A Mathematical Diagnostics Model for Autoimmune Disease

Guðný Anna Árnadóttir

Thesis of 30 ECTS credits

Master of Science in Biomedical Engineering

April 2014

A Mathematical Diagnostics Model for Autoimmune Disease

Guðný Anna Árnadóttir

Thesis of 30 ECTS credits submitted to the School of Science and Engineering

at Reykjavík University in partial fulfillment

of the requirements for the degree of

Master of Science in Biomedical Engineering

April 2014

Supervisors:

Bjarni V. Halldórsson, PhD
Associate Professor, Reykjavík University, Iceland

Björn Rúnar Lúðvíksson MD, PhD
Department of Immunology, Landspítali University Hospital, Iceland

Examiner:

Jón Guðnason, PhD
Assistant Professor, Reykjavík University, Iceland

A Mathematical Diagnostics Model for Autoimmune Disease

Guðný Anna Árnadóttir

30 ECTS thesis submitted to the School of Science and Engineering

at Reykjavík University in partial fulfillment

of the requirements for the degree of

Master of Science in Biomedical Engineering

April 2014

Student:

____

Guðný Anna Árnadóttir

Supervisors:

____

Bjarni V. Halldórsson

____

Björn Rúnar Lúðvíksson

Examiner:

____

Jón Guðnason

The undersigned hereby declares that this thesis, entitled **A Mathematical Diagnostics Model for Autoimmune Disease**, shall not be open to public access, or to distribution of any kind until the **30th of April 2019**. Following this date, the author grants permission to the Reykjavík University Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the project report, and except as herein before provided, neither the project report nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

____

Date

____

Guðný Anna Árnadóttir
Master of Science

Abstract

The topic of this thesis is the development of a diagnostic model for autoimmune disease. We present ten classifiers, based on the data mining of immunological blood test results that are specific to each disease. Our training sets consist of immunological blood test results from 1750 clinical attendees that presented symptoms of one or more autoimmune diseases. The sparseness of the blood tests that are performed leads to highly structured missingness of data. We develop a classifier that uses the distinction between meaningless missing values and meaningful missing values to lead to improved modeling of these incomplete sets of data. The implementation of this new approach to modeling autoimmune blood test data is described from the perspective of rheumatoid arthritis (RA) data. Several, competing data mining approaches were implemented for our data, of which the best results were obtained for our hybrid classifier of Decision trees and an inverse-distance-weighted k-Nearest neighbors (kNN) approach. Results indicate that our model, assuming a certain margin of error, is classifying up to 100% of the RA cases correctly, computed by a method of *leave-one-out* (LOO) cross-validation. The classifiers specific to the other nine diseases all average more than 90% of correctly classified cases, computed by LOO. New RA test cases were run through our classifier, which successfully classified all the cases correctly. The method of Random forests showed compatible results to our hybrid classifier, i.e. classified 97.91% of LOO RA cases correctly, but did not compare in structural transparency. Prototypes of a clinical solution system based on our work have been implemented on a web server.

Keywords: Disease advisor, Structured prediction, Mathematical model, MNAR, Hybrid classifier, Autoimmune disease

Útdráttur

Stærðfræðilegt Greiningarlíkan fyrir Sjálfsofnæmissjúkdóma

Umfjöllunarefni þessarar ritgerðar er þróun flokkunarlíkans fyrir sjálfsofnæmissjúkdóma. Við kynnum tíu flokkunarlíkön, sem hvert og eitt byggir á gagnanámi blóðprufuniðurstaða fyrir hvern sjúkdóm. Þjálfunargögn okkar eru samansafn blóðprufuniðurstaða frá 1750 skjólstæðingum heilsugæslu sem höfðu klínísk einkenni sjálfsofnæmissjúkdóma. Þar sem stórum hluta slíkra blóðprufa er yfirleitt sleppt fyrir úrvinnslu þá innihéldu gögnin mikið magn tómra gagnasvæða. Við kynnum nýja aðferð til gagnanáms sem byggir á aðgreiningu þeirra tómu gagnasvæða sem hafa enga merkingu annars vegar og hins vegar tómra gagnasvæða sem hafa merkingu. Þessari nýju aðferð er lýst út frá gögnum fyrir iktsýki (RA). Margar vel þekktar gagnanámsaðferðir voru útfærðar fyrir gögnin okkar en bestu greiningarniðurstöðurnar fengust með því að setja líkanið fram sem blendingslíkan flokkunartrjáa og andhverfu-fjarlægðar-vigtaðra k-Næstu granna (kNN). Niðurstöður okkar gáfu til kynna að með því að gera ráð fyrir ákveðnu öryggisbili þá flokkaði líkanið okkar allt að 100% RA tilvikanna rétt, út frá reikningum sem byggðu á einum-sleppt (LOO) sannprófunaraðferð. Hin, níu flokkunarlíkönin fengu öll niðurstöður upp á meira en 90% rétt flokkaðra tilfella, metin með LOO. Ný prófunartilfelli fyrir RA voru keyrð gegnum flokkunarlíkanið okkar og náði það að flokka öll tilfellin rétt. Aðferð Slembiskóga sýndi sambærilegar niðurstöður og okkar líkan, eða um 97.91% rétt flokkaðra tilfella, metin með LOO sannprófunaraðferð, en sú aðferð stóðst hins vegar ekki samanburð á gegnsæi líkananna. Fyrsta útgáfa af hugbúnaði sem byggir á okkar niðurstöðum hefur nú verið sett upp á vefsvæði.

Lykilorð: Sjúkdómsgreinir, Uppbyggð gagnasvæði, Stærðfræðilíkan, MNAR, Blendingslíkan, Sjálfsofnæmissjúkdómar

Acknowledgements

I would like to express great thanks to my supervisor Bjarni V. Halldórsson, for giving me opportunity, guidance, encouragement and cooperation throughout this project. Special thanks go out to Erla Björg Skúladóttir, my aunt, for proofreading. Finally, I would like to thank my thesis committee members Dr. Björn Rúnar Lúðvíksson, for helping me with the theoretical background and data, and to Jón Guðnason for introducing me to data mining and providing invaluable knowledge in the field.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Autoimmune diseases are a group of systemic, inflammatory disorders caused by hyperactivity of the immune system [1, 2]. The disorders are commonly associated with a range of autoantibodies that are secreted into the blood by the immune system [2]. Initial clinical presentations of autoimmune diseases have been known to overlap substantially, both within the defined group of autoimmune disorders but also with non-rheumatological disorders. Therefore, the detection of autoantibodies in the bloodstream has been accepted as a preferable approach to relying solely on clinical symptoms for the diagnosis of these diseases [3, 4]. However, there are dozens of available tests for measuring these autoantibodies, most of which can be performed in a number of different ways [1]. Due to this, the diagnosis of autoimmune disorders generally consists of lengthy and unreliable analytical processes, producing a large number of results that are uninterpretable to others than experts in rheumatology [3, 4]. Although individual autoimmune diseases can be considered rare, around five percent of the population of Western countries suffer from some variation of autoimmune disease [2]. The need for a speedy and accurate diagnosis of these diseases is therefore pressing.

The focus of this thesis will be on the development of a classifier that uses known methods of data mining in new ways, to predict diagnostic outcomes of patients suspected of suffering from autoimmune diseases. A clinical support system that bases its diagnostic prediction on the classifier we describe here has already been established as a prototype. The first and second versions of this prototype are presented in the Appendix of this thesis. The classifier's development and training process were based on available data from ten autoimmune diseases; rheumatoid arthritis (RA), systemic lupus erythematosus (SLE), Sjögren's syndrome, mixed connective tissue disease (MCTD), systemic sclerosis, polymyositis/dermatomyositis (PM-DM), Wegener's granulomatosis, Churg-Strauss syndrome, microscopic polyangiitis and antiphospholipid syndrome. The classification method was developed based on data obtained from healthcare attendees suffering from these disorders. This produced specific classifiers for each of the ten diseases. RA has the highest prevalence of the above mentioned disorders and the criteria regarding RA diagnosis are among the best established criteria for autoimmune diseases [2, 5]. Experiments that composed the developmental process of the classifier were therefore carried out for the data set that originated from clinical cases of RA. The methodology for these experiments is described from the perspective of the RA data, and results obtained from various implementations of the RA-based classifier are presented in Chapter 4. Currently, our final classifier for RA shows successful predictions for 100% of clinical cases, produced by a *leave-one-out* cross-validation method that assumes a certain margin of error. Furthermore, new test cases that were obtained for final assessments of the classifier's performance, showed correct diagnostic outcomes for all available cases. The same approach as the one described for RA in this text has been used for the development of the individual classifiers belonging to the nine other autoimmune disorders. The results belonging to those individual classifiers are presented in the Appendix of this thesis, with *leave-one-out* results showing more than 90% correctly classified cases for all diseases.

1.1 Motivation

The main difficulty of modeling medical data is the complex nature of such data sets. Medical data generally don't possess a formal, mathematically defined structure into which the information involved with the data can be organized [6]. Most unprocessed medical data sets are heterogeneous; they may be collected from images, patient interviews, laboratory data and physician-based interpretations of presented symptoms [6]. In many cases, combined information from all these different sources is necessary to obtain a final diagnosis. An additional problem widely encountered in medical data mining is the large number of missing values that is bound to accompany any larger medical data sets [6]. Most medical data are acquired as a byproduct of standard diagnostic processes and are, as such, not collected in an organized, mathematical manner. Values are also commonly omitted in medical data sets due to technical, economical or ethical reasons [6]. Immunological blood tests are an example of this. They are known to involve labor-intensive and very costly analytical processes [1]. Therefore, in most cases, only about a half to one third of these tests are ordered and performed, leading to a substantial missingness of data [1, 2]. The variables that compose the sets may furthermore be hard to interpret, especially for non-experts in the field [2]. A demand for an expert system that can model such challenging data sets, and assist physicians in the diagnostic process for autoimmune diseases has therefore arisen.

The aim of this thesis will be to describe our attempt to construct a classifier for autoimmune diseases that can handle the structured data sets that accompany these diseases. We show that we can build a classifier that identifies whether an individual suffers from an autoimmune disease. Further we show that the classifiers we build constitute the best approach to modeling this type of data.

1.2 Project Description

The project started out as an approach to mining already available autoimmune data. This was to be carried out through the use of an interactive programming platform, to possibly identify new trends in the data or develop a novel approach to modeling this type of data. However, the project quickly developed into something more. An idea had been evolving amongst Icelandic experts in rheumatology; to develop a diagnostics model that could knowledge-map autoimmune data into providing a prediction for disease. The complex nature of the serological data available for this purpose did, however, require the development of new implementations for known data mining methods. We describe these new implementations, which lead to the final outcome of an expert system able to mimic the decision-making processes of experts in rheumatology. The platform that was employed throughout the development of our model was the R programming interface [7]. Several built-in packages were utilized in the development of the model, although the final version did not directly rely on any already established code from such packages. The reason for this was the nature of the data; the sets of serology results of the various autoimmune tests were unique in that previously defined rules had determined which tests were to be performed and which were deemed unnecessary. Missing data in the sets were substantial, and there were different reasons for different types of missing data entries. This invoked the need to address the data mining in a newfound manner, by developing an algorithm that could model these different types of missing data in the best possible way.

For most data mining problems, different approaches may result in similar accuracies. However, in some cases few algorithms may perform unambiguously better than others. We therefore subject our autoimmune data to analysis by several competing data mining methods, including Bayesian networks, Decision tree learners and k-Nearest neighbors classification.

1.3 Disposition of the Thesis

The disposition of this thesis is as follows: In Chapter 2, a theoretical background of the methods that were used in the development of the classifier is explained. In the following chapter, Chapter 3, the methodology of the model development is described. Results are discussed in Chapter 4, with a twofold approach; one reveals results that are based on the *leave-one-out* method (to be discussed) and the other focuses on results obtained from running new test cases through the model. Finally, the Appendix contains a list of *leave-one-out* results from the final classifiers belonging to each autoimmune disease, and a demo of the web-based interface of a clinical support system that is, in part, based on our classifiers.

Chapter 2

Theoretical Background

2.1 Artificial Intelligence in Medicine

Medical data sets have been considered an ideal field of application for artificial intelligence (AI) techniques ever since AI was first developed as a tool for data analysis [8]. Data accumulation in healthcare is considerable; the amount of patient data has accumulated exponentially in recent years and as a result modern day hospitals possess large amounts of a wide variety of data sets [9]. Data mining, a subfield of AI, can be described as the process of analyzing large amounts of data with the intention of extracting meaningful information from it [10]. The data mining process involves selecting useful data from a set of input variables, cleansing the data (which may involve correcting for errors or handling missing data), transforming the data (such as mapping individual variables in order to make them useful, weighting others, representing some as binary values etc.) and finally mining the data to discover interesting patterns or trends within the sets [10].

Considering the large amounts of data available in healthcare, it was a somewhat natural continuation in the development of AI to use data mining techniques as a means to support and improve physician-based diagnostic processes [8]. Artificial intelligence in medicine has since been established as an interdisciplinary field, a branch of applied computer science that focuses on the analysis of complex medical data in order to reveal useful information for diagnostic purposes [11].

2.1.1 Medical Classifiers

The most important aspects of a medical classifier involve good overall performance of the underlying algorithm, i.e. with diagnostic accuracy at least as good as that of a physician, the ability to deal with missing data, and the transparency of diagnostic knowledge [8]. This last aspect refers to the underlying classification system being sufficiently transparent for a physician to be able to analyze and understand the knowledge that the classifier generates. It has been estimated that for a wide variety of clinical conditions, the rate of misdiagnosis averages about ten percent in Western countries [12]. Misdiagnosis is therefore considered a relatively common problem; a problem that can lead to adverse health-related effects [13, 14]. Risks associated with misdiagnosis can range from emotional and economic implications to examples of renal failure or pulmonary fibrosis [14]. Computer aided diagnostics offer a realistic solution to this problem [13]. In fact, diagnostic models, or classifiers (the two terms will be used interchangeably in this text), can substantially improve the reliability of diagnosis. Furthermore, they present the possibility of increasing diagnostic speed along with the accuracy of prediction.

A common characteristic of medical data sets is the large number, or high dimensionality of input variables [6]. This induces a problem to classification, the so-called "curse of dimensionality", which refers to complications that may arise for data mining in multiple dimensions [15]. Data sets are generally considered low-dimensional sets for when the number of input variables is less than ten, $d \leq 10$ [16]. For a classification to be considered practical, especially in classification of medical data, the diagnostic models have to be able to handle high-dimensional data without being affected by dimensionality issues. In fact, many popular data mining methods are based on addressing the problem of high dimensionality directly [8]. Some of the most promising models for medical data analysis are approaches that break the problem of mining high dimensional data into sub-problems. They accomplish this by dividing the input data into smaller and smaller sets, thereby reducing the dimensionality of the data [6]. Alternatively, dimensionality reduction can be achieved by manually selecting relevant input variables [6].

Those classification methods that have shown greatest promise in dealing with medical data are statistical methods for classification, especially Bayesian classifiers, Artificial neural networks (ANN's), Nearest neighbor (NN) algorithms and Decision tree learners [8, 17]. Out of these, the methods that furthermore possess a sufficiently transparent structure for applications in clinical diagnostics, are the approaches of Bayesian-theory networks and Decision trees [15, 18]. Tree-based models have been widely accepted and used within the field of medical diagnostics, the most recent applications involving the diagnosis of heart disease [17, 19, 20].

The data mining methods that we have emphasized in the process of developing our final classifiers for autoimmune disease are Decision tree learners and Bayesian networks, since these were believed to be able to constitute an adequate backbone structure for a diagnostic model. We furthermore explored the possibility of intertwining these approaches with other methods that have been known to perform well in the mining of medical data; the method of k-Nearest neighbors (kNN) and Discriminant function analysis [17, 21].

2.1.2 Missing Data

In applications of data mining, incomplete data sets are extremely common [22]. The so-called missingness of data can have a range of origins, such as measurement errors, a deliberate omission of measurements, device malfunctions, etc. [16]. There are two conventional approaches that are widely accepted and used in dealing with missing data [23]. The first approach involves deleting all variables that correspond to a defined factor, if at least one of the variables is missing. The second approach addresses the problem by imputing new values for all missing variables [22, 23]. Imputation can be an ideal solution if the fraction of missing data is sufficiently small, but for larger proportions of missing entries, or if the missing data entries originate from different causes, the method begins to produce relevant errors [16]. Furthermore, it is important to differentiate between various contexts of missing data. On the one hand there exists missing data at induction time (i.e. in the training of the classification model) and on the other, missing data at prediction time (when using the model) [22].

Most medical data sets are somewhat incomplete; missing or meaningless entries commonly comprise the majority of the sets. Research on missing data in medical databases has to date primarily been concerned with data missing at induction time, i.e. in the training of the classification model [22]. Furthermore, missing values in medical data sets commonly occur due to known, and possibly deliberate reasons [6]. If, for example, a physician decides to omit a patient's measurement because the patient's state is unreliable, that missing data value is distinctly non-random. Such non-random missingness of data has a different meaning than values that have no meaning at all, i.e. values that are missing completely at random [24]. To this extent, two types of missing data can be defined. Data is considered to be missing at random (MAR) if the probability of the missing values is independent of the set of data expected to contain missing variables [23].

$$P(Y_{missing}|Y, X) = P(Y_{missing}|X) \tag{2.1}$$

Here Y is the part of the data set, X, for which missingness is to be assessed. Y consists of both observed and missing variables, $Y = \{Y_{observed}, Y_{missing}\}$ [23, 24]. If Equation 2.1 is not fulfilled, the variable is defined as missing not at random (MNAR).

MAR entries are generally much easier to deal with, their absence does not involve any hidden information and their values can be omitted during the mining itself [23]. If the missing data involved is MNAR the complexity of the mining problem is bound to increase. The knowledge that is associated with the entries cannot simply be disregarded during the mining process, and neither can the possibility be ignored that these entries might be influential to the final diagnosis of a patient.

2.1.3 Our Classifier

We decided to base our approach to classification of autoimmune disease data on the theory of supervised learning. Learning is the process of estimating a systemic structure of some given data by using the data as a basis [10]. The computer can accomplish this through two approaches; supervised and unsupervised learning. Applications in which the learning process is based on training the classification model with a previously defined set of data consisting of input variables and a target vector, are known as supervised learning applications [15]. Our model uses training data that consists of examples of possible cases along with their corresponding diagnostic values as target variables. The problem was therefore directly identified as a supervised learning problem, where the learning is based on a process of feeding the classifier with established cases of autoimmune patients.

Applications of data mining in medicine have primarily been based on a type of classification known as hard classification [13, 25]. Hard classifiers directly assign a new instance to the suitable class, by estimating distinct boundaries between classes [25]. This can be advantageous in some clinical applications, e.g. for when a simple result of "sick" versus "not sick" is desired. Another approach to classification, known as soft classification, is based on estimating the class conditional probabilities associated with a new input data point [25]. The new point is assigned to the class that shows the greatest conditional probabilities for that particular point [25]. Soft classification can prove useful for when an assessment is needed of how close to, or far from a specific diagnostic result the output of the classifier is [10]. Soft classifiers can output a value from a range of continuous variables, rather than a single class, as a regression of sorts [25]. This way, an estimate can be obtained as to how close the delivered results of the model are to the target values. For our data mining problem, soft classification provided an extremely convenient approach. If the problem would have been approached as a hard classification problem, estimating the performance of the model would have consisted solely of whether or not the delivered results fit the previously defined classes: "Sick" or "not sick". However, the original estimates, or target values, that were provided with our data sets consisted of values that were highly subjective to the opinion of experts. Therefore, the knowledge of how close to, or how far from these estimates the output of the classifier lies can give a more meaningful result than the simple "correctly classified" versus "misclassified" result.

Immunological blood tests are very often omitted due to high costs, complex recovery of results, etc. [1]. Much of the omitted tests constitute meaningless, MAR variables. However, the data may be missing from different causes, and for autoimmune laboratory results this is exactly the case. A set of specific reasons or rules have been defined for processing the serology results; these indicate which omitted measurements are completely meaningless and which were omitted to avoid the high cost and complexity associated with their recovery. This set of rules thus distinguishes between meaningless missing data, MAR data, and meaningful missing data, MNAR. The most important distinction lies in the fact that the MNAR data may indicate the probability of the unmeasured factor being found in the blood, while MAR values contain no such information [23]. We attempt to utilize these MNAR variables as additional sources of information for the development of our classification model. The mathematical model that we introduce is furthermore based on the mining of data sets that contain missing variables both at induction and prediction times, i.e. both training and test sets are composed of structured data. Our training sets involve missing entries at the induction time of the model, since they are mostly composed from the blood test results of clinical attendees. Further, it can be assumed that the blood test results of a new individual who is yet to be classified will consist of a structured data set. Our model therefore has to take on prediction-time missing entries as well. Prediction-time missing data is an aspect that has not been as popular in recent model development [22].

The main contribution with our work is therefore the modeling of autoimmune disease data that is composed of structured sets, including MNAR data, both in induction and prediction.

2.2 Graphical Models

Graphical models are based on diagrammatic representations of the variables of the model, an aspect that makes easily explainable and transparent. We wanted the structure of our classifier to be similar to real-life decision-making processes, and a transparent model structure was therefore imperative. The choice of which graphical model to apply was, however, not clear. This chapter describes the two different graphical modeling approaches that were applied to our autoimmune data. Firstly, the graphical method of Bayesian networks will be discussed, followed by an outline of Decision tree methods.

2.2.1 Bayesian Networks

Probabilistic graphical models are a well-known approach to classification. They involve applying simple probabilistic methods to random variables. One of the most common algorithms in probabilistic graphical modeling is the method of Bayesian networks [26]. Bayesian networks have commonly been employed in biomedical classifiers. Due to their ability to deal with incomplete data sets, they have proved especially useful in probabilistic expert systems for medical diagnosis [27, 28, 29]. The Bayesian networks approach involves probabilities being calculated for each relevant random variable in the data set. The calculated probabilities are then linked to probabilities of related variables in a network of random variable nodes that are conditional on each other [30]. The network begins with one, or a few, parent nodes that are variables assumed to have the highest correlation to the target values of the data set in question. These are the variables that influence the outcome of the model the most. The parent nodes lead to several child nodes, whose values are dependent on the values of the parent nodes. The probabilities that are calculated for these child nodes are therefore conditional probabilities [30].

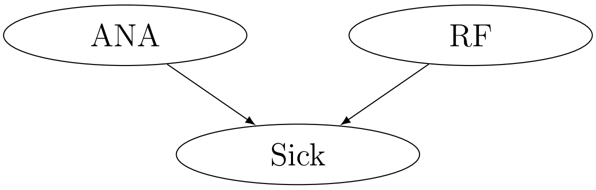


Figure 2.1: Basic elements of a Bayesian network.

Each node in a network of this sort contains the probabilities of a variable that is a determining factor to the final outcome of the classification model. Figure 3.1 shows a simplified idea of a Bayesian network, where two blood tests that are commonly performed for the diagnosis of RA (measurements of antinuclear antibodies (ANA) and rheumatoid factor (RF)) lead to the diagnosis of a patient as "Sick". However, the level of complexity of Bayesian networks can easily become very high, the consideration of which should be kept in mind when developing such a model. Yet, the calculations on which the networks are based are surprisingly simple; they are based on the multiplication rule of probability [15, 24].

$$p(X_1, X_2) = p(X_2|X_1)p(X_1) \tag{2.2}$$

where X_1 and X_2 represent two independent factors, and the classical Bayes theorem [15, 24]:

$$p(disease|data) = \frac{p(data|disease)p(disease)}{p(data)} \tag{2.3}$$

Equation 2.3 indicates how the conditional probability of disease can be obtained by probabilistic calculations for the available data. In order for such a model to be applicable to incomplete data sets, the probability calculations for each node must, however, be possible even if the parent node is a missing entry. A solution to this can be obtained by calculating both conditional and unconditional probabilities for each node, and then using either outcome depending on whether or not the values that this particular node is dependent on are available or not [26].

2.2.2 Decision Trees

Decision trees are generally preferred to other graphical techniques because of the straightforward readability of their structure and the efficiency of training and evaluation of their outcome. Tree-based models rely on partitioning the input space, i.e. data sets, into smaller subsets of data mining regions. A specific modeling method can subsequently be applied to each region, thereby breaking the mining problem down into smaller problems and furthermore leading to a more accurate result for each subregion [15]. Decision trees therefore constitute a highly advantageous approach for data as complex as autoimmune serology results.

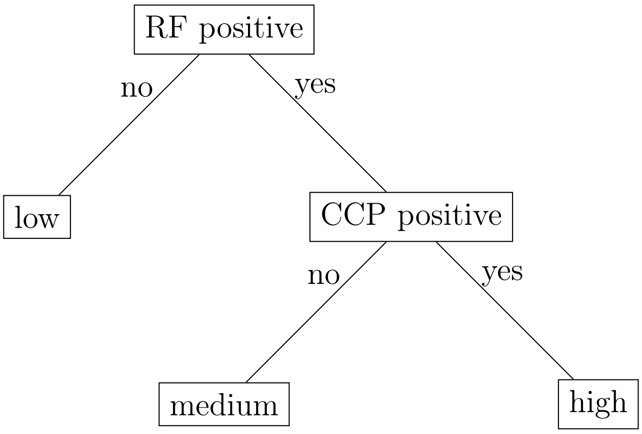


Figure 2.2: An example of a simple Decision tree for RA.

Partition sites of a Decision tree can be either multi-branch splits, if the split parameter leads to a number of different outcomes, or binary splits, if the parameter has two possible outcomes [10]. A binary tree is portrayed in Figure 2.2 as a simplified application of a Decision tree for RA. Classification and regression trees (CART) are a type of binary Decision trees. Breiman et al. developed the first CART algorithm in 1984, a tree structure that was able to model both categorical predictor variables (classification) and continuous ones (regression) [31]. The CART approach has since become increasingly popular for all fields of data mining applications. A wide variety of tree-based models are in existence. A model that best fits the data mining problem at hand must be carefully chosen in each case, whether it is a pure classification tree, a regression tree or a mixture of the two.

2.3 Discriminant Function Analysis

Discriminant function analysis (DA) is a method of classification that involves predicting the class label, or posterior probabilities, of a data point [32]. The labeling is based on prior probabilities and a certain discriminant equation. This discriminant equation returns a numerical value that is calculated from the input data point and specific weights, which are distinctive features (internal variances, mean value etc.) of a certain class, from a number of previously defined class [32]. The equation yields a linear function of the data point, which indicates within which class the data point would be best situated [32]. An equation is constructed for each defined class, based on the part of the training data that belongs to the class in question. The best-known approach of discriminant functions produces a strictly linear division, or boundary, between the classes of the model, and has a close relation to linear regression analysis. This approach is also known as Linear discriminant functions analysis (LDA), and is quite simple in its implementation. However, LDA assumes a common covariance matrix[1] for all classes and may therefore be limiting in applications to real problems; e.g. for when the amount of missing data is relevant to the modeling approach being taken. Another popular DA choice, which furthermore assumes varying covariance matrices between classes, is the Quadratic discriminant function analysis (QDA). As the name implies, this method produces a quadratic boundary between classes. When discriminant functions are used in multi-class classification, the results obtained from fitting the input data point to each discriminant equation are evaluated, with the highest outcome indicating the best fit. The data point is then classified with the class whose discriminant equation returns the highest numerical result [32].

2.4 *k*-Nearest Neighbors

The method of k-Nearest neighbors (kNN) is a widespread technique for estimating density of data, i.e. finding data points that show definite similarities and can therefore be assumed to belong to the same cluster of data points [10]. These similarities, most commonly defined as metric distances between data points in Euclidean space, are used to situate a new data point (in our case, a new patient) within the correct class [33, 24]. Thus, some k nearest values are found and the new data point classified with them. Most implementations of the classical kNN use *Euclidean* distances as a basis for the calculations. In cases of highly correlated data, *Euclidean* distances may however produce biased results, and a choice of basing the calculations on the so-called *Mahalanobis* distance may be considered instead [33].

The main issue with kNN involves establishing the number of data points to be taken into account for each region. If the data points are only one or two, the classification of a new point may become biased, especially if the set has a high variance [24]. The number of points also determines the degree of model smoothing[2], so it can neither be too small nor too large [15]. There are many approaches to selecting the proper value for k, the simplest one being to run the algorithm several times with different values and choose the value for k that results in the best performance of the classification model [34, 35]. Nearest neighbor approaches generally don't rely on previously known probability distributions, i.e. they don't make any specific assumptions about the probability distributions that may or may not dominate the available data. kNN

[1] A covariance matrix is a matrix that contains covariances between all elements of a given set of data [15].

[2] Smoothing refers to the complexity of a classification model, in kNN this relates to the size of the defined "neighborhood" [24].

does furthermore not assume a fixed region of space to which the modeling will be constricted to, but rather defines the amount of relevant, available data points [15]. Therefore the size of the input space can undergo posterior change [15]. For these reasons and others, kNN was considered an ideal approach for the data set of our study.

2.4.1 kNN Metrics

The accuracy of kNN classification is highly dependent on the metric used for calculating the distances between an input data point and points of the training set [35]. Nearest neighbor estimations for standard implementations of kNN are based on *Euclidean* distance calculations:

$$D_E = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.4}$$

where x_i and y_i are the standard variables between which a distance is to be evaluated, $x_i = \{x_1, x_2\}$ refers to points of the training data ($x_i = \{x_1, x_2, x_3, ...\}$ for higher dimensions) and y_i to a new point that is to be classified [24]. In kNN these distances are assessed for the entire set of training data and the k smallest distances define the neighbor points with greatest similarity to the new data point, subsequently classifying the new point with the most popular classes amongst these neighbors. The *Euclidean* metric is the best-known approach to estimating distances, both in general mathematical applications and in data mining. However, *Euclidean* distances don't take any statistical regularities that might be presented within the data into account [35]. For cases where trends and correlations are evident in the underlying set, a method that identifies this statistical information can result in a better performance of classification [35].

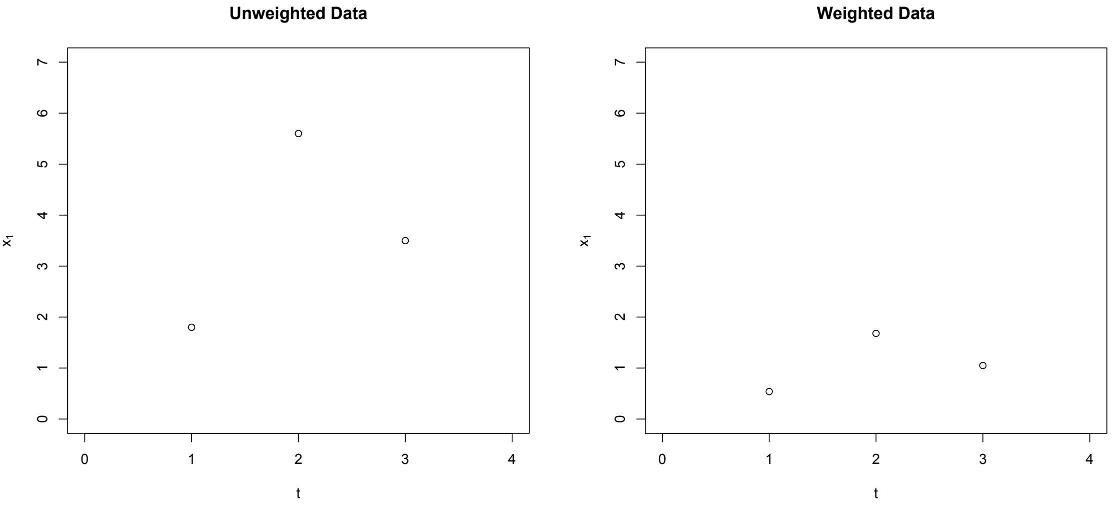
The *Mahalanobis* distance accounts for possible internal correlations. The distance constitutes a measure between two data points in the space defined by relevant features [36]. It accounts for unequal variances in the data, as well as correlations between different parameters. Therefore, it provides an unbiased evaluation of the distances by using a weight matrix, the covariance matrix [24]. The *Mahalanobis* distance is calculated in the following way:

$$D_M = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^T W (x_i - y_i)} \tag{2.5}$$

Here, W is the covariance matrix. Only when the parameters are uncorrelated, i.e. when W becomes the identity matrix, will the distance computed by the *Mahalanobis* distance metric be identical to the *Euclidean* distance [36].

2.4.2 kNN Weighting

kNN may need specific implementations for when the training data are of a variable nature, and specifically if structured data become relevant for the data sets in question. A method that is commonly employed to avoid the variable nature of data involves weighting the variables of the input space with a specific, predetermined weight, $w_i \geq 0$. This method is known as a feature-weighted kNN [37]. Common methods used to achieve a feature-weighted kNN are e.g. adding a fixed weight to the parameters of each feature of the input space. Figure 2.3 illustrates this, we can assume that the x_1-axis represents one feature from our data set and t the target vector (result variables). If a certain weight, which is fitting to that exact feature, is added to the variables belonging to the feature x_1, then the variables are brought into closer proximity in the feature space.



(a) Unweighted data points

(b) The feature x_1 multiplied by a scaling factor of 0.3

Figure 2.3: Feature-weighting in kNN.

The distance metric for feature-weighted kNN simply becomes:

$$D_{FW} = \sqrt{\sum_{i=1}^{n} w_i(x_i - y_i)^2} \tag{2.6}$$

where w_i represents parameter weights, which can be a form of standardization values that normalise the parameters to a range of $[0, 1]$, or correlations between specific parameters and the target vector [37].

Another implementation has proved beneficial for applications of kNN, especially when the distances of the neighbors vary widely and closer neighbors are considered more significant to the final prediction. This is a method that involves weighting the neighbors with an inverse of their distances; therefore known as inverse-distance-weighting [37]. The feature-weighted distances are first calculated for a defined number (k) of neighbors, and an added weight is then added to those distances to emphasize the data points closest to the new point.

$$w_j = \frac{1}{D(x_j, y_j)} \tag{2.7}$$

w_j represents the inverse-distance-weights and $D(x_j, y_j)$ the distance between the new data point, y_j, and a point in the training data, x_j [37]. Using this method of inverse-distance-weighting can be convenient for sparse data, or structured data sets [37].

A possible downside to classification by kNN is the high cost that the classification of new instances entails [34]. This is due to the fact that the most computationally demanding procedures of the method, i.e. the distance calculations, all take place at prediction time rather than induction time. It has therefore been speculated that kNN works best when employed for regions of smaller amounts of data, or combined with other modeling approaches [36].

2.5 Auxiliary Methods for Classification

Machine learning applications commonly rely on auxiliary methods for purposes such as estimating errors involved in the approach that is being implemented, as tools for handling missing data etc. The most important tool that we employed was an algorithm that served as a method of imputation for our missing values. This was the *expectation-maximization* (EM) algorithm [38]. Another auxiliary method that was used extensively throughout the development of our classifier was a method for performing cross-validation; the *leave-one-out* (LOO) approach.

2.5.1 The EM Algorithm

Data mining applications frequently have to deal with different problems of estimation. The most commonly encountered estimation problem is probably the estimation of the mean of a signal [15, 17]. Other common estimations involve finding values for missing data in incomplete data sets [22]. The EM algorithm is ideally suited to problems of this sort. It is a general method of finding a maximum likelihood estimate for available parameters of an input data set, with the goal of using the estimate as an imputation value for missing variables [17]. The likelihood function that the algorithm uses does not have a closed-form formula that can be differentiated and optimised. The likelihood is rather obtained through an iterative procedure; first an expectation (E) is assumed by evaluating posterior probabilities for the likelihood parameters, and then a maximization (M) is performed, where the parameters are re-estimated [15]. This way, the total likelihood increases and finally a maximum likelihood estimate is obtained [15]. The estimate can be used as an imputation value, e.g. as a numerical representation of the information that is hidden within MNAR entries.

2.5.2 Cross-Validation and LOO

To incorporate the testing of our model with the ongoing development of the classification methods, some sort of test data was needed. Blood test data from new patients was however unavailable at the time of the development. In such cases, a special case of cross-validation can be applied to the classification model in question, in order to obtain preliminary results that can serve as indicators of the performance accuracy of the model. One of the most widely employed methods of cross-validation is the so-called *leave-one-out* approach [13]. The method is based on the reduction of a data set of a total of n entries, by one data point, which is used as a test case for the method. The remaining $n - 1$ data points are then used as training data for the assessment of the performance of the model [13]. LOO can be applied for the whole set, thus providing a type of test data with size equal to that of the training set. This way relatively unbiased, preliminary results are obtained. For smaller data sets, of less than 100 data points, results from LOO cannot be considered reliable since these instances may invoke overfitting of the model to the limited number of cases the learning is based on [23]. Still, for larger sets of data, computations involved in LOO cross-validation become increasingly demanding, so LOO works best for intermediate-sized ($n \simeq 100 - 500$) sets of data [13].

2.5.3 Prediction Accuracy and the Mean Squared Error

Prediction accuracy and error estimation constitute a specific field in data mining [23]. Estimating the accuracy that a classifier provides is important to the process of evaluating future performances of a classifier, as well as for selecting the model that best suits the classification task at hand [39]. For pure classification models, the performance can be estimated by examining the misclassification rate of the model. The misclassification rate should preferably be minimized for the model in question; a so-called confusion table is often used to monitor the misclassification rate and make adjustments to the model according to the lowest possible rate of misclassification [15]. For regression models the mean squared error (MSE) is more commonly applied for estimating model performances. The method is presented as a measurement of the deviation of the model output from the provided target vector [39]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\{y(x) - t\}^2 \tag{2.8}$$

where $y(x)$ is a regression vector outputted by the model, and t is the target vector provided with the training data. The lower the value obtained from MSE calculations, the closer the model is to the desired output [39].

2.6 Random Forests

A fairly recent approach to several learning algorithms has become increasingly popular in recent years; these are the so-called ensemble learning methods [40]. Significant improvements in classification performances have especially been shown for implementations of ensemble Decision tree learners, compared to stand-alone classification trees [41, 42]. The Random forests algorithm, initially developed by Breiman and Cutler in 2001 (as a Fortran code), is a method that involves growing an ensemble of trees and letting them vote amongst themselves for the most prevalent class [41]. Each tree of these forests is constructed using random subsets of the input data [40]. Separator variables for a tree, i.e. the variables defining the branching sites, are randomly selected and restricted for each split in the tree [42]. This way the contribution of each separator variable to the final prediction can be better examined [42]. Two of the main parameters to influence the performance of Random forests classifiers are the number of trees in the forest, and the number of randomly preselected separator variables [42]. According to the Random forests Fortran-based implementation, available as an R-package interface, the default number of trees in a forest is 500 and the number of separator variables is five [43]. These can be optimized through cross-validation procedures, although a generally accepted approach is to increase the number of trees for increasing sizes of input data [43]. The trees are grown using a classification and regression tree methodology, and normally without pruning[3] [42].

Applications of Random forests for medical data mining have been growing more common in recent years [44, 45]. The results of a Random forests method can vary from run to run, so the method is considered a truly random statistical method, a fact that can be both beneficial and disadvantageous for applications in diagnostic modeling [42]. However, Random forests have been shown to be extremely robust to overfitting, and are therefore convenient for a variety of applications [40].

[3] Pruning is commonly employed in tree growing, a bottom-up procedure that shortens the tree to find an optimal number of terminal nodes [24][31].

Chapter 3

Materials and Methods

3.1 The Data

3.1.1 Data Gathering and Establishing Class Labels

The training data for our classifier consisted of blood test results from 1750 clinical attendees suspected of suffering from one or more autoimmune diseases, along with some predefined cases to represent the more uncommon symptoms of autoimmune disorders. With each and every suspected autoimmune case, a great deal of serological information is presented. Dozens of measurements can be performed in order to detect various autoantibodies from blood samples, and experts in rheumatology face the difficult task of evaluating which measurements are necessary and whether or not obtained results are sufficient to present a diagnosis. Objectivity is difficult under such circumstances, and the process as a whole (estimating which measurements are necessary, interpreting significance from a wide range of test results, etc.) is extremely time-consuming. These reasons, among others, were the motives behind developing a classifier for the diagnosis of autoimmune diseases.

A target vector was obtained in the form of diagnostic estimates belonging to each of the 1750 clinical cases. Experts in the field of rheumatology provided these inputs to our model, by assessing the status of each case. Their estimates were originally presented as the probability each patient had of suffering from a specific autoimmune disease; "none", "very low", "low", "medium" and "high" probability of disease [28]. Classification criteria were established from these probabilistic estimates, and a set of classes defined with the estimates comprising the class labels. To represent the estimates as numerical labels they were mapped to the values 0, 1, 2, 3, or 4, corresponding to increasing likelihood of disease. Therefore, five distinct classes were defined for our classifier, with labels ranging from 0 to 4.

3.1.2 Challenges with the Data

A challenging characteristic of our data, and a core issue in medical data mining, was the large number of input variables or high dimensionality that the raw data sets presented [44]. Dimensionality reduction is normally considered a necessary part of preprocessing raw, medical data. The reduction of dimensions of the data sets, or feature reduction, can be achieved through feature filtering [44]. This involves selecting those variables of the input space that contribute the most to the final output of the model. Statistical properties of the input data can be used as a means to identifying relevant variables, thus "filtering" out the variables most suitable for serving as a classification basis [44]. This is an approach that we utilized for the dimensionality reduction process of our classifiers. The autoimmune data sets originally presented around 20 features (parameters related to autoantibody measurements), but were reduced to the 5-10 features most relevant to the classification. Other properties than statistical information may be used to achieve dimensionality reduction. For our distinct sets of data, specific to each autoimmune disorder, different variables were of different importance to the final output of the classifiers. Variables, or autoantibody measurements, that did not present statistical importances but were still known to be specific to the disease in question, were included as features for those classifiers.

Another difficult aspect of modeling our data was attributed to how much of the values of the sets were missing, and missing due to different causes, i.e. the high structuredness of the underlying data sets (see Section 2.1). Much of the missing entries were meaningless, missing at random (MAR) data that were easily manageable for classification [23]. Other missing entries of the set represented missing results that could be traced back to specific blood tests being omitted. A large number of blood tests are omitted in immunoassays due to high cost, complex recovery of results, etc. These omitted measurements still possess some meaning, i.e. they are not MAR data but defined as missing not at random (MNAR). The MNAR values were found among all autoantibodies, since nearly all blood sample analyses involve some tests being knowingly omitted. These values were handled differently for each method that was implemented in the developmental process of our classifier.

3.2 Processing the Blood Test Data

All immunological blood test that were obtained for the purpose of developing this classifier were subjected to extensive interpretation and preprocessing. Raw serology results have a range of manifestations, from being presented as a mere positive-negative result to being a specific value from a continuous scale of values [4]. All available results were converted to numerical form according to a standard that was defined for the process. Results above a certain numerical threshold, were assigned the value of that threshold, other results that were presented as simply a positive or negative result were given a binary form of 1 or 0, respectively, and so on. As an addition to the variables that comprised our training set, a set of rules was supplied with the data. These rules presented a comprehensive overview of reasons for missing entries within the sets, therefore defining the missing entries as either MAR data or MNAR values. For example, a missing measurement of rheumatoid factor (RF) is always handled as an MNAR entry, since RF is a discriminative factor for almost all autoimmune disorders and even when measured as negative, that result can still contribute to a diagnosis by eliminating a variety of disorders [4]. The only variables that can be deemed MAR are omitted measurements of some factors, or autoantibodies, that are directly related to the manifestation of other factors in the bloodstream. And so if the dominant factors have been measured as negative and measurements for the dependent factors were omitted they can be deemed MAR values [4, 23].

The actual development of our diagnostic model began by an intricate study of the data. This meant learning the meaning of each type of entry and searching for obvious characteristics or patterns within the data frame. The majority of the entries of the data sets were direct measurements of different autoantibodies detected in the bloodstream, others were some variations of the direct measurements. The set of rules that had been supplied with the data, describing the meaning of omitted variables along with the importance of each factor of the set, were applied to the set to distinguish the MAR variables from the MNAR entries. This constituted an extremely important step in the data processing stage.

In order to estimate which factors of the data sets had a significant statistical effect on the stage of the diseases, regression analysis was performed on each of the factors with regards to the target vector of the data set. Considering two parameters specifically, the r^2 and β values, regression analysis was run for each factor of the set [24]. The β coefficient represented the degree of linearity, i.e. the amount of increase of the target value induced by each factor, and r^2 provided information regarding the correlations of the variables of the set [24]. The results of the statistical methods were studied, resulting in an overview of all relevant factors for our model structure. For RA, these were the variables that are listed in Table 3.1.

Table 3.1: Factors relevant to the structure of an RA diagnostic model.

Cyclic citrullinated peptide (CCP)	
Antinuclear antibodies (ANA)	
Rheumatoid factor (RF)	RF-ELISA
	RF-IgM
	RF-IgG
	RF-IgA

A knowledge-based estimation of the results of the most determinative factors was used in this manner, to determine which autoantibodies would be used as relevant features for the classifier of each autoimmune disease. However, some of the factors of the data sets were altered in order to observe linearity under other circumstances or to check for underlying linear behavior. Such alterations mainly involved logarithmic mappings of the factors; if these indicated advantageous statistical results, the data sets were subsequently augmented to contain the mapped factors.

3.3 The Backbone of the Model

A graphical model was considered an ideal choice for the backbone of our model, aiming for maximum transparency of the decision-making process of the classifier.

3.3.1 Bayesian Network Implementation

The Bayesian network modeling of our data was developed as a linked net of factors that were considered to have a fundamental impact on the likelihood of whether or not a patient was suffering from a specific disease. Our approach involved the calculated probabilities being based on the knowledge that the determining factor's parent nodes were either positive or negative. This type of Bayesian network assumes a binary form of the variables, i.e. they are either positive or negative, and this form of Bayesian networks is what we chose to focus on in the development of our autoimmune classifier. Figure 3.1 shows a simplified Bayesian network, similar to the one that was set up for our RA data.

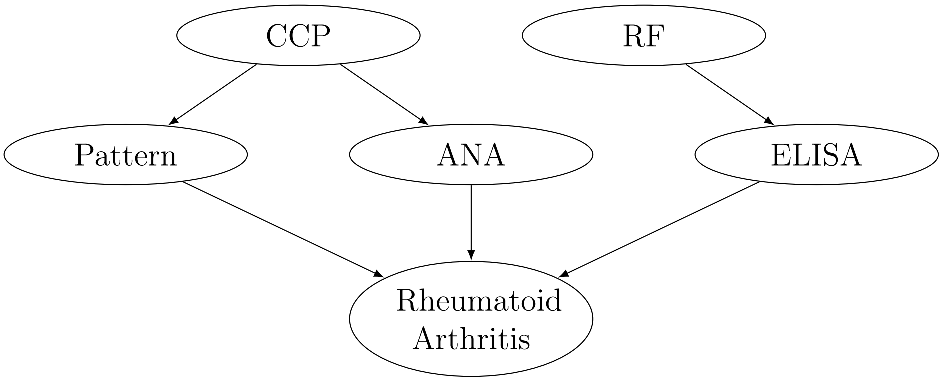


Figure 3.1: Basic elements of a Bayesian network.

The factors CCP and RF, which are factors extremely specific to the diagnosis of RA, have been defined as the roots of the network, and other measurements, such as those of antinuclear antibodies (ANA), an ELISA measurement and pattern presentations of certain autoantibodies are represented as factors conditional on CCP and RF. Basic calculations of the nodal probabilities were obtained from examining the network and calculating the joint probabilities of the data that comprises the network:

$$\begin{aligned} p(Data) = p(CCP)p(ANA|CCP)p(Pat|CCP)... \\ ...p(RF)p(ELISA|RF) \end{aligned} \tag{3.1}$$

Thus, traversing the nodes of the network, and performing a point-wise product of the conditional probabilities rendered the final probability of a new data point fitting to a given class. The conditional probabilities of the factors in each node were calculated as the mean of the previously known probabilistic estimates ("none" $\simeq$ 0% probability of disease to "high" $\simeq$ 95% probability), which could be attributed to that factor being either positive or negative in the underlying data set.

3.3.2 Decision Tree Implementation

Having decided upon a graphical model as the best fit for the backbone of the model, an approach through Decision tree learners had seemed like an ideal approach to our data from the very beginning. Preliminary results of the Bayesian network method, presented in Chapter 4, indicated insufficient accuracies for the performance of individual classifiers based on this method. The focus was therefore turned towards a tree-based model.

Establishing the basic structure of a Decision tree involved identifying those variables that provided the greatest separation of the input space (the training data) into decision regions; thus creating smaller subsets of data in these regions [15]. A couple of built-in packages were tested in the predevelopment of the Decision tree. The purpose of testing various approaches of tree models was first and foremost to employ the learning involved with these models to establish those variables that would provide the best separation of the data into prediction regions, i.e. leaves of the tree. There are various implementations of classification trees available in the R programming language. Among these are two CART algorithms, the *tree* algorithm and the *rpart* algorithm. The *rpart* is a recursive partitioning and regression algorithm, while *tree* is a more basic classification tree algorithm [46, 47]. Both algorithms are based on a tree-growing method that consists of three phases [17, 31]:

1. **Start** with a single node containing all points.
2. **Stop** if all the points in the node can be assumed to belong to the same class.
3. **Otherwise**, search over all binary splits of all variables for the one that will reduce the input set as much as possible. If one of the resulting nodes contains less than some predefined number of points, stop. Otherwise, take the split and create two new nodes.
4. **Go back** to step 1 for each new node.

After having used the built-in CART algorithms as a tool to establishing an ideal structure for our tree, the actual implementation of our tree-based model was begun. Our tree was not a strictly learned tree, in the sense that we previously defined the separator variables of the branching sites. These separators were obtained from both the learned approach of the built-in CART algorithms, but were in some cases those variables that had resulted in highest statistical importance for a specific autoimmune disease. The tree structure was implemented as a series of if-else statements, with each branching taking place at such a statement.

The CART algorithms employed linear regression in the leaves of their trees, resulting in a final output value. In those cases where the branching of our trees had lead to a determined section of variables, which all were known to indicate either a high affinity of disease or very low affinity, the numerical values according to the diagnosis induced by those variables (e.g. 4 for a "high" probability of disease or 3 for a "medium" probability) were hard-coded into the model. Thus for extreme, high-probability cases, i.e. where all variables strongly indicated a specific diagnosis, the results were returned as hard-coded values. In those leaves where the division of the input region had not lead to as distinct a result, the final prediction was obtained by a manner of model hybridization, i.e. by adding other classification methods to the classification tree. The act of combining multiple models into one classifier, or hybrid modeling, has often been found to improve the performance of a classifier [48, 49]. Therefore, implementations of a suitable method for classification that involved this sort of hybrid modeling, defined the following steps in the developmental process of our classifier [15].

3.3.3 Classification in Sub-regions of the Decision Tree

In order to establish which methods to use in the hybrid model, two methods that have shown promise in combining models for classification were implemented for our data; the classical approaches of Discriminant function analysis (DA) and k-Nearest neighbors (kNN).

The implementation of a classifier based on discriminant functions involved a comparison between two methods of DA; a linear and a quadratic discriminant function. The discriminant function was, in both cases, calculated from the covariance matrix/-matrices and mean values of the factor in question. In the linear (LDA) approach, a single covariance matrix was constructed for the data set as a whole, before any class labels were established. The set was then divided into five classes, according to the blood test results of each patient. The first class was thus defined as patients with zero probability of having the disease, the second defined as those with a very low likelihood of the disease and so on. With the implementation of a quadratic discriminant (QDA), a specific covariance matrix was constructed for each class. These independent covariance matrices tended to yield a zero determinant, due to induced incalculability (logarithm of a zero determinant). For the cases in which this occurred, a method of switching into a linear model was carried out, thereby constituting a mixture model of sorts.

The implementation of k-Nearest neighbors for the leaves of our Decision tree involved some specific approaches to calculating the distances of the neighbors. The fact that so many crucial factors of the training set were missing, MNAR values, proved challenging for when distances to these factors were to be calculated. This was especially true since we wanted to include the information stored in the MNAR values for the final prediction. In the MNAR cases, the EM algorithm was used to estimate an imputation value for these missing entries. Distances to the imputed values were subsequently calculated, thus obtaining those k neighbors closest to our new data point. An additional challenge involved establishing the optimum number of k for adequate modeling of the data. The nearest neighbors taken into account in our model were originally defined as five. Five is a standard number of neighbors for k-Nearest neighbors applications, although in some cases five points do not address the problem with adequate precision. The value for k was tried from 2 to 20 points for our data, but none of those tests indicated a better result than for keeping k as five. In fact, since we took the approach of inverse-distance-weighting the neighbors that were found for a new data point, the value we chose for k didn't account for as large a part of the model performance as it would have otherwise, since this approach involves weighting the neibhors' distances after a k number of neighbors have been found [37].

3.3.4 Performance Assessment

In order to obtain a rough estimate for how well each implemented method was performing, LOO cross-validations were carried out constantly. These provided a percentage that described the amount of cases being classified correctly, but did not give any other indications of the reliability of the classification. However, calculating the mean squared error (MSE) of the results vectors of each implementation provided the information necessary to be able to assess the reliability of the different methods. Furthermore, the MSE is directly related to the amount of outliers[4] presented for a LOO run of a model. Outliers presented by the various models were thus detected both

[4] Outliers are output values that deviate considerably from expected target values [15].

by visually estimating the more extreme outliers from graphical presentations of the results, and by calculating the MSE of the output values.

In the final stages of development of the classifier, the observed outliers were examined further. In some cases these outliers indicated severe cases of misclassification. A physician might e.g. have estimated a patient as having a "high" probability of autoimmune disease, but the model gave prediction results of a "low", or in some cases "very low" probability of disease. This problem was approached as an interactive process, set up between the author and experts in rheumatology. The outliers predicted by our model were presented to experts so as to verify that they were indeed either severe cases of misclassification, or due to an inconsistency in the estimates that had originally been provided with the data sets. Most of these outliers were in fact found to be due to the latter, an inconsistency in the diagnostic estimates of experts, and these same data points were subsequently updated for the training data sets. Other outliers were mainly minimal deviations from the proposed expert estimates, and were therefore left unaltered to avoid overfitting of the model.

3.3.5 Implementing Random Forests

After having established a final structure for the classifier and obtained adequate results, a new method was applied to the data sets. This was the method of Random forests, developed by Leo Breiman et al. A built-in package in the R programming interface exists for the Random forests method, implemented for R directly from the original Fortran code by Breiman [41]. This package was employed for the purpose of testing the method with minimal handling of the data. The variables that had been established as determinant factors for the outcome of our classifier in previous methods were used as separators for the forest setup. However, the method of Random forests uses different separator variables for each of its trees to provide a broader spectrum of possible data mining results. The model was therefore supplied with sufficient, possible separator variables so as to be able to choose from. A controlled approach to the size of the forest was taken at first, with the forest size set as 100 trees and the random assortment of separators limited to three separator variables for each tree. The approach of allowing for more freedom within the algorithm itself was tried in subsequent tests, where the algorithm decided upon an ideal number of trees and separators. Finally, the results from the Random forest approach were compared to the results previously obtained from our specifically designed classifier. A thorough discussion and comparison of the two can be found in the following chapter.

Chapter 4

Results

All methods that have been discussed in this thesis were implemented as diagnostic models, specifically adjusted to each of the ten autoimmune disorders. However, the results presented here are those that were obtained from implementations specific to the RA data set. The RA data consists of blood test results from 240 patients, including a target vector of 240 diagnostic estimates on the scale of 0 to 4. The cross-validation results are mainly presented as the percentage of how well model output values matched the target vector values. In the case of RA we had 240 such match or mismatch indications, although these were presented as regression results rather than a match-mismatch classification. The output of the model was therefore on a continuous scale, from 0 to 4, which provided more meaningful results than a classification otherwise would have, since it indicates how close to or how far from a known prediction the output actually is. It is known that a relatively low difference can be between diagnostic classes obtained from the possibly subjective estimates of physicians, where a "medium" likelihood of disease does not provide a distinct difference from either a "low" or "high" possibility of disease. In accordance to this, the results that we present have been given a defined margin of error to indicate a correct or false prediction value. The margin of error corresponds to one class higher or lower than the actual outputted value.

Some implementations have been presented graphically as the linear relationship between the target values, which are in fact the diagnostic estimates of experts in rheumatology, and the values outputted by the model. The diagonal of these graphical presentations represents a perfect linear fit. However, the points that fall within the margin of error of one class (indicated as a shaded area) are also defined as correctly classified points due to the difficult distinction between the diagnostic classes that were defined prior to building the classifier. The results are presented separately; Section 4.1 describes those results that were obtained through the LOO cross-validation and comprise the actual developmental process leading to our final classifier, concluding that the final version of the classifier shows a 100% success rate in classification. Section 4.2 presents a more accurate assessment of the performance of our model, with results for the classification of twelve new test cases showing that all of the cases were successfully classified.

4.1 Leave-One-Out Results

Results were produced and studied throughout the developmental process of our classifier. This enabled constant assessment of the performance of the model, allowing different approaches and implementations to be tested. The method of LOO cross-validation was utilized for this purpose, revealing preliminary results for each implementation of the methods that have been discussed in Chapter 3. We present the LOO results as a form of comparison to the target vector of the original data set, as a percentage or by graphical representation. The results should give a good indication of the effectiveness of each implementation, although a certain bias is always bound to accompany such validation approaches.

4.1.1 Bayesian Network

The outcome of the Bayesian network-based model was not as good as had been expected, the classification showed promising results from a probabilistic point of view only. However, when a complete prediction, based on both the prior probabilities and conditional probabilities of each Bayesian node, was estimated, the results were found to be poor.

Table 4.1: Bayesian network LOO results.

	Compliance to target values (%)	**Mean squared error**
Bayesian network	53.4	0.536

As Table 4.1 shows, first results from LOO cross-validation of the model indicated 53.4% correctly classified cases and presented a mean squared error of 0.536. This misclassification rate describes a highly unreliable classifier, as no conclusions can be drawn from a mere 50% probability of the classifier classifying new cases correctly.

The unfavorable results from our implementation of a Bayesian network were believed to originate from two possible causes. Firstly, the model, in a similar way as blood tests performed for suspected autoimmune patients do, bases its prediction on two roots (two main parameters). This can result in problematic conditional probability calculations. If either one, or both of these roots is not measured, it becomes extremely difficult for the Bayesian network to produce conditional probabilities for the rest of the nodes. Secondly, many of the variables taken into account in the model, are interrelated in a more complex way, with internal correlations that cannot be described in as simple a way as the Bayesian network intends to do. These complex relations are e.g. correlations between a child node and more than one grandparent node, or sibling nodes being related in various ways. As the amount of calculations required to obtain a conditional probability outcome for each node grows exponentially with the number of parents related to that particular node, increasing the network links was not attempted for our model [26].

4.1.2 Tree Implementation

The main limitation we faced when intending to use built-in R packages for the development a Decision tree, was the lack of control over specific actions available for missing data values. This proved especially limiting in this project, since the main goal was to be able to distinguish between different types of missing values (MAR versus MNAR) and have the classifier take action based on that distinction. For the purpose of a final classification, or prediction, the built-in packages were therefore only used to decide which factors would best serve as separator variables in the tree structure. We ran our RA data through two available CART packages to obtain some useful information about an adequate tree structure, such as the ideal depth of a Decision tree for the data (obtained from automatic pruning) and the variables that provided the best separation of the training data into decision regions. The packages that were tried were the *rpart* and *tree* packages, which have been discussed in Section 3.3.2. These obtain their outputs from linear regression of the target values associated with the parameters in each decision region (leaves of the trees), as can be seen in Figures 4.1 and 4.2.

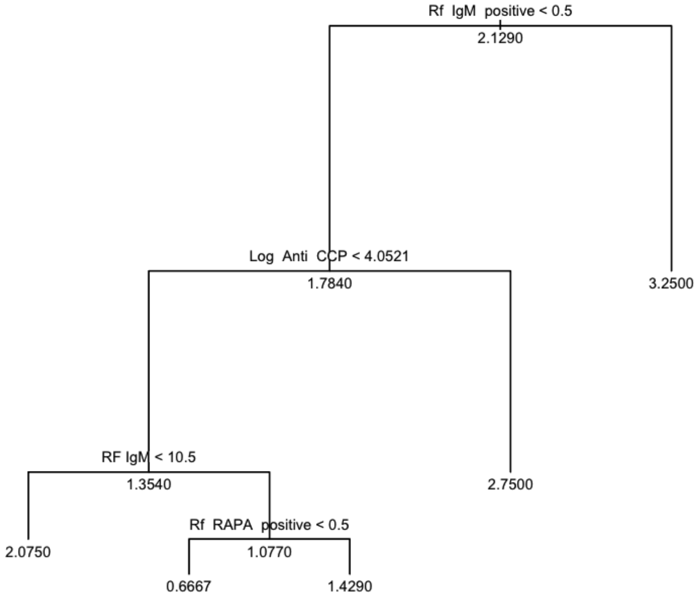


Figure 4.1: The tree structure obtained from the "tree" package in R, output values associated with each sub-region of the tree are presented as linear regression values of the target vector [47].

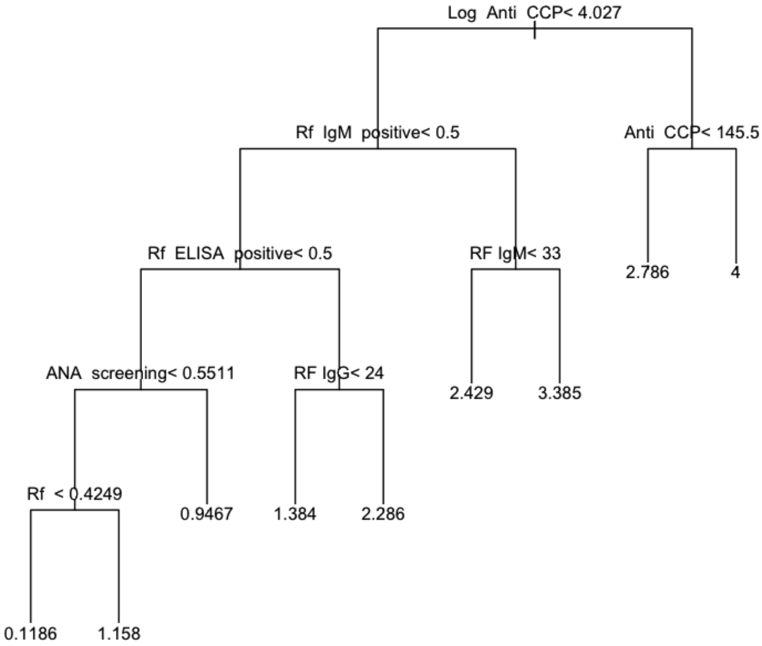


Figure 4.2: The tree structure obtained from the "rpart" package in R [46].

A decision was made to handle the MNAR values by imputation through the EM algorithm. LOO results that were obtained from both of the built-in packages are shown in Table 4.2.

Table 4.2: Built-in CART results, MNAR's exchanged for EM means.

CART algorithms	**Compliance to target values (%)**
tree	93.43
rpart	95.05

Even though the *rpart* package resulted in a higher percentage of correctly classified cases, both trees were regarded as relevant in the ongoing development of the classifier. The results were found to be especially useful in respect to the structure of the tree. Both algorithms involved automatic tree pruning, and the structures seen in Figures 4.1 and 4.2 were obtained for the optimum amount of pruning. When developing the tree-structure of our classifier, the structures that were obtained from the *rpart* and *tree* packages were therefore used as a basis. A mean squared error was not estimated for the CART models since they were not intended for a final comparison of classification methods.

The backbone for our classifier had now been established. The separator variables obtained from the CART packages, as well as the variables that presented a statistically relevant effect to the final diagnosis of a patient (see Table 3.1) were implemented through a series of if-else statements as various branching sites of the tree. To obtain final prediction values from these branching sites, several values were hard-coded into the model in a similar way as the values returned by the CART packages, shown in Figures 4.1 and 4.2. This was only considered appropriate for cases where measured autoantibodies showed a very strong indication of a specific probability of disease, e.g. as can be seen from Figure 4.2 where the factor CCP, which is a factor that has shown high specificity for RA patients, is measured as higher than 145.5, that case is almost surely a high-probability RA case and the according prediction value of 4 can be hard-coded into the model. In other cases, the results from the blood tests are not so definite. The following sections describe our attempt to intertwine other classification methods with the tree structure to hopefully obtain a final classifier with the best performance possible for our autoimmune disease data.

4.1.3 DA Tree-based Model

As has been discussed, the act of combining other methods with the basic tree structure was considered an ideal approach to modeling our incomplete sets of autoimmune data. First implementations for this purpose involved a Discriminant function analysis (DA) in the leaves of the Decision tree, an approach that was intended as a more accurate classification approach for the sub-regions into which the tree had partitioned the input data. However, the first runs of an LOO cross-validation for the DA-tree hybrid classifier resulted in worse classification percentages than the built-in CART packages had previously shown. This can be seen in Table 4.3.

Table 4.3: LOO results from DA-based tree.

	Compliance to target values (%)	**Mean squared error**
Tree-based DA	68.20	0.410

Comparing the result of 68.2% correctly classified points to the results from the *rpart* package, which amounted to 95.05% of correctly classified points, the DA approach was determined to be ineffective for the desired classification task of modeling our autoimmune data.

These inadequate results are believed to be due to the fact that autoimmune data are comprised of parameters that usually increase in value according to class rank. Thus, values coherent to the highest class can be expected to give the best results for the classification of a point in many instances, since that discriminant equation would produce a maximum value [6]. Therefore, an excessive amount of cases will be classified as high-probability cases, resulting in a poor diagnostic model.

4.1.4 kNN Tree-based Model

The kNN model for classification was the subsequent method to be implemented with the structural backbone of a Decision tree. Most classical approaches to classification through kNN use the standard *Euclidean* distance for finding the nearest neighbors. Our first implementations therefore made use of the *Euclidean* metric for distance calculations. So as to involve the information from the MNAR values of our set, these missing values were temporarily filled in for by EM estimations of the corresponding factor. The distances were furthermore inverse-distance-weighted, thus emphasising the contribution of the closest neighbors even further for the final output value of the model. Having implemented this approach, the first results for *Euclidean*-based kNN were obtained, and are shown in Table 4.4 and Figure 4.3.

Table 4.4: First LOO results from the classification tree with Euclidean-based calculations of kNN.

	Compliance to target values (%)	**Mean squared error**
Tree-based Euclidean kNN	96.65	0.307

These first results of using kNN for classification in the leaves of the tree were extremely promising, as can be seen in Table 4.4, where 96.65% of the cases have been successfully predicted. However, in the case of highly correlated data the *Euclidean* distance can produce biased results, as has been discussed in Chapter 3. The approach of basing the distance calculations on the *Mahalanobis* distance was therefore tried as well, and those results will be presented in the following text.

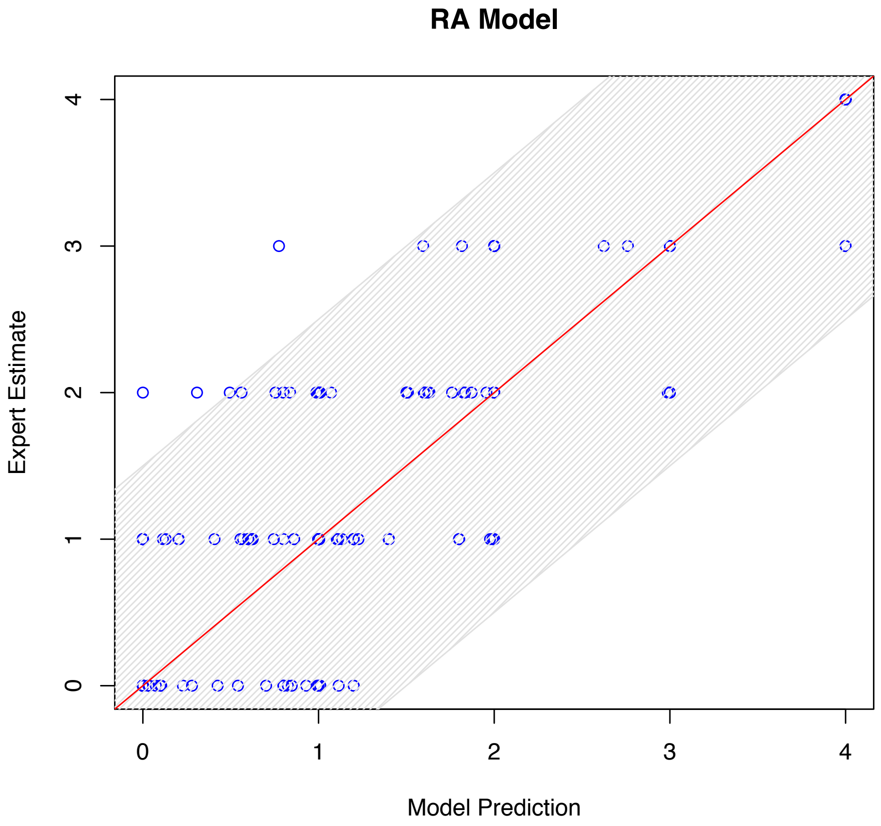


Figure 4.3: First results of the tree-based Euclidean kNN prediction for RA.

Figure 4.3 shows the expert estimates of each RA case, ranging from "none" to "high" probability (0 to 4) of disease. These are plotted against the outputs of our tree-based *Euclidean* kNN model to examine linearity between the two, or the compliance of the model outputs to the target values (expert estimates). As can be seen in the figure, nearly all cases fall within the predefined margin of error (shaded area) and are therefore correctly classified. Those points that fall outside of the shaded area are regarded as misclassified, and will be discussed in greater detail in the following sections.

The *Mahalanobis* calculations required a covariance matrix to be obtained of the factors of the set, as has been described in Section 2.4.1. Calculating the covariance matrix involved some complications. These were mainly related to the high missingness of data in the sets, where all missing entries (MAR's and MNAR's) were initially omitted for calculations of the covariance matrix. This lead to the matrix becoming biased, the covariance entries of the matrix were not always calculated for the same two numbers. Therefore, the covariance matrices that were obtained, frequently became singular, or had negative eigenvalues. In these cases the matrices had to be forced into a positive-definite[5] state, in order to be applicable in the *Mahalanobis* distance calculations. Forcing the covariance matrix into a positive-definite state was believed likely to lead to poor final results, and was furthermore demanding. Other options than simply omitting all missing values were therefore considered. The first option involved simply fixing distances to the MNAR values as one standard deviation from the mean value presented by the measured instances of the factor in question. A second option involved applying the EM algorithm to the MNAR values, before calculating the covariance matrix. The EM estimates for the MNAR values were thus used as imputation values for the MNAR's to estimate the distance to these values, instead of simply ignoring the MNAR's in the nearest neighbor estimation.

[5] A matrix is positive-definite if its eigenvalues are non-negative [24].

The results for the different variations of the *Mahalanobis*-based kNN are presented in Table 4.5.

Table 4.5: Mahalanobis-based kNN results from LOO.

	Compliance to target values (%)	**Mean squared error**
MNAR entries omitted	92.89	0.544
MNAR value distances set as one std.dev. from factor values	95.82	0.320
EM algorithm used to obtain means for MNAR values	95.82	0.315

None of the *Mahalanobis*-based approaches resulted in better performances than the *Euclidean*-based kNN approach. This, combined with the fact that the covariance matrix calculations were computationally demanding, lead to the decision of relying solely on the approach of *Euclidean*-based kNN.

4.1.5 Model Refinements

Initial versions of the classifier resulted in some extreme outliers from the expert estimates that had been associated with these cases. These misclassified points were regarded as severe, since such results can distort the overall performance of the model. Figure 4.4 shows the misclassified points of patients who are in fact medium risk RA patients, being classified as very low risk patients.

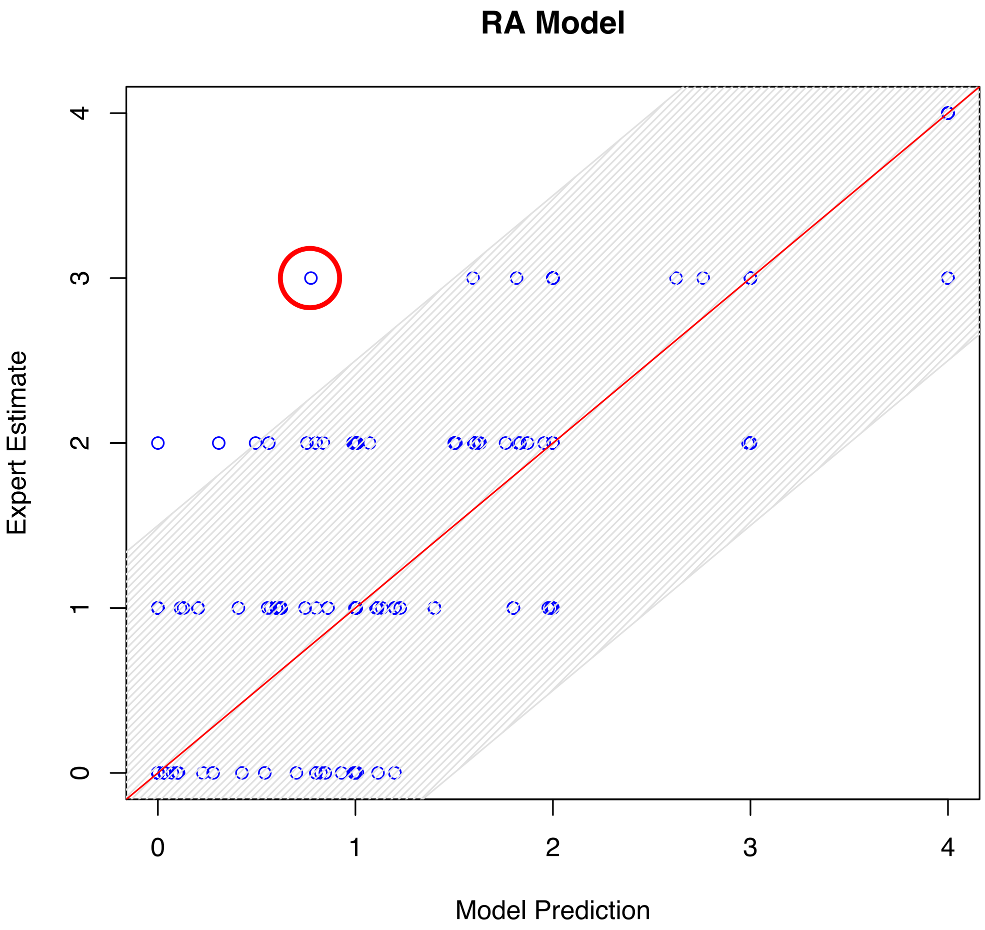


Figure 4.4: Outliers presented in the development of the RA model.

Furthermore, a substantial number of data points are classified outside of the previously defined margin of error of one class (gray area). In order to try to rectify these misclassified instances, the outliers were examined with regard to their expected target value. Some of the outliers could be attributed to a false estimate of the patients' status, based on their blood test results. These were mainly statuses ranging from a "very low" probability of disease to a "medium" probability. The outliers that could be traced to faulty target values, obtained for the data set, were subsequently corrected after having been submitted to experts in rheumatology.

Table 4.6 shows the improvement in performance accuracy, achieved from outlier rectifications.

Table 4.6: Euclidean-based kNN Results from LOO, before and after outlier corrections.

	Compliance to target values (%)	**Mean squared error**
Before outlier improvements	96.65	0.307
After improvements	100.00	0.297

The extremely good results of a 100% of correctly classified cases (with a margin of error of one class) were obtained for when the outliers had been adjusted; an improvement of nearly 3.5%. Figure 4.5 shows the results of the final tree-based kNN model for RA, with adjusted outliers.

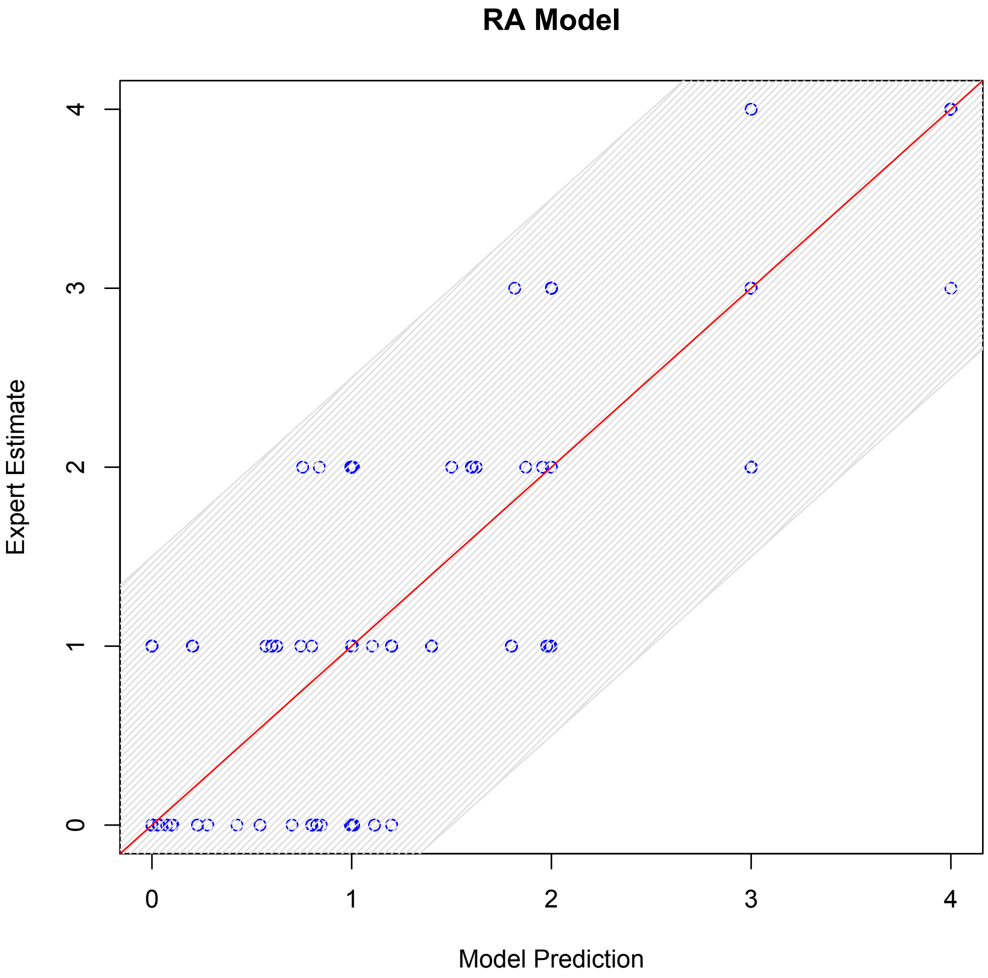


Figure 4.5: The final RA model results.

4.1.6 Random Forests

As a final approach to developing a classifier suitable for the task of classifying autoimmune data, the relatively new method of Random forests was implemented for our set. For this purpose, use was made of a built-in package in R, which is described as the R interface for the original Fortran code that was developed by Breiman et al. [41]. LOO results from this implementation are shown in Table 4.7.

Table 4.7: Random forests LOO results.

	Compliance to target values (%)	**Mean squared error**
Random forests approach	97.91	0.326

The performance can furthermore be seen in Figure 4.6.

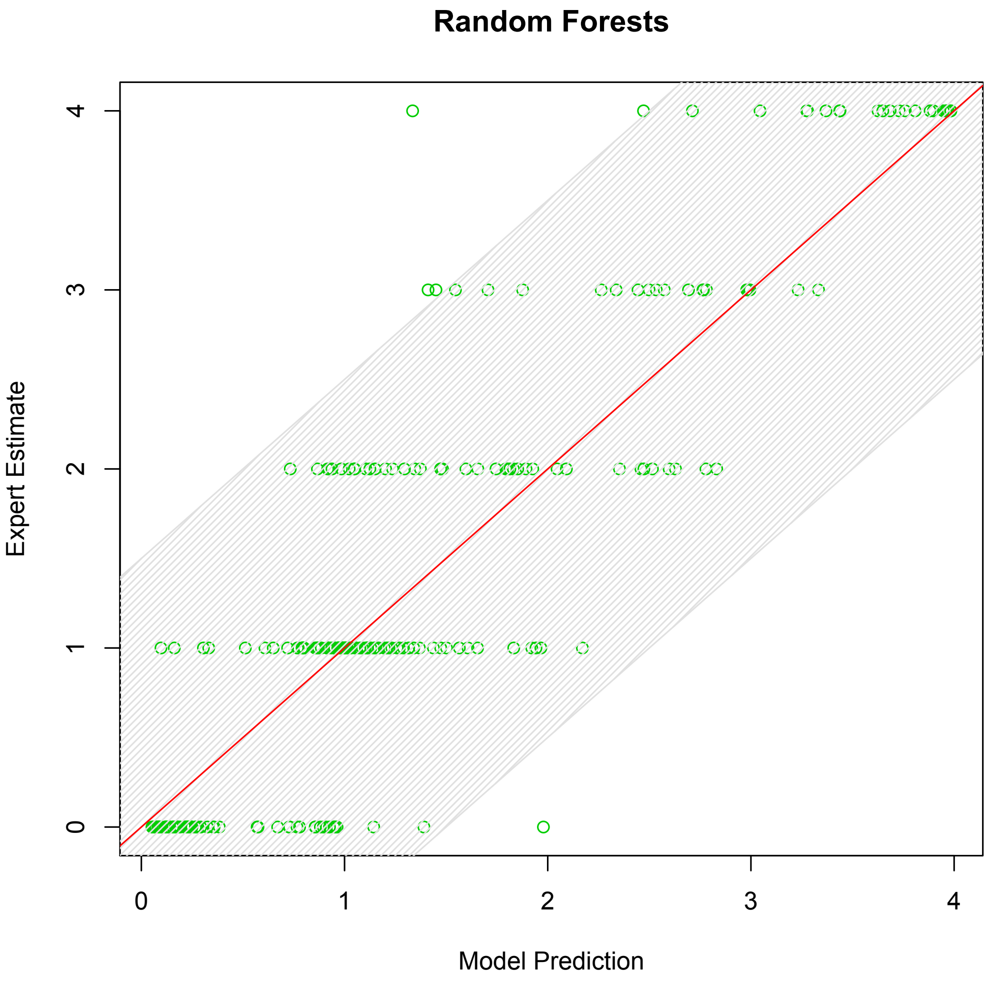


Figure 4.6: Random forests model results.

The approach of Random forests introduced some intriguing results with the LOO run showing that 97.91% of cases were being predicted correctly. However, some extreme outliers were presented as well, as can be seen from Figure 4.6, and so the model would require some significant reforms before being presented as an actual diagnostic model. Such reforms can prove difficult for an approach through built-in, already established packages. The Random forests method itself is extremely complex and was considered overly extensive for a hands on implementation of each tree to be attempted within the scope of this thesis.

4.2 Results from New Test Cases

Very recent additional data has been obtained for rheumatoid arthritis. This data, intended as test cases for the model, enabled the unbiased validation of our model. This was considered an extremely important aspect in verifying that the model had not been overly trained, or overfitted to our data. The results from the run of these new test cases through our model are presented in Table 4.8 as a "yes" or "no" to being classified as RA cases, with the prediction value presented as well (usually ranging from 0 to 4).

Table 4.8: Results from the classification of new RA test cases.

Case number	**Positive for RA**	**Prediction value**
1	*yes*	4.00
2	*yes*	4.00
3	*yes*	4.00
4	*yes*	4.00
5	*yes*	3.00
6	*yes*	4.00
7	*yes*	3.00
8	*yes*	4.00
9	*yes*	4.00
10	*yes*	4.00
11	*yes*	4.00
12	*no*	3.66

Table 4.8 shows that only one case, out of the twelve presented, was incorrectly classifier. This indicates that the overall performance of our model for the task of classifying new patients is good. A more detailed overview of the results for each case, is provided in Table 4.9.

Table 4.9: Results from each, individual classifier for the 12 RA cases.

	Cases and corresponding prediction values											
Disease	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
RA	***4.00***	***4.00***	***4.00***	***4.00***	***3.00***	***4.00***	***3.00***	***4.00***	***4.00***	***4.00***	***4.00***	***3.00***
Sjögren's	0.00	2.00	0.59	2.20	0.59	1.01	0.40	1.84	2.01	1.44	1.92	***3.66***
SLE	2.24	1.89	0.60	0.76	0.60	2.38	0.60	0.97	1.20	1.20	2.00	2.61
MCTD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Anti-SX	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Ch-Strauss	1.80	1.80	0.40	1.80	0.40	1.40	2.00	0.60	0.40	0.40	1.40	1.40
Micro-Poly	1.80	1.80	0.40	1.80	0.40	1.40	2.00	0.60	0.40	0.40	1.40	1.40
PM-DM	1.00	1.60	0.40	0.75	0.40	1.00	0.40	0.97	0.63	1.20	***2.60***	2.25
Systemic	2.17	***3.00***	0.40	0.77	0.40	2.00	0.40	0.96	1.20	1.20	2.20	1.31
Wegener's	1.60	1.60	1.00	1.60	1.00	0.00	1.00	2.20	0.00	0.20	***3.40***	***3.00***

Table 4.9 shows individual prediction results from all the classification models we have developed, for running the established RA cases that we obtained as test data through the models. The prediction values that may indicate the presence of disease are accented in the table. It is evident that in some instances, more than one autoimmune disease is possible, and even probable. This is, in fact, a known complication in the diagnosis of autoimmune disorders, many of the disorders overlap significantly in autoimmune patients. The twelfth case of the test data we obtained was classified as a Sjögren's case, with the prediction value of 3.66 for Sjögren's syndrome. However, the case also scored the prediction value of 3.00 for RA and Wegener's Granulomatosis, and may therefore be a case of clinical overlapping of these three disorders. The final version of a diagnostic model will integrate the output probabilities from all ten autoimmune disorders, and as such introduce diagnoses for overlap disorders as well.

4.3 Summary

This chapter has presented results from all stages of the development of our final, hybrid classifier. Most results were obtained for LOO runs, constituting the ongoing test results from the developmental process. The results that were obtained from new test cases are however just as important, if not of greater importance, to the final assessment of the model's performance. These results were encouraging, as they indicated that all test cases were correctly classified, if we assume that the possibility of overlapping disorders exists. These test runs furthermore showed that no severe overfitting of the model had occurred.

After having presented results from implementations of various methods that were considered likely candidates for the classification of autoimmune data, we can truly say that our hybrid model of a Decision tree and inverse-distance-weighted kNN surpassed all other candidates. However, results from the method of Random forests did appear to have the possibility of presenting a comparable performance, with a mere 2.09% difference in success rates. The final chapters follow up on the results that have been presented here with a more detailed discussion and conclusions of our findings.

Chapter 5

Discussion

The classifier that we have presented shows an extremely encouraging performance for its task of correctly predicting the presence of autoimmune disease. Several methods were implemented and cross-validated, including the method of Bayesian networks, Discriminant analysis and Random forests, but the best results and performance were obtained for our final classifier of a Decision tree combined with an inverse-distance-weighted kNN model.

Decision trees presented an obvious choice for the underlying model structure, with the other graphical model candidate, the Bayesian network, not being able to provide a reliable classification when implemented for our data. However, the method of Random forests seemed to produce results compatible to those of our Decision tree classifier. Nonetheless, the transparency of a single classification tree will always provide an additional advantage over the forests model for applications in clinical diagnostics. The decision of developing the tree structure into a hybrid model, rather than using a stand-alone tree, arose from the fact that a Decision tree by itself, using linear regression in its leaves, could not provide reliable outputs for non-extreme cases. A tree by itself could furthermore not utilize the information stored in the meaningful missing (MNAR) variables to its advantage. In fact, one of the greatest challenges we faced in developing our classifier was implementing the methods to be applied in sub-regions of the classification tree so that they would model the MNAR variables adequately. The approach of basing kNN distances on MNAR values, through temporary imputation of EM-means, and re-weighting those distances by an inverse distance, proved successful for including the MNAR information in the prediction. The method of Discriminant function analysis did not compare to the classification by kNN, and furthermore presented a relatively high performance error.

A classifier this specific to the characteristics of autoimmune disease blood test data has, to our knowledge, not been presented before. The methods we have described and implemented for the purpose of finding the ideal classifier for this type of data are all well-established methods in the field of artificial intelligence in medicine. However, the data gathered from autoimmune disease blood tests are unique in their aspects of MNAR values. We believe that our hybrid classifier is the best approach to handling this type of data, a belief supported by the favourable classification results that have been presented in this thesis.

Classifiers for supervised learning approaches are highly dependant on the quality and amount of underlying training data. Therefore, it can be assumed that the classifiers we have developed for the more prevalent autoimmune disorders will be more reliable in their predictions than those containing a smaller amount of available blood test data. The classification models that are presented as individual classifiers in this thesis have, in fact, been integrated into a final model as part of a clinical solution system. The prototype of this system, shown in Appendix C, indicates that integrating the individual classifiers into one final model has proved beneficial, even though some of the individual classifiers can be considered more reliable than others.

Chapter 6

Conclusions

Our implementation of a Decision tree and inverse-distance-weighted kNN into a single, hybrid classifier has shown to successfully classify all available test cases of RA correctly, as well as providing the possibility of diagnosing overlapping autoimmune disorders. LOO cross-validation results for RA indicated a 100% success rate of classification, and LOO results for the other nine diseases (see Appendix A for further details) all show over 90% correctly classified cases. The method that came closest to achieving similar success as our hybrid classifier was the method of Random forests, resulting in 97.91% of RA cases being correctly classified. However, our hybrid classifier still has an advantage of 2.09% in performance over the Random forests approach, and the additional advantage of a more transparent structure.

A final software that makes use of our classifier is already available as part of a web-based solution system. The system is currently being made available to healthcare professionals; two versions of a demo-interface for the system are presented in Appendix C. Further work on the classifiers presented here will involve expanding the models so as to cover an increased number of autoimmune disorders, such as seronegative arthritis and fibromyalgia. Even though the scope of this thesis has been limited to the field of autoimmune diseases, the methodology we have described here can easily be adjusted to other classification problems. Hopefully, it will prove useful for data mining approaches of similar challenges to those we have faced in our classification of autoimmune data.

Bibliography

[1] K. A. Aziz and A. A. Faizal, “The role of the clinical immunology laboratory in the diagnosis and monitoring of connective tissue diseases,” *Saudi Med. J.*, vol. 25, no. 12, pp. 1796–1807, 2004.

[2] A. Davidson and B. Diamond, “Advances in immunology,” *N. Engl. J. Med.*, vol. 345, pp. 340–350, Aug. 2001.

[3] W. Taylor, D. Gladman, P. Helliwell, A. Marchesoni, P. Mease, and H. Mielants, “Classification criteria for psoriatic arthritis: Development of new criteria from a large international study,” *Arthritis Rheum.*, vol. 54, pp. 2665–2673, Aug. 2006.

[4] V. Nell, K. Machold, T. Stamm, G. Eberl, H. Heinzl, M. Uffmann, J. Smolen, and G. Steiner, “Autoantibody profiling as early diagnostic and prognostic tool for rheumatoid arthritis,” *Ann. Rheum. Dis.*, vol. 64, pp. 1731–1736, Dec. 2005.

[5] A. Davidson and B. Diamond, “Immunologic basis of autoimmunity,” in *The Autoimmune Diseases* (I. R. Mackay and N. R. Rose, eds.), pp. 19–32, San Diego, CA: Elsevier Science, 5th ed., 2014.

[6] K. J. Cios and G. William Moore, “Uniqueness of medical data mining,” *Artif. Intell. Med.*, vol. 26, pp. 1–24, Sept. 2002.

[7] R Core Team, *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2014.

[8] I. Kononenko, “Machine learning for medical diagnosis: History, state of the art and perspective,” *Artif. Intell. Med.*, vol. 23, pp. 89–109, Aug. 2001.

[9] P. Groves, B. Kayyali, D. Knott, and S. van Kuiken, “The ’big data’ revolution in healthcare: Accelerating value and innovation,” tech. rep., Center for US Health System Reform, Jan. 2013.

[10] P.-N. Tan, *Introduction to Data Mining.* Boston: Pearson Addison Wesley, 1st ed., 2006.

[11] A. Ramesh, C. Kambhampati, J. Monson, and P. Drew, "Artificial intelligence in medicine," *Ann. R. Coll. Surg. Engl.*, vol. 86, pp. 334–338, Sept. 2004.

[12] R. M. Wachter, "Why diagnostic errors don't get any respect - and what can be done about them," *Health Aff.*, vol. 29, pp. 1605–1610, Sept. 2010.

[13] M. Cherkassky, "Application of machine learning methods to medical diagnosis," *Chance*, vol. 22, no. 1, 2009.

[14] S. Narain, H. Richards, and M. Satoh, "Diagnostic accuracy for lupus and other systemic autoimmune diseases in the community setting," *Arch. Intern. Med.*, vol. 164, pp. 2435–2441, Dec. 2004.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning.* New York: Springer, 2006.

[16] E. Eirola, G. Doquire, M. Verleysen, and A. Lendasse, "Distance estimation in numerical data sets with missing values," *Inf. Sci*, vol. 240, pp. 115–128, Aug. 2013.

[17] N. Lavrač, "Machine learning for data mining in medicine," in *Artificial Intelligence in Medicine* (W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, and J. Wyatt, eds.), pp. 47–62, Springer Berlin Heidelberg, Jan. 1999.

[18] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, pp. 131–163, Nov. 1997.

[19] J. Soni, U. Ansari, D. Sharma, and S. Soni, "Predictive data mining for medical diagnosis: An overview of heart disease prediction," *Int. J. Comput. Appl.*, vol. 17, Mar. 2011.

[20] S. U. Amin, K. Agarwal, and R. Beg, "Data mining in clinical decision support systems for diagnosis, prediction and treatment of heart disease," *Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET)*, vol. 2, pp. 1–218, Jan. 2013.

[21] T. J. Cleophas and A. H. Zwinderman, "Discriminant analysis for making a diagnosis from multiple outcomes (45 patients)," in *Machine Learning in Medicine - Cookbook*, pp. 57–61, The Netherlands: Springer International Publishing, Jan. 2014.

[22] M. Saar-Tsechansky and F. Provost, "Handling missing values when applying classification models," *J. Mach. Learn. Res.*, vol. 8, p. 1623–1657, Dec. 2007.

[23] F. Cismondi, A. S. Fialho, S. M. Vieira, S. R. Reti, J. M. C. Sousa, and S. N. Finkelstein, "Missing data in medical databases: Impute, delete or classify?," *Artif. Intell. Med.*, vol. 58, pp. 63–72, May 2013.

[24] T. Hastie, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York, NY: Springer, 2nd ed., 2009.

[25] Y. Liu, H. H. Zhang, and Y. Wu, "Hard or soft classification? large-margin unified machines," *J. Am. Stat. Assoc.*, vol. 106, pp. 166–177, Mar. 2011.

[26] B. Das, "Generating conditional probabilities for bayesian networks: Easing the knowledge acquisition problem," *Comput. Res. Repos.*, vol. 15, 2004.

[27] P. Sebastiani, V. G. Nolan, C. T. Baldwin, M. M. Abad-Grau, L. Wang, A. H. Adewoye, L. C. McMahon, L. A. Farrer, J. G. Taylor, G. J. Kato, M. T. Gladwin, and M. H. Steinberg, "A network model to predict the risk of death in sickle cell disease," *Blood*, vol. 110, pp. 2727–2735, Oct. 2007.

[28] G. Yadav, Y. Kumar, and G. Sahoo, "Prediction of parkinson's disease using data mining methods: A comparative analysis of tree, statistical and support vector machine classifiers," in *2012 National Conference on Computing and Communication Systems (NCCCS)*, pp. 1–8, Nov. 2012.

[29] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, vol. 303, pp. 799–805, Feb. 2004.

[30] A. Singh and A. Moore, "Finding optimal bayesian networks by dynamic programming," tech. rep., Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, Jan. 2005.

[31] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and Regression Trees.* New York, N.Y.: Chapman & Hall, 1984.

[32] R. O. Duda, *Pattern Classification.* New York: Wiley, 2nd ed ed., 2001.

[33] J. d. A. Silva and E. R. Hruschka, "An experimental study on the use of nearest neighbor-based imputation algorithms for classification tasks," *Data Knowl. Eng.*, vol. 84, pp. 47–58, Mar. 2013.

[34] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE* (R. Meersman, Z. Tari, and D. C. Schmidt, eds.), pp. 986–996, Springer Berlin Heidelberg, Jan. 2003.

[35] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, no. 2, pp. 207–244, 2009.

[36] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," *Pattern Recognit.*, vol. 41, pp. 3600–3612, Dec. 2008.

[37] W. Liu and S. Chawla, "Class confidence weighted knn algorithms for imbalanced data sets," in *Advances in Knowledge Discovery and Data Mining* (J. Huang, L. Cao, and J. Srivastava, eds.), vol. 6635 of *Lecture Notes in Computer Science*, pp. 345–356, Springer Berlin Heidelberg, 2011.

[38] A. P. Dempster, N. M. Laird, D. B. Rubin, *et al.*, "Maximum likelihood from incomplete data via the em algorithm," *J. R. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[39] K. Kobayashi and M. U. Salam, "Comparing simulated and measured values using mean squared deviation and its components," *Agron. J.*, vol. 92, no. 2, p. 345, 2000.

[40] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[41] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

[42] C. Strobl, J. Malley, and G. Tutz, "An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests," *Psychol. Methods*, vol. 14, no. 4, pp. 323–348, 2009.

[43] S. Shih, "Random forests for classification trees and categorical dependent variables: An informal quick start r guide," tech. rep., Stanford University, Stanford, CA, Feb. 2011.

[44] A. Özçift, "Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis," *Comput. Biol. Med.*, vol. 41, pp. 265–271, May 2011.

[45] P. J. Mazzone, J. Hammel, R. Dweik, J. Na, C. Czich, D. Laskowski, and T. Mekhail, "Diagnosis of lung cancer by the analysis of exhaled breath with a colorimetric sensor array," *Thorax*, vol. 62, pp. 565–568, July 2007.

[46] T. Therneau, B. Atkinson, and B. Ripley, *rpart: Recursive partitioning and regression trees*, 2014. R package version 4.1-6.

[47] B. Ripley, *tree: Classification and regression trees*, 2014. R package version 1.0-35.

[48] J. D. Kelly Jr. and L. Davis, "A hybrid genetic algorithm for classification," *Int. J. Conf. on Artifi. Intell. (IJCAI)*, vol. 91, pp. 645–650, 1991.

[49] F. Berzal, J.-C. Cubero, D. Sánchez, and J. M. Serrano, "ART: a hybrid classification model," *Mach. Learn.*, vol. 54, pp. 67–92, Jan. 2004.

Appendix A

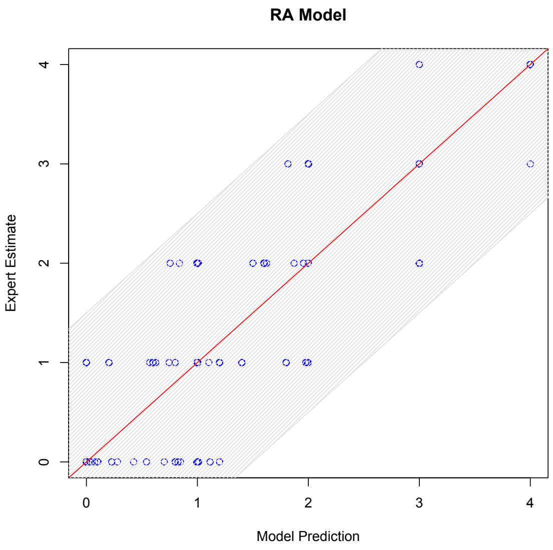
Results From Other Autoimmune Disease Classifiers

Table A.1: LOO results from the final, individual, autoimmune disease classifiers.

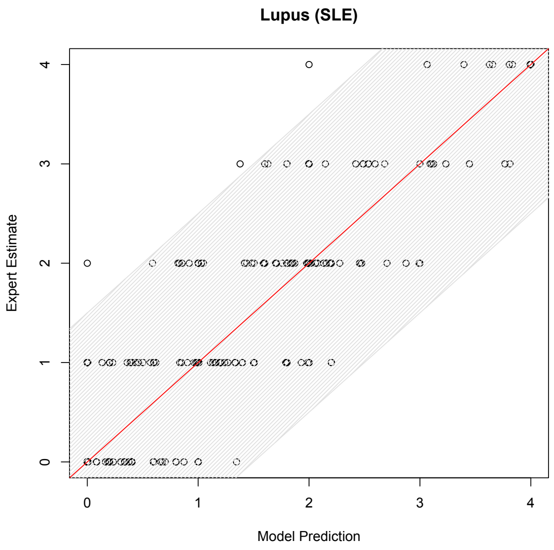
Autoimmune Disorder	**Number of Patients**	**LOO Prediction Results (%)**
Lupus (SLE)	322	99.07
Sjögren's Syndrome	305	99.34
Systemic Sclerosis	299	94.59
MCTD	293	97.93
RA	240	100.00
Antiphospholipid SX	68	98.46
Wegener's Granulomatosis	62	98.36
Microscopic Polyangiitis	57	98.21
Churg-Strauss	56	98.18
PM-DM	33	90.63

Appendix B

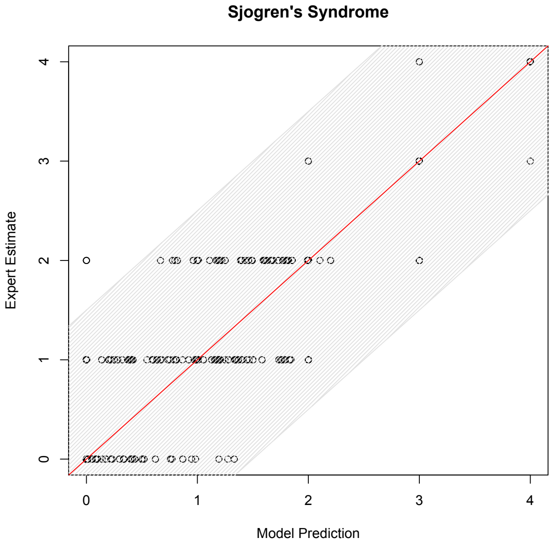
Graphically Presented Results



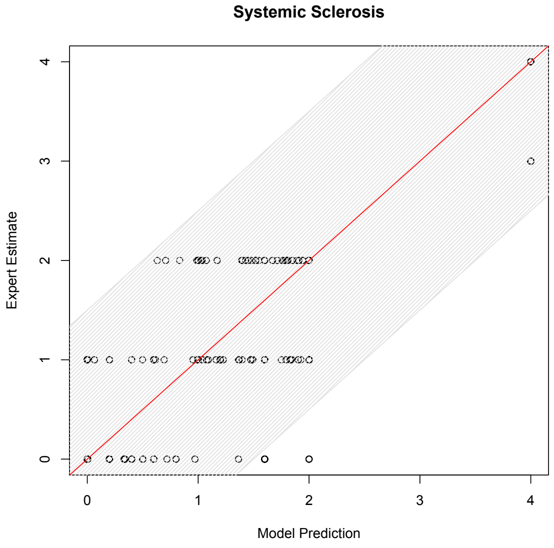
RA Model



Lupus (SLE)



Sjogren's Syndrome



Systemic Sclerosis

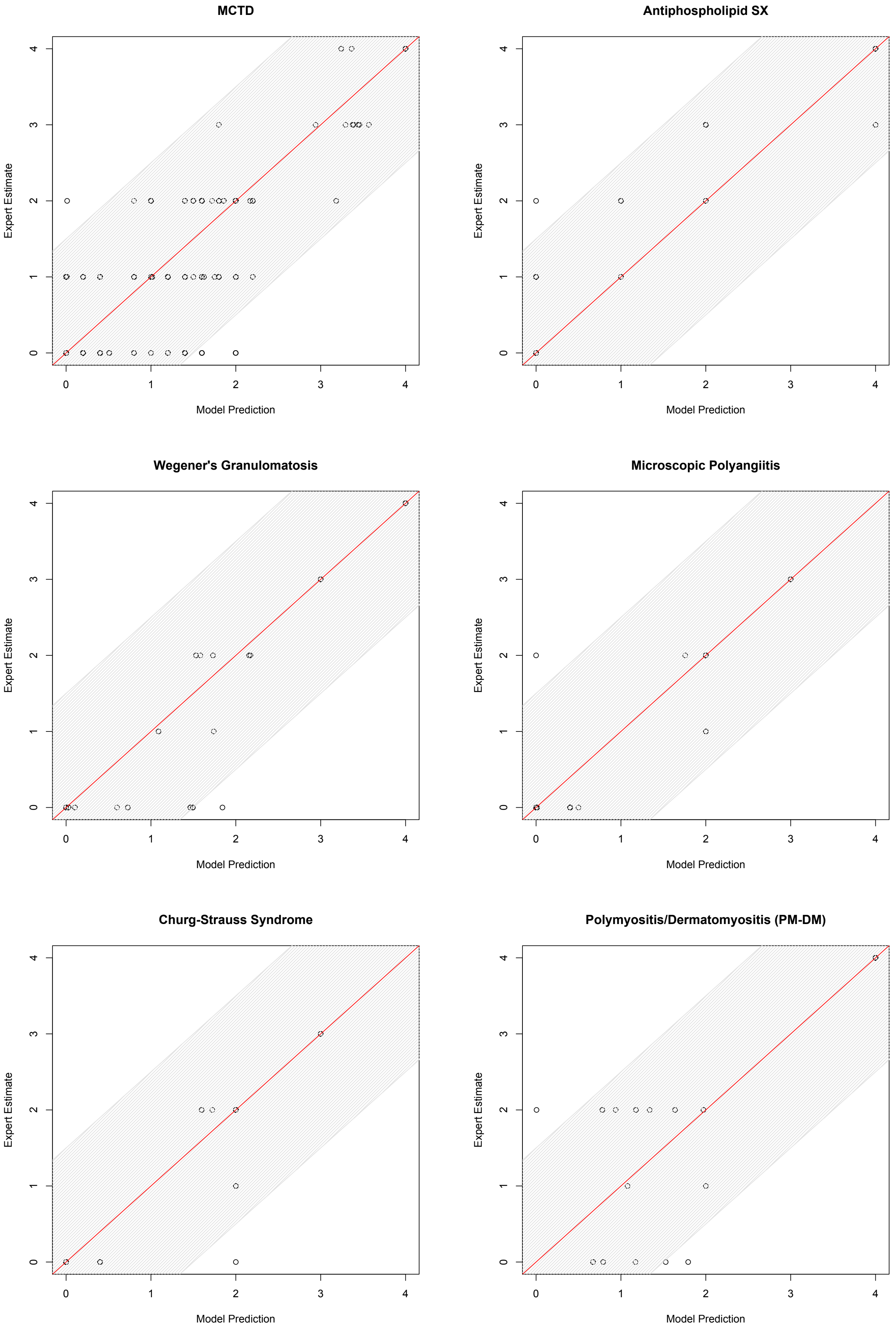
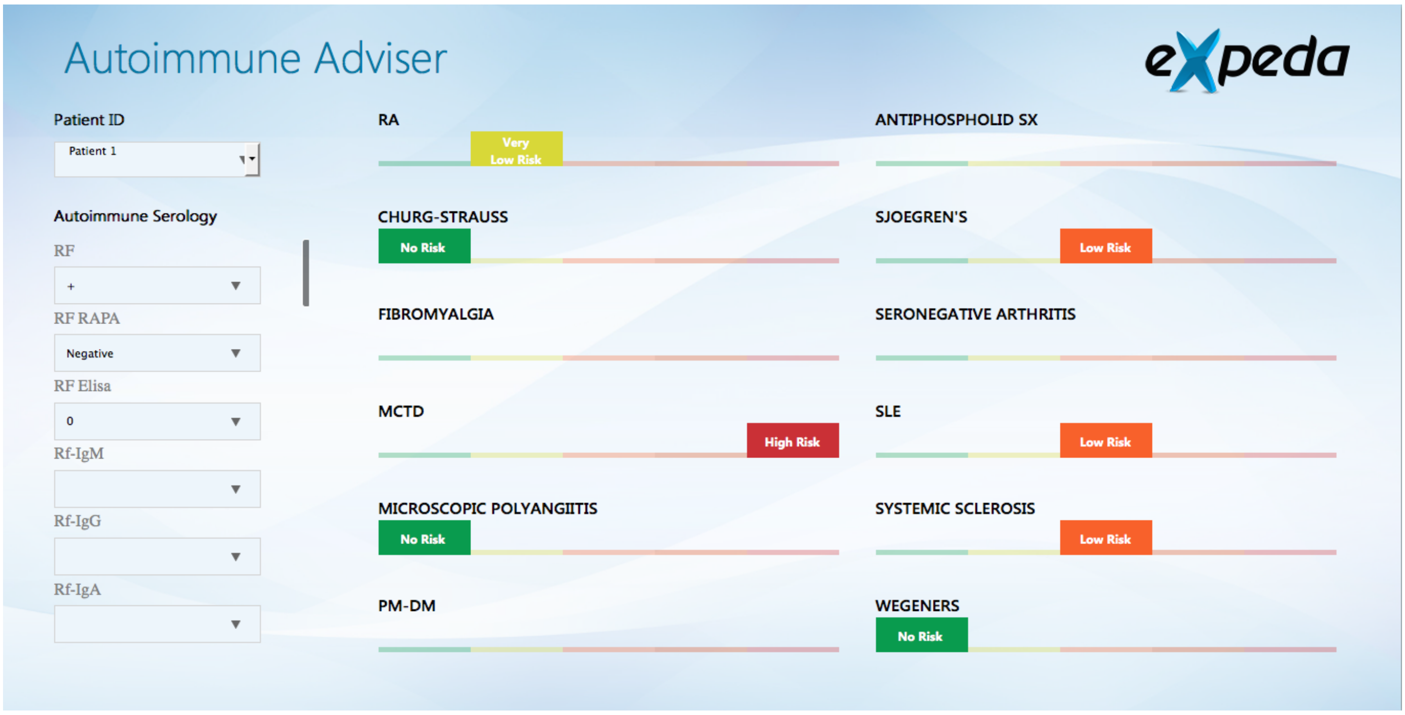


Figure B.1: Graphical presentations of LOO results from all individual classifiers.

Appendix C

Web-based Prototypes



Patient ID
Patient 1

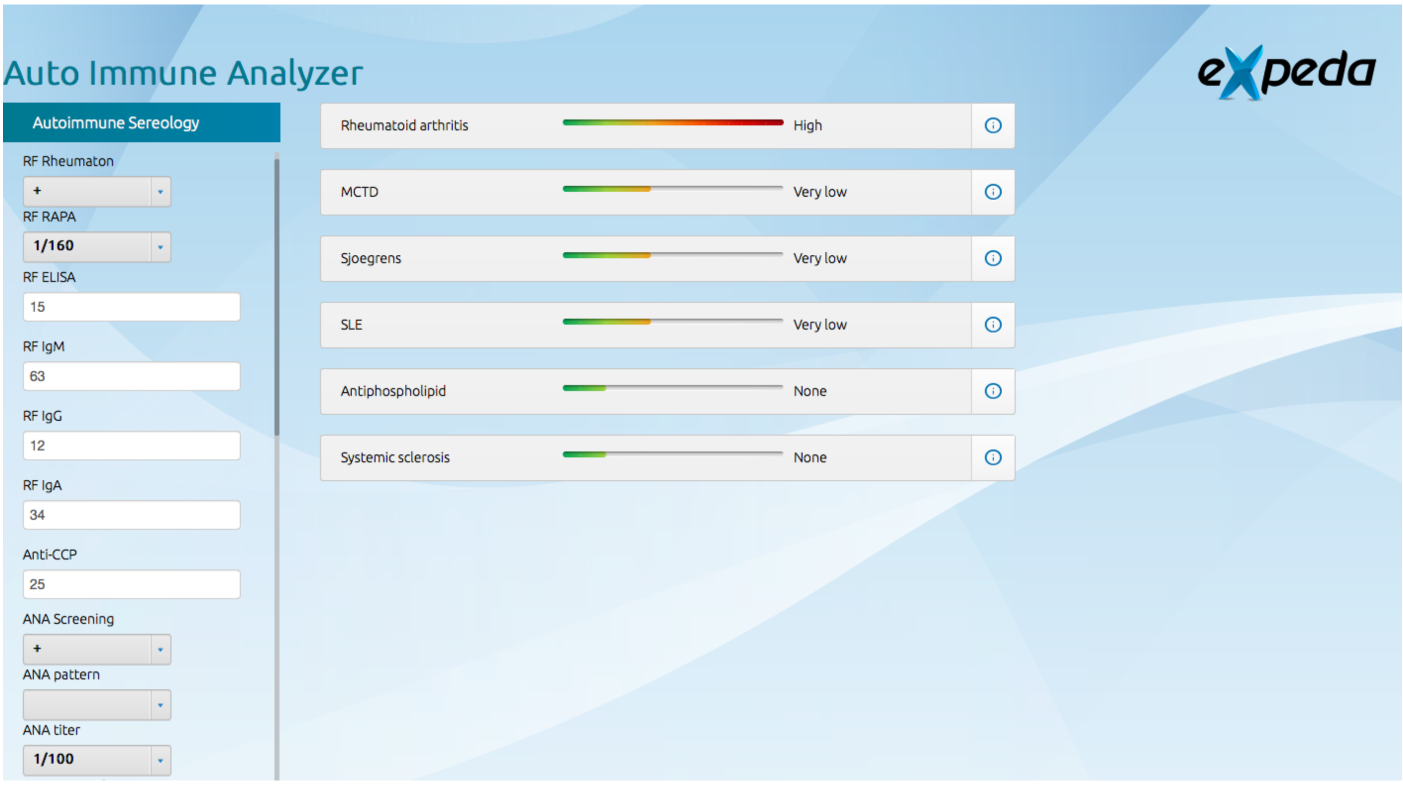
Autoimmune Serology
RF
+
RF RAPA
Negative
RF Elisa
0
Rf-IgM

Rf-IgG

Rf-IgA

RA: Very Low Risk
ANTIPHOSPHOLID SX
CHURG-STRAUSS: No Risk
SJOEGREN'S: Low Risk
FIBROMYALGIA
SERONEGATIVE ARTHRITIS
MCTD: High Risk
SLE: Low Risk
MICROSCOPIC POLYANGIITIS: No Risk
SYSTEMIC SCLEROSIS: Low Risk
PM-DM
WEGENERS: No Risk

Figure C.1: Interface for the web-based classifiers, version 1.



Auto Immune Analyzer

Autoimmune Sereology

RF Rheumaton: +

RF RAPA: 1/160

RF ELISA: 15

RF IgM: 63

RF IgG: 12

RF IgA: 34

Anti-CCP: 25

ANA Screening: +

ANA pattern:

ANA titer: 1/100

Rheumatoid arthritis	High
MCTD	Very low
Sjoegrens	Very low
SLE	Very low
Antiphospholipid	None
Systemic sclerosis	None

Figure C.2: Interface for the web-based classifiers, version 2.

School of Science and Engineering
Reykjavík University
Menntavegi 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.reykjavikuniversity.is
ISSN 1670-8539