.

[32] M. Graefen. International Validation of a Preoperative Nomogram for Prostate Cancer Recurrence After Radical Prostatectomy. *Journal of Clinical Oncology*, 20(15):3206–3212, Aug. 2002. doi: 10.1200/JCO.2002.12.019.

[33] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18, 2009. doi: 10.1145/1656274.1656278.

[34] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde. Research electronic data capture (REDCap)–a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of biomedical informatics*, 42(2):377–81, 2009. doi: 10.1016/j.jbi.2008.08.010.

[35] J. Hilden. The area under the ROC curve and its competitors. *Medical Decision Making*, 11(2):95–101, 1991.

[36] D. W. Hosmer and S. Lemeshow. *Applied logistic regression*. Wiley-Interscience Publication, second edition, 2000. ISBN 0471356328.

[37] J. Huang and C. X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17 (3):299–310, Mar. 2005. doi: 10.1109/TKDE.2005.50.

[38] A. Installé, T. Van den Bosch, D. Van Schoubroeck, J. Heymans, L. Zannoni, L. Jokubkiene, P. Sladkevicius, L. Valentin, B. De Moor, and D. Timmerman. Clinical Data Miner (CDM) - A web-based electronic data capture framework for multi-centric studies with imaging modalities. In *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology*, volume 38, page 33, Los Angeles, 2011.

[39] I. Jacobs, D. Oram, J. Fairbanks, J. Turner, C. Frost, and J. G. Grudzinskas. A risk of malignancy index incorporating CA 125, ultrasound and menopausal status for the accurate preoperative diagnosis of ovarian cancer. *British Journal of Obstetrics and Gynaecology*, 97:922–929, 1990. doi: 10.1016/0378-5122(91)90134-C.

[40] A. Jemal, R. Siegel, E. Ward, Y. Hao, J. Xu, T. Murray, and M. J. Thun. Cancer statistics, 2008. *CA: a cancer journal for clinicians*, 58(2):71–96, 2008. doi: 10.3322/CA.2007.0010.

[41] A. Jemal, F. Bray, M. M. Center, J. Ferlay, E. Ward, and D. Forman. Global Cancer Statistics. *CA: a Cancer Journal for Clinicians*, 61(2):69–90, 2011. doi: 10.3322/caac.20107.

[42] J. Juneau, J. Baker, V. Ng, L. Soto, and F. Wierzbicki. *The Definitive Guide to Jython*. Paul Manning, 2010. ISBN 9781430225270.

[43] M. W. Kattan, J. a. Eastham, a. M. Stapleton, T. M. Wheeler, and P. T. Scardino. A preoperative nomogram for disease recurrence following radical prostatectomy for prostate cancer. *Journal of the National Cancer Institute*, 90(10):766–771, May 1998.

[44] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(30), 2006.

[45] F. P. G. Leone, D. Timmerman, T. Bourne, L. Valentin, E. Epstein, S. R. Goldstein, H. Marret, A. K. Parsons, B. Gull, O. Istre, W. Sepulveda, E. Ferrazzi, and T. Van den Bosch. Terms, definitions and measurements

to describe the sonographic features of the endometrium and intrauterine lesions: a consensus opinion from the International Endometrial Tumor Analysis (IETA) group. *Ultrasound in Obstetrics and Gynecology*, 35: 103–112, 2010. doi: 10.1002/uog.7487.

[46] H. Li, Q. Li, and M. Lu. Software Reliability Modeling with Logistic Test Coverage Function. *2008 19th International Symposium on Software Reliability Engineering (ISSRE)*, (4):319–320, Nov. 2008. doi: 10.1109/ISSRE.2008.51.

[47] L. Li, P. Khatri, T. K. Sigdel, T. Tran, L. Ying, M. J. Vitalone, A. Chen, S. Hsieh, H. Dai, M. Zhang, M. Naesens, V. Zarkhin, P. Sansanwal, R. Chen, M. Mindrinos, W. Xiao, M. Benfield, R. B. Ettenger, V. Dharnidharka, R. Mathias, A. Portale, R. McDonald, W. Harmon, D. Kershaw, V. M. Vehaskari, E. Kamil, H. J. Baluarte, B. Warady, R. Davis, a. J. Butte, O. Salvatierra, and M. M. Sarwal. A peripheral blood diagnostic test for acute rejection in renal transplantation. *American journal of transplantation : official journal of the American Society of Transplantation and the American Society of Transplant Surgeons*, 12(10):2710–2718, Oct. 2012. doi: 10.1111/j.1600-6143.2012.04253.x.

[48] J. Litchfield, J. Freeman, H. Schou, M. Elsley, R. Fuller, and B. Chubb. Is the future for clinical trials internet-based? A cluster randomized clinical trial. *Clinical Trials*, 2(1):72–79, Feb. 2005. doi: 10.1191/1740774505cn069oa.

[49] N. Llopis. Stepping through the looking glass: Test-driven game development (part 1), 2005. Accessed: February 2, 2014; archived by WebCite® at `http://www.webcitation.org/6N5zwMo4Z`.

[50] C. Lu, T. Van Gestel, J. A. K. Suykens, S. Van Huffel, I. Vergote, and D. Timmerman. Preoperative prediction of malignancy of ovarian tumors using least squares support vector machines. *Artificial Intelligence in Medicine*, 28(3):281–306, July 2003. doi: 10.1016/S0933-3657(03)00051-4.

[51] Y. K. Malaiya, N. Li, J. M. Bieman, and R. Karcich. Software reliability growth with test coverage. *IEEE Transactions on Reliability*, 51:420–426, 2002.

[52] J. Mercer. Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 209 (441-458):415–446, Jan. 1909. doi: 10.1098/rsta.1909.0016.

[53] C. A. Micchelli, Y. Xu, and H. Zhang. Universal Kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

[54] T. E. Oliphant. Python for Scientific Computing. *Computing in Science and Engineering*, 9(3):10–20, 2007. doi: 10.1109/MCSE.2007.58.

[55] OpenClinica. OpenClinica. `http://community.openclinica.com`, 2013. Accessed: January 12, 2014; archived by WebCite® at `http://www.webcitation.org/6Ma1mK0E9`.

[56] C. Parker. An Analysis of Performance Measures for Binary Classifiers. In *2011 IEEE 11th International Conference on Data Mining*, pages 517–526. IEEE, Dec. 2011. doi: 10.1109/ICDM.2011.21.

[57] I. Pavlović, T. Kern, and D. Miklavcic. Comparison of paper-based and electronic data collection process in clinical trials: costs simulation study. *Contemporary clinical trials*, 30(4):300–316, 2009. doi: 10.1016/j.cct.2009.03.008.

[58] P. Peduzzi, J. Concato, E. Kemper, T. R. Holford, and A. R. Feinstein. A simulation study of the number of events per variable in logistic regression analysis. *Journal of clinical epidemiology*, 49(12):1373–1379, Dec. 1996.

[59] N. L. M. M. Pochet and J. A. K. Suykens. Support vector machines versus logistic regression: improving prospective performance in clinical decision-making. *Ultrasound in Obstetrics and Gynecology*, 27(6):607–608, June 2006. doi: 10.1002/uog.2791.

[60] F. Provost, T. Fawcett, and R. Kohavi. The Case Against Accuracy Estimation for Comparing Induction Algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning1*, pages 445–453, 1997.

[61] M. A. Richards. The size of the prize for earlier diagnosis of cancer in England. *British Journal of Cancer*, 101:S125–S129, Dec. 2009. doi: 10.1038/sj.bjc.6605402.

[62] J. L. Schafer. Multiple imputation: a primer. *Statistical Methods in Medical Research*, 8(3):3–15, 1999. doi: 10.1177/096228029900800102.

[63] K. Schwaber and M. Beedle. *Agile Software Development with SCRUM*. Prentice Hall, 2001. ISBN 0130676349.

[64] R. Siegel, J. Ma, Z. Zou, and A. Jemal. Cancer Statistics, 2014. *CA: a cancer journal for clinicians*, 64(1):9–29, 2014. doi: 10.3322/caac.21208.

[65] M. Siniaalto and P. Abrahamsson. Does Test-Driven Development Improve the Program Code? Alarming Results from a Comparative Case Study. In B. Meyer, J. R. Nawrocki, and B. Walter, editors, *Balancing Agility*

and Formalism in Software Engineering, pages 143–156. Springer Berlin Heidelberg, 2008. ISBN 9783540852780.

[66] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, and F. de Bona. The SHOGUN Machine Learning Toolbox. *Journal of Machine Learning Research*, 11:1799–1802, 2010.

[67] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999. doi: 10.1023/A:1018628609742.

[68] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Leuven, 2002. ISBN 981-238-151-1.

[69] D. Timmerman, L. Valentin, T. H. Bourne, W. P. Collins, H. Verrelst, and I. Vergote. Terms, definitions and measurements to describe the sonographic features of adnexal tumors: a consensus opinion from the International Ovarian Tumor Analysis (IOTA) Group. *Ultrasound in Obstetrics and Gynecology*, 16(5):500–505, Oct. 2000. doi: 10.1046/j.1469-0705.2000.00287.x.

[70] D. Timmerman, A. C. Testa, T. Bourne, E. Ferrazzi, L. Ameye, M. L. Konstantinovic, B. Van Calster, W. P. Collins, I. Vergote, S. Van Huffel, and L. Valentin. Logistic regression model to distinguish between the benign and malignant adnexal mass before surgery: a multicenter study by the International Ovarian Tumor Analysis Group. *Journal of Clinical Oncology*, 23(34):8794–8801, Dec. 2005. doi: 10.1200/JCO.2005.01.7632.

[71] D. Timmerman, A. C. Testa, T. Bourne, L. Ameye, D. Jurkovic, C. Van Holsbeke, D. Paladini, B. Van Calster, I. Vergote, S. Van Huffel, and L. Valentin. Simple ultrasound-based rules for the diagnosis of ovarian cancer. *Ultrasound in Obstetrics and Gynecology*, 31(6):681–690, June 2008. doi: 10.1002/uog.5365.

[72] B. Van Calster, D. Timmerman, C. Lu, J. A. K. Suykens, L. Valentin, C. Van Holsbeke, F. Amant, I. Vergote, and S. Van Huffel. Preoperative diagnosis of ovarian tumors using Bayesian kernel-based methods. *Ultrasound in Obstetrics and Gynecology*, 29(5):496–504, May 2007. doi: 10.1002/uog.3996.

[73] B. Van Calster, D. Timmerman, I. T. Nabney, L. Valentin, A. C. Testa, C. Van Holsbeke, I. Vergote, and S. Van Huffel. Using Bayesian neural networks with ARD input selection to detect malignant ovarian masses prior to surgery. *Neural Computing and Applications*, pages 489–500, Sept. 2008. doi: 10.1007/s00521-007-0147-1.

[74] B. Van Calster, S. Van Huffel, D. Timmerman, E. Kirk, T. Bourne, and G. Condous. Towards a Clinical Decision Support System for Pregnancies of Unknown Location. *2008 21st IEEE International Symposium on Computer-Based Medical Systems*, pages 581–583, June 2008. doi: 10.1109/CBMS.2008.123.

[75] T. Van Gestel, J. A. K. Suykens, B. De Moor, and J. Vandewalle. Automatic Relevance Determination for Least-Squares Support Vector Machine Classifiers. In *European Symposium on Artificial Neural Networks (ESANN'2001)*, number April, pages 13–18, Bruges, Belgium, 2001.

[76] C. Van Holsbeke, B. Van Calster, L. Valentin, A. C. Testa, E. Ferrazzi, I. Dimou, C. Lu, P. Moerman, S. Van Huffel, I. Vergote, and D. Timmerman. External validation of mathematical models to distinguish between benign and malignant adnexal tumors: a multicenter study by the International Ovarian Tumor Analysis Group. *Clinical Cancer Research*, 13(15 Pt 1): 4440–4447, Aug. 2007. doi: 10.1158/1078-0432.CCR-06-2958.

[77] C. Van Holsbeke, B. Van Calster, T. Bourne, S. Ajossa, A. C. Testa, S. Guerriero, R. Fruscio, A. A. Lissoni, A. Czekierdowski, L. Savelli, S. Van Huffel, L. Valentin, and D. Timmerman. External validation of diagnostic models to estimate the risk of malignancy in adnexal masses. *Clinical Cancer Research*, 18(3):815–825, Feb. 2012. doi: 10.1158/1078-0432.CCR-11-0879.

[78] D. Van Schoubroeck, A. Installé, N. J. Raine-Fenning, D. De Neubourg, T. Van den Bosch, B. De Moor, T. Bourne, and D. Timmerman. Interobserver variability in the ultrasound diagnosis of congenital uterine anomalies. In *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology*, page 124, Copenhagen, 2012. doi: 10.1002/uog.11490.

[79] D. Van Schoubroeck, A. Installé, N. J. Raine-Fenning, D. De Neubourg, T. Van den Bosch, B. De Moor, T. Bourne, and D. Timmerman. Interobserver variability in the ultrasound diagnosis of polycystic ovaries using pattern recognition. In *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology*, page 88, Copenhagen, 2012. doi: 10.1002/uog.11490.

[80] V. Vapnik and A. Lerner. Pattern Recognition using Generalized Portrait Method. *Automation and Remote Control*, 24:774–780, 1963.

[81] B. Venners. *Inside the Java virtual machine*. McGraw-Hill, New York, New York, USA, 1996. ISBN 0079132480.

[82] A. J. Vickers and E. B. Elkin. Decision curve analysis: a novel method for evaluating prediction models. *Medical Decision Making*, 26(6):565–574, Dec. 2006. doi: 10.1177/0272989X06295361.

[83] H. Visser, S. le Cessie, K. Vos, F. C. Breedveld, and J. M. W. Hazes. How to diagnose rheumatoid arthritis early: a prediction model for persistent (erosive) arthritis. *Arthritis & Rheumatism*, 46(2):357–365, Feb. 2002. doi: 10.1002/art.10117.

[84] A. Votino, A. Installé, T. Van den Bosch, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, D. Timmerman, and C. Van Pachterbeke. Optimal ultrasound visualization of the endometrial-myometrial junction (EMJ). In *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology*, volume 40, pages 166–167, Copenhagen, 2012. doi: 10.1002/uog.11748.

[85] A. Votino, A. Installé, T. Van den Bosch, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, D. Timmerman, C. Van Pachterbeke, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, D. Timmerman, T. Van den Bosch, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, D. Timmerman, and C. Van Pachterbeke. The influence of patient characteristics on the image quality of the endometrial-myometrial junction (EMJ). In *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology*, volume 40, pages 62–63, Copenhagen, 2012. doi: 10.1002/uog.11747.

[86] A. Votino, A. Installé, C. Van Pachterbeke, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, D. Timmerman, and T. Van den Bosch. Optimization of the image quality of endometrial-myometrial junction (EMJ). In *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology*, volume 40, page 166, Copenhagen, 2012. doi: 10.1002/uog.11747.

[87] C. Walls and R. Breidenbach. *Spring in Action*. Manning Publications Co., Greenwich, CT, USA, second edition, 2007. ISBN 9781933988139.

[88] B. Walther, S. Hossin, J. Townend, N. Abernethy, D. Parker, and D. Jeffries. Comparison of electronic data capture (EDC) with the standard data capture method for clinical trial data. *PloS one*, 6(9):e25348, 2011. doi: 10.1371/journal.pone.0025348.

[89] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann series in data management systems. Morgan Kaufmann, second edition, June 2005. ISBN 0120884070.

Curriculum vitae

Arnaud Installé was born on December 21, 1973, in Brussels, Belgium. He obtained the degree of master of science in engineering at the KU Leuven Department of Electrical Engineering (ESAT), option micro-electronics.

He started his professional career in 2000, as a Quality Engineer at Filepool, then a Belgian startup company. Due to his Linux expertise, he quickly changed to a role in which he transitioned the company's existing small, Windows-based setup, to an extensible, fault-tolerant Linux cluster for cloud storage. Following the acquisition of Filepool by EMC^2 in 2001, in his new role as Senior Software Engineer, he designed and implemented features such as redundant storage and self-healing, which formed the defining functionality of EMC^2's Centera product, brought to market in 2002. He would continue working on various aspects of this product, including cluster extensibility and federation, until EMC^2 closed its Belgian development offices in 2009.

Seeking a new challenge, he started his PhD research in 2009, under supervision of Prof. Dr. Ir. Bart De Moor and Prof. Dr. Dirk Timmerman. His research interests include the development of the Clinical Data Miner (CDM) software framework for facilitating clinical diagnostic model research, using sophisticated kernel methods to enable automation of the machine-learning workflow. In 2011, he obtained the Outstanding Poster Presentation award at the Benelux Bioinformatics Conference 2011, for his poster "Clinical Data Miner – an Electronic Data Capture software framework that improves interrater agreement", as well as a Short Oral Presentation Award at the 21st World Congress on Ultrasound in Obstetrics and Gynaecology.

He co-authored an IWT-TBM funding proposal with Prof. Dr. Dirk Timmerman and Prof. Dr. Ir. Bart De Moor, which was accepted in 2013. In the context of this proposal, he will research machine-learning models for the diagnosis of endometrial cancer, and extend CDM's user interface to enable these diagnostic models to be used in clinical practice.

List of publications

A. Installé. "Clinical Data Miner – from Electronic Data Capture to machine-learning." *Technical Report* (2014).

A. Installé, T. Van den Bosch, B. De Moor, D. Timmerman. "Studying inter-rater agreement of sonographic video clips using Clinical Data Miner." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2013* (2013).

A. Installé, J. Suykens, B. Van Calster, T. Van den Bosch, J. Kaijser, B. De Moor, D. Timmerman. "Continually updating models for ovarian cancer diagnosis." *Technical Report* (2013).

A. Votino, A. Installé, T. Van den Bosch, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, D. Timmerman, C. Van Pachterbeke. "The influence of patient characteristics on the image quality of the endometrial-myometrial junction (EMJ)." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2012* (2012).

A. Votino, A. Installé, C. Van Pachterbeke, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, dirk, T. Van den Bosch. "Optimization of the image quality of endometrial-myometrial junction (EMJ)." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2012* (2012).

A. Votino, A. Installé, T. Van den Bosch, D. Van Schoubroeck, Y. Kacem, J. Kaijser, B. De Moor, D. Timmerman, C. Van Pachterbeke. "Optimal ultrasound visualization of the endometrial-myometrial junction (EMJ)." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2012* (2012).

A. Votino, A. Installé, N. Raine-Fenning, D. De Neubourg, T. Van den Bosch, B. De Moor, T. Bourne, D. Timmerman. "Interobserver variability in the ultrasound diagnosis of polycystic ovaries using pattern recognition." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2012*

(2012).

D. Van Schoubroeck, A. Installé, N. Raine-Fenning, D. De Neubourg, T. Van den Bosch, B. De Moor, T. Bourne, D. Timmerman. "Interobserver variability in the ultrasound diagnosis of congenital uterine anomalies." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2012* (2012).

A. Installé, T. Van den Bosch, J. Suykens, B. De Moor, D. Timmerman. "Comparing performance of Least-Squares Support Vector Machines versus logistic regression as patient group sizes increase." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2012* (2012).

A. Installé, T. Van den Bosch, J. Suykens, D. Timmerman, B. De Moor. "Clinical Data Miner – Electronic Data Capture software providing a data querying library allowing integration in data analysis software." *Proceedings of Medicine 2.0'12* (2012).

A. Installé, T. Van den Bosch, D. Van Schoubroeck, J. Heymans, L. Zannoni, L. Jokubkiene, P. Sladkevicius, L. Valentin, bart, D. Timmerman. "Showing pictograms in Electronic Data Capture (EDC) software improves interrater agreement." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2011* (2011).

A. Installé, D. Timmerman, B. De Moor, T. Van den Bosch. "Image Study Webapp – a web interface for imaging-based interrater agreement studies." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2011* (2011).

A. Installé, D. Timmerman, T. Van den Bosch, B. De Moor. "Clinical Data Miner – an EDC software framework that improves interrater agreement." *Proceedings of the Benelux Bioinformatics Conference 2011* (2011).

A. Installé, T. Van den Bosch, D. Van Schoubroeck, J. Heymans, L. Zannoni, L. Jokubkiene, P. Sladkevicius, L. Valentin, B. De Moor, D. Timmerman. "Clinical Data Miner – a web-based Electronic Data Capture framework for multi-centric studies with imaging modalities." *Proceedings of the International Society of Ultrasound in Obstetrics and Gynecology 2011* (2011).



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING (ESAT)
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10, bus 2446
B-3001 Leuven
arnaud.installe@esat.kuleuven.be
http://www.esat.kuleuven.be