Patient-specific management of adrenal incidentalomas using predictive and prescriptive analytics

Visser, J.B.R.

Award date:
2016

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
Patient-specific Management of Adrenal Incidentalomas using Predictive and Prescriptive Analytics

Job Visser BSc.
0674699

Supervisors:
prof. dr. ir. Uzay Kaymak (TU/e)
dr. Rui Jorge de Almeida (TU/e)
dr. Jan-Jaap Visser MSc MSc (Erasmus MC)
drs. Martijn van der Meijden (EY)

Eindhoven, January 20, 2016
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>4</td>
</tr>
<tr>
<td>List of Tables</td>
<td>6</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>1.1 Problem Statement</td>
<td>15</td>
</tr>
<tr>
<td>1.2 Research Objective</td>
<td>16</td>
</tr>
<tr>
<td>1.3 Research Questions</td>
<td>18</td>
</tr>
<tr>
<td>1.4 Research Methodology</td>
<td>18</td>
</tr>
<tr>
<td>1.5 Scope</td>
<td>19</td>
</tr>
<tr>
<td>1.5.1 Decisions</td>
<td>20</td>
</tr>
<tr>
<td>1.5.2 Assumptions</td>
<td>20</td>
</tr>
<tr>
<td>1.6 Company Collaboration</td>
<td>21</td>
</tr>
<tr>
<td>1.7 Thesis Outline</td>
<td>23</td>
</tr>
<tr>
<td>1.8 Contribution to Existing Research</td>
<td>24</td>
</tr>
<tr>
<td>2 Management of Adrenal Incidentalomas</td>
<td>25</td>
</tr>
<tr>
<td>2.1 AACE/AAES International Guideline</td>
<td>25</td>
</tr>
<tr>
<td>2.2 Erasmus MC Daily Practice</td>
<td>28</td>
</tr>
<tr>
<td>3 Research Methods for Predictive Modeling</td>
<td>31</td>
</tr>
<tr>
<td>3.1 Methodology for Predictive Modeling: CRISP-DM</td>
<td>31</td>
</tr>
<tr>
<td>3.2 Data Collection</td>
<td>34</td>
</tr>
<tr>
<td>3.2.1 Patient Identification Using Natural Language Processing</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Data Acquisition</td>
<td>37</td>
</tr>
<tr>
<td>3.2.3 Data Description</td>
<td>40</td>
</tr>
<tr>
<td>3.3 Data Cleaning</td>
<td>44</td>
</tr>
<tr>
<td>3.3.1 Missing Value Imputation</td>
<td>44</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Phases of the CRISP-DM reference model ............................................ 19
1.2 Organogram of Erasmus MC people involved in the project ...................... 22

2.1 AACE/AAES algorithm for the management of an adrenal incidentaloma ... 27
2.2 Erasmus MC daily practice ................................................................. 30

3.1 CRISP-DM hierarchical breakdown ....................................................... 31
3.2 CRISP-DM mapping from generalized to specialized tasks ................. 33
3.3 Automated patient identification pipeline ............................................. 35
3.4 Workflow graph of work-up data collection ......................................... 39
3.5 Example of the patient dataset ............................................................ 40
3.6 Example of the ICD-9 dataset .............................................................. 41
3.7 Example of the measurements dataset ................................................... 41
3.8 Example of the blood type dataset ...................................................... 41
3.9 Example of the medication dataset ...................................................... 42
3.10 Example of the lab test dataset .......................................................... 42
3.11 Missing value imputation using KNNimpute ....................................... 45
3.12 Adrenal incidentaloma work-up event timeline .................................... 49

4.1 Box plots of size and age ................................................................. 60
4.2 Box plots of endo_int, fu1_int, fu2_int, fu3_int ..................................... 60
4.3 Histogram of size ............................................................................. 62
4.4 Histogram of age .............................................................................. 62
4.5 Correlation heatmap of continuous variables ....................................... 63
4.6 Scatter histogram of size and age grouped on clin_rel ......................... 64
4.7 Scatter plot matrix of biochemical screening lab tests grouped by clin_rel ... 65
4.8 Missing value analysis biochemical screening data ................................ 66
4.9 Missing value analysis measurements data ........................................ 67

5.1 Architecture of adaptive neuro-fuzzy inference system ......................... 72
List of Tables

3.1 Most frequent patterns in report sentences ........................................ 36
3.2 Adrenal incidentaloma class concept hierarchy .............................. 36
3.3 SVM classifier performance for patient identification ..................... 37
3.4 Summary of data sources ............................................................. 40
3.5 Work-up variables ........................................................................ 43
3.6 Variable descriptions datasets M1 and M2 .................................. 51
4.1 Frequency table of find_class ......................................................... 56
4.2 Frequency table of endo ............................................................... 56
4.3 Cross-tabulation of aldos and endo ............................................... 56
4.4 Cross-tabulation of hyp_med and aldos for dataset 2 ...................... 57
4.5 Frequency table of gr_sign ........................................................... 57
4.6 Frequency table of diag .............................................................. 57
4.7 Frequency table of clin_rel .......................................................... 58
4.8 Frequency table of death_6mon .................................................... 58
4.9 Descriptive statistics of continuous variables ............................... 59
4.10 Descriptive statistics of biochemical screening tests ..................... 61
5.1 Logistic regression model M1 ......................................................... 80
5.2 Logistic regression model M2 ......................................................... 82
6.1 M1: Model comparison on the over-sampled test data ...................... 101
6.2 M1: Model comparison on the unbalanced validation data ................ 101
6.3 M2: Model comparison on the over-sampled test data ....... 102
6.4 M2: Model comparison on the unbalanced validation data ....... 102
7.1 Confusion matrix M1 at $P_t^{M1} = 1\%$ ........... 114
7.2 Confusion matrix M2 at $P_t^{M2} = 1\%$ ........... 114
7.3 Confusion matrix M1 at $P_t^{M1} = 3\%$ ........... 114
7.4 Confusion matrix M2 at $P_t^{M2} = 5\%$ ........... 114
9.1 Frequency table of find_type ....................... 126
9.2 Frequency table of gender ......................... 126
9.3 Frequency table of hypertension .................. 127
9.4 Frequency table of hyp_med ....................... 127
9.5 Frequency table of aldos ......................... 127
9.6 Cross-tabulation of hyp_med and hyp_med_pre ....... 127
9.7 Cross-tabulation of endo and fu1 ................. 127
9.8 Frequency table of hist_malign ................... 127
9.9 Frequency table of loc ............................ 128
9.10 Frequency table of advise ....................... 128
9.11 Frequency table of find_concl .................... 128
9.12 Frequency table of fu_unrel ..................... 128
9.13 Frequency table of dead ......................... 129
9.14 Frequency table of adrl ......................... 129
9.15 Descriptive statistics of routine lab tests ........... 129
Executive Summary

This research is part of the integrated diagnostics project and has been conducted at Erasmus University Medical Center. The integrated diagnostics project strives for seamless collaboration among diagnostic specialists. Its goal is to reduce the time to diagnosis, expenses of diagnostic processes and provide clinicians with practical and actionable results. At Erasmus MC management of adrenal incidentalomas is suspected to be highly variable in terms of guideline adherence. Therefore it was identified as the low hanging fruit for the integrated diagnostics project and is the focus of this research.

Adrenal Incidentalomas

Adrenal incidentaloma (AI) is an adrenal mass, larger than 1 cm in diameter, detected on imaging studies performed for other indications than adrenal disease[1]. Because of the increasing use of computed tomography (CT) and recent advancements in imaging technology, detection of incidentalomas has increased[2]. In approximately 6% of all abdominal CT scans an incidentaloma of the adrenal gland is discovered[3]. The likelihood of AI for an individual patient increases with age up until 10% at an age of 70 or more[4].

In spite of their frequent finding most incidentalomas are benign and nonfunctional tumors (80%). However, some incidentalomas (15%) cause hormonal hypersecretion or appear to be malignant (<5%)[5]. Therefore follow-up is needed for three reasons: (1) to detect appearance of a hormonal hyperfunction, (2) to assess if the mass has a malignant histology, and (3) to determine if there is growth of more then 1 cm in one year. A finding of either one of these is a clinically relevant outcome and would prompt surgical removal of the adrenal. AI management resembles a management dilemma where the challenge is to recognize and treat the small percentage of clinically relevant AI that pose a significant risk.

Among the most controversial issues is that of follow-up of the large group of 80% nonfunctioning, benign-appearing masses[6]. Follow-up strategies range from a minimalist approach using a single scan within about 6 months to annual CT scanning for up to 4 or 5 years[7]. By evalua-
tion these incidental findings, patients are exposed to a certain dose of radiation from CT scans. This results in a risk of cancer induction that is similar to that of the adrenal becoming malignant [8]. Besides that the chance of an AI, initially diagnosed as benign and non-functional, becoming malignant or hormonally active is extremely low [9]. Additionally, for this large group of nonfunctional benign-appearing tumors current guidelines only offer one moment to decide whether to continue or stop work-up; at the moment of finding (M1). This decision is based on just the size of the masses, from that point on every patient is subjected to the same expensive cascade of tests and procedures [9][10]. For every patient however, additional information is generated for every follow-up that is performed. For example after biochemical evaluation (M2) which screens for the presence of a hormonal hyperfunction or after subsequent imaging follow-ups. In order to avoid unnecessary procedures, reduce risks and to limit costs clinical management of patients with AI should be tailored to the individual patient [10].

Predictive Analytics

The aforementioned points indicate that there is demand for a patient-specific management approach that offers multiple moments at which we want to assess whether or not to continue AI work-up. This work utilizes prediction models to predict if an individual patient is at risk of a clinically relevant outcome. Clinical relevance refers to either one of the following outcome scenario’s: (1) hormonal hyperfunction diagnosis, (2) adrenocortical carcinoma, or (3) significant growth of the mass.

The CRISP-DM framework is used for development of predictive models. Patient identification is carried out using text mining to identify all adrenal incidentaloma cases that were found or received follow-up from 2010 till 2012. Patient data, lab test data, medication data and work-up data are collected for 643 patients. The dataset is heavily imbalanced given that just 5.3% of incidentalomas are clinically relevant. We propose to predict if a patient is clinically relevant at the moment of AI finding (M1) and after biochemical evaluation (M2). First a validation set, with the original class distribution, is constructed based on a 15% hold-out sample. The remaining data is used for training and testing the prediction models with 10-fold cross-validation. Lasso logistic regression and random forest were used to select which subsets of variables has most predictive power for the models. To overcome the problem of unbalanced data the synthetic minority over-sampling technique (SMOTE) is used [11]. Hence a 50-50 class distribution of the target variable clinically relevant is generated by interpolation between existing cases.
The prediction task is posed as a classification problem. Logistic regression, random forest and Takagi–Sugeno fuzzy inference systems are the algorithms used to construct the prediction models. 10-fold cross-validation is used to better estimate the performance of the models. The performance of these models is evaluated based on the over-sampled test sets as well as the validation dataset with original class distribution of the target variable.

**Prescriptive Analytics**

In order to assess the clinical usefulness of the prediction models decision curve analysis is applied at M1 and M2. The most appropriate prediction model for use in clinical practice is integrated with the Erasmus MC daily practice for AI management. This results in a novel prescriptive model for patient-specific management of adrenal incidentalomas. Results of the prescriptive model on the M1 validation data (n=96) indicate that 15% of patients at M1 can be saved from unnecessary follow-up while correctly detecting all clinically relevant patients. At M2 25% of the validation set (n=16) are saved from unnecessary follow-up without omitting any clinically relevant patients.

**Conclusions and Future Work**

This research resulted in three predictive models which can be used to predict if a patient will have a clinically relevant outcome at multiple moments during AI work-up. Results indicate that their performance on the over-sampled test sets is very good. Results on the original validation set are good but tend to be highly fluctuating. This is due to the size of the validation set and the imbalanced class distribution of the target variable. By integrating the most appropriate model in terms of clinical usefulness with the AI management guideline a novel prescriptive model for patient-specific adrenal incidentaloma management is introduced. The prescriptive model shows promising results.

This research is expected to profit significantly from a larger dataset for training, testing and validating the proposed models. Additionally the dataset can be refined by collecting variables from CT images. With the enhanced dataset it is possible to add additional decision moment to the prescriptive model based on the data collected from imaging follow-ups. In order to ensure general applicability of the prediction models external validation is
Acknowledgements

This is the page that I use to show appreciation to those who contributed in conducting this research, but first let me sketch the situation for you. This research involved three groups of stakeholders: TU/e, Erasmus MC and EY. A perfect combination when you want to learn about rigour versus relevance and 'traditional' biostatistics versus machine learning. However my personal goals were to finally learn how to program, make a contribution to a real world data mining project and actually graduate somewhere along the way.

I would not have been able to perform the work in front of you without the help of several people. First I would like to thank Uzay Kaymak (TU/e) for his feedback during our meetings. Normally upon answering, people tend to give you answers. Instead Uzay asks another question to make you approach the problem from another perspective, this greatly increased the quality of this work as well as the mental workload. Additionally I would like to thank Rui de Almeida (TU/e) for his enthusiasm and expertise, especially regarding the development of the prescriptive model. I would like to thank Jan-Jaap Visser (Erasmus MC) for his pleasant and pragmatic way of working. His door was always open for vivid discussions, with topics ranging from adrenal glands to starting a startup. Also I would like to thank Martijn van der Meijden (EY) and Erik Vermeulen (EY) for providing me with this opportunity and supervision throughout the project.

Special thanks go to Romy de Haan (Erasmus MC) for the tremendous amount of effort that she put into the manual data collection. Without her this project would have taken another 3 months extra. Many thanks to Myriam Hunink (Erasmus MC), Ewoud Pons (Erasmus MC) and the rest of the Integrated Diagnostics project team, for the feedback and guidance.

Finally I would like to thank all persons that have been involved with the project in one way or another, ranging from the ones that raised me to the ones that offered me temporary accommodation when I was homeless.
Definitions

This section provides definitions and abbreviations for readers with a non-medical background that are used in this review. This information was retrieved from the McGraw-Hill Concise Dictionary of Modern Medicine[12].

Gland: The internal or hormonal secretion of a ductless gland.

Adrenal Gland: Endocrine glands that sit at the top of the kidneys; in humans, the right adrenal gland is triangular shaped, while the left adrenal gland is semilunar shaped. They are chiefly responsible for releasing hormones in response to stress through the synthesis of corticosteroids such as cortisol and catecholamines such as epinephrine (adrenaline) and norepinephrine.

Adrenal mass: Adrenal tumor containing benign or malignant neoplasms (abnormal growth of tissue) of the adrenal gland, several of which are notable for their tendency to overproduce endocrine hormones.

Adrenal Incidentaloma (AI): An adrenal mass, larger than 1 cm in diameter, detected on imaging studies performed for other indications than adrenal disease[1].

Adenoma: A benign epithelial neoplasm in which the tumor cells form glands or gland-like structures.

Myelolipoma: A rare benign tumor of the adrenal gland composed of adipose tissue, lymphocytes, and primitive myeloid cells.

Lesion: Any pathological or traumatic discontinuity of tissue or loss of function of a part, i.e. any localized, abnormal structural change in the body.

Cyst: A cyst is a closed sac, having a distinct membrane and division compared to the nearby tissue. It may contain air, fluids, or semi-solid material.

Malignant: Tending to become progressively worse and to result in death.

Benign: Not recurrent and favorable for recovery with appropriate treatment. The opposite of malignant.

Hypercortisolism (i.e. Cushing’s Syndrome (CS)): Describes the signs and symptoms associated with prolonged exposure to inappropriately high levels of the hormone cortisol. Signs and symptoms may include: high blood pressure, abdominal obesity but with thin
arms and legs, reddish stretch marks, a round red face, a fat lump between the shoulders, weak muscles, weak bones, acne, and fragile skin that heals poorly. This can be caused by taking glucocorticoid drugs, or diseases that result in excess cortisol, adrenocorticotropic hormone (ACTH), or CRH levels.

**Subclinical Cushing Syndrome (SCS)**: An ill-defined endocrine disorder that may be observed in patients bearing an incidentally found adrenal adenoma. The concept of SCS stands on the presence of ACTH-independent cortisol secretion by an adrenal adenoma, that is not fully restrained by pituitary feedback[13].

**Metanephrine**: A metabolite of epinephrine (adrenaline) created by action of catechol-O-methyl transferase on epinephrine. The measurement of plasma free metanephrines is the best tool in the diagnosis of pheochromocytoma.

**Normetanephrine**: A metabolite of norepinephrine created by action of catechol-O-methyl transferase on norepinephrine. It is excreted in the urine and found in certain tissues. It is a marker for catecholamine-secreting tumors such as pheochromocytoma.

**Pheochromocytoma (PCC)**: A neuroendocrine tumor of the medulla of the adrenal glands, or extra-adrenal chromaffin tissue that failed to involute after birth and secretes high amounts of catecholamines, usually noradrenaline (norepinephrine), and adrenaline (epinephrine) to a lesser extent.

**Aldosteronism**: A medical condition where too much aldosterone is produced by the adrenal glands, which can lead to lowered levels of potassium in the blood also known as hypokalemia.

**Primary Aldosteronism (PA)**: Previously thought to be most commonly caused by an adrenal adenoma, termed Conn’s syndrome. However, recent studies have shown that bilateral idiopathic adrenal hyperplasia is the cause in up to 70% of cases. Differentiating between the two is important as this determines treatment. Adrenal carcinoma is an extremely rare cause of primary aldosteronism.

**Hypertension**: High blood pressure, sometimes called arterial hypertension, is a chronic medical condition in which the blood pressure in the arteries is elevated.

**Pheochromocytoma (PCC)**: A functional chromaffinoma, usually benign, derived from adrenal medullary tissue cells and characterized by the secretion of catecholamines, resulting in hypertension, which may be paroxysmal and associated with attacks of palpitation, headache, nausea, dyspnea, anxiety, pallor, and profuse sweating.

**Urine Free Cortisol (UFC)**: Urinary free cortisol measurements are used primarily in the diagnosis of Cushing’s syndrome.
Adrenocortical Carcinoma (ACC): A rare disease in which malignant (cancer) cells form in the outer layer of the adrenal gland.

Resection: The removal by surgery of all or part of an organ or other body structure.

Work-up: The procedures done to arrive at a diagnosis, including history taking, laboratory tests, radiologic imaging, etc.

Clinical Diagnosis: Determination of the nature of a cause of a disease based on signs, symptoms, and laboratory findings during life.

Differential Diagnosis: The determination of which one of several diseases may be producing the symptoms.

Endocrine: Relating to or denoting glands which secrete hormones or other products directly into the blood.

Endocrinology: The study of hormones, the endocrine system, and their role in the physiology of the body.

Hounsfield Units (HU): A linear transformation of the original linear attenuation coefficient measurement into one in which the radiodensity of distilled water at standard pressure and temperature (STP) is defined as zero Hounsfield units (HU), while the radiodensity of air at STP is defined as -1000 HU. HU is a unit that represents the dilution values of basic pixel in CT images.

Hemorrhage: Bleeding, technically known as hemorrhaging, is the loss of blood or blood escaping from the circulatory system.

Comorbidity: Either the presence of one or more disorders (or diseases) in addition to a primary disease or disorder, or the effect of such additional disorders or diseases.
Chapter 1

Introduction

This chapter contains an in depth outline of the approach chosen for this thesis starting with the problem statement and research objective. Within this research predictive modeling has a prominent role, therefore the outcome variable (i.e. what do we want to predict?) is elaborated upon in the third section. Then the research questions are posed followed by the methodology and scope of the project. The collaboration between the different parties involved is explained in company collaboration. The chapter ends with a detailed thesis outline and contribution to existing research.

1.1 Problem Statement

Adrenal incidentaloma (AI) is an adrenal mass, larger than 1 cm in diameter, detected on imaging studies performed for other indications than adrenal disease[1]. Because of the increasing use of computed tomography (CT) for the upper abdominal section and recent advancements in imaging technology, detection of incidentalomas has increased[2]. The likelihood of AI increases with age up until 10% at an age of 70 or more[4].

In spite of their frequent finding most incidentalomas are benign and nonfunctional tumors (70-80%). However, some incidentalomas (5-15%) cause hormonal hypersecretion or appear to be malignant (<5%)[5]. When an AI causes a hormonal hyperfunction there are typically four related diagnoses: hypercortisolism (Cushing’s syndrome), subclinical autonomous cortisol hypersecretion (subclinical Cushing’s syndrome), pheochromocytoma, or primary aldosteronism (Conn’s Syndrome)[1]. Among hormonal diseases subclinical Cushing’s syndrome (SCS) occurs most often, in 5–20% of patients who have no endocrine signs or symptoms, analysis reveals subclinical Cushing’s syndrome[14]. SCS can potentially develop into overt Cushing’s syndrome (CS). Pheochromocytoma makes up about 1.5–14% of incidentalomas, adrenocortical carci-
noma (ACC) is found in 1.2–11%, aldosterone-producing adenoma is found in 1.6–3.3%, and adrenal metastases are found in 1–18%[15]. Adrenocortical carcinoma (ACC) is the smallest fraction yet most dangerous among possible diagnoses. The prognosis for patients with this kind of cancer is better when the adrenal gland is resected completely at an early stage. Total resection is referred to as adrenalectomy. Previous studies indicate that tumor size is a strong predictor for malignancy[16][14].

Follow-up of incidentalomas is needed for three reasons: (1) to detect appearance of a hormonal hyperfunction, (2) to assess if the mass has a malignant histology, and (3) to determine if there is growth of more than 1 cm in one year detected CT imaging. A finding of either one of these is a clinically relevant outcome and would prompt surgical removal of the adrenal. AI management resembles a management dilemma where the challenge is to recognize and treat the small percentage of clinically relevant AI that pose a significant risk. Among the most controversial issues is that of follow-up of the large group of 80% nonfunctioning, benign-appearing masses[6]. Follow-up strategies range from a minimalist approach using a single scan within about 6 months to annual CT scanning for up to 4 or 5 years[7]. By evaluation of these incidental findings, patients are exposed to a certain dose of radiation from CT scans. This can result in a potential risk of cancer induction that is similar to that of the AI becoming malignant[8]. Besides that the chance of an AI, initially diagnosed as benign and non-functional, becoming malignant or hormonally active is extremely low[9]. Additionally, for this large group of nonfunctional benign-appearing tumors current guidelines only offer one moment to decide whether to continue or stop work-up; at the moment of finding (M1). This decision is based on just the size of the masses, from that point on every patient is subjected to the same expensive cascade of tests and procedures[9][10]. For every patient however, additional information is generated for every follow-up that is performed. For example after biochemical evaluation (M2) which screens for the presence of a hormonal hyperfunction or after subsequent imaging follow-ups. In order to avoid unnecessary procedures, reduce risks and to limit costs clinical management of patients with AI should be tailored to the individual patient[10]. This way unnecessary use of diagnostics can be reduced.

1.2 Research Objective

There is demand for a patient-specific management approach that offers multiple moments at which we want to assess whether or not to continue AI work-up. This broad desire can be broken down into two main objectives.
Firstly, to develop a clinical prediction model to predict if an adrenal incidentaloma is *clinically relevant*. Secondly, to derive from this prediction model a prescriptive model for patient-specific management of patients with adrenal incidentalomas.

**Defining Clinically Relevant**

Now that the objective is clear it is time to shed some light on what exactly we are trying to predict. During the early stages of the project there have been many brainstorm sessions about the choice of outcome variable. Among the options were adrenalectomy, growth of the tumor and relevant diagnoses. During data collection all of these individual subsets turned out to be very small. Class imbalance is problematic for most prediction algorithms, so the outcome variable needed to be reframed.

The goal of the prediction model is to predict if the patients outcome will be relevant for clinical practice, i.e. does this patient need extra treatment because it carries a higher risk? This refers to risk prediction. Probability estimation is key to the area of risk prediction, which is growing in importance in medicine, where personalized medicine becomes more and more possible through the combination of classical risk predictors and biomarkers[17]. What exactly is an accurate estimate of class membership probability is a subject of debate beyond the scope of this research. Throughout this thesis when speaking about the output of any of the prediction models for an individual patient, given an outcome (e.g. *not clinically relevant*) we refer to likelihood. Probability is used when describing a function of the outcome given a fixed parameter value (e.g. sensitivity of a model).

We want to predict the likelihood that an individual patient is *clinically relevant*. Based on AACE/AAES guidelines from 2009 the following outcomes are considered to be *clinically relevant*[18]:

- **Relevant diagnosis**: Subclinical Cushing’s Syndrome (SCS), Cushing’s Syndrome (CS), Primary Aldosteronism (PA), Pheochromocytoma (FEO), Adrenocortical Carcinoma (ACC)
- **Lesion growth**: \( \geq 1 \) cm in one year
- **Adrenalectomy**: surgical removal of one or both adrenal glands

So *clinically relevant* refers to one or more of the above events occurring at any point in the adrenal incidentaloma work-up.
1.3 Research Questions

In order to avoid unnecessary procedures, reduce risks and to limit costs clinical management of patients with AI should be tailored to the individual patient[10]. This research is linked to a major paradigm shift in healthcare from evidence-based medicine to personalized medicine. Using data mining techniques we tailor a patient’s work-up based on his personal characteristics.

This research carries one main research question in line with the research objective.

RQ 1 Are prediction models based on machine learning algorithms suitable for predicting a clinically relevant outcome of adrenal incidentalomas?

To support the rather broad research question above four research subquestions can be defined.

RSQ 1 Which data can be obtained from the Erasmus MC hospital databases to develop the clinical prediction model?

RSQ 2 Which features have good predictive power for the clinical prediction models?

RSQ 3 Can a prescriptive model for patient-specific management of patients with adrenal incidentalomas be derived from this clinical prediction model?

1.4 Research Methodology

This research is conducted using the Cross Industry Standard Process for Data Mining (CRISP-DM). It is a data mining methodology that captures the commonly used approaches used by data mining experts[19]. CRISP-DM can be considered the leading methodology based on responses of industry data miners. Polls conducted in 2002, 2004, 2007 and 2014 at the website KDNuggests, which is popular in the data mining community, show that it was used by generally 3 - 4 times as many people compared to the other data mining frameworks like SEMMA[20][21][22][23].

Figure 1.1 provides an overview of the life-cycle of a data mining project. It depicts the phases of a data mining project. Every phase holds its tasks, and the relationships between these tasks. The model contains many iterative steps where moving back and forth between phases is inevitable because of the progressive insights during the project. In fact the outer circle also reflects the cyclical nature of a data mining project which does not end after deployment.

The six phases are:

Business Understanding : Involves understanding the project objectives and requirements from a business perspective. In the end of this phase these have to be translated to a data mining problem.
Data Understanding: Involves data collection, data exploration and an assessment of its quality.

Data Preparation: Covers all activities to construct the final dataset that will be fed to the models. Tasks include: feature selection, transformation and cleaning.

Modeling: Here the modeling techniques are selected and configured. Typically several data mining techniques are used for the same problem.

Evaluation: This stage evaluates and reviews the steps taken so far. A choice on the use of the data mining results should be made.

Deployment: The knowledge gained will need to be organized and presented in a way that the Erasmus MC can use it. Depending on the requirements this can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise.

1.5 Scope

The scope of this research project is defined by the following points:

- The research limits itself to patients having adrenal incidentalomas (AI), i.e. belonging to the AI patient group.
- The research population generated consists of patients who’s AI has been found or received follow-up in the period from 2010 till 2012.
- The research does not collect diagnostic work-up data from patients that were referred by an external institution.
The research does not collect diagnostic work-up data from patients that didn’t have adrenal incidentalomas, i.e. patients identified for potentially having adrenal incidentalomas that turn out not to.

Patients with suspected metastatic disease were excluded from the study. Work-up data of these patients was collected though for future research outside the scope of this project.

1.5.1 Decisions

To remain within the scope of the project certain decisions had to be made about the level of detail captured by the variables. This section describes all relevant decisions that were made under strict guidance of Erasmus MC supervisor and radiologist dr. J.J. Visser.

1. The nominal variable history of malignancy hist_malign only describes if a primary malignancy had occurred in a patient’s clinical history. Most AI studies exclude patient with a history of malignancy. We have to choose to include them because unless they are suspected of metastasis all patients have to be treated equally.

2. The AI finding class find_class can only be external (EXT) if there is explicit proof provided by the Electronic Health Record (EHR). When external treatment is suspected this can be recorded under the variable report discrepancy rep_discr.

3. To find out if a patient received biochemical screening laboratory test data are used to detect if that unique combination of tests has been performed. This unique combination consists of all lab tests that check for pheochromocytoma.

1.5.2 Assumptions

1. Adrenal Incidentaloma: Adrenal mass > 1 cm in diameter of the longest axis incidentally discovered during CT or MRI-scanning, performed for reasons other than an evaluation for adrenal disease[4]. The size is collected from the radiology reports, this introduces bias into the study since we rely on human interpretation resulting in an error margin in measurements. It must be noted though that the borders of lesion can appear very vague. This can make it very hard, even for an experienced radiologist, to correctly measure an adrenal lesion.

2. If the finding report mentions "suspected for metastasis" or "possibly metastatic disease" and the adrenal mass respects the properties of assumption 1 the AI finding class of these patients is metastasis (META).
3. An adrenal incidentaloma can only be discovered on a CT, MRI or ECHO on which the adrenal is displayed. i.e. incidentalomas found during surgery are not considered.

4. If follow-up treatment can’t be found in the Electronic Health Record (EHR) then no follow-up exists.

5. If the variable growth gr equals 0 there is no growth of the lesion and therefore it is assumed that no significant growth gr_sign (\( \geq 1 \text{ cm in one year} \)) took place either.

7. If there are no biochemical lab tests available it is assumed that the patient didn’t visit the endocrinologist.

8. When adrenalectomy is performed or when the patient is diagnosed with a relevant diagnosis (ACC, FEO, PA, CS, SCS) the patient exits AI work-up.

1.6 Company Collaboration

The research has been conducted in fulfillment of obtaining the degree of Master of Science and consisted of an internship at EY Netherlands and Erasmus MC. Professor of information systems in the healthcare prof. dr. ir. Uzay Kaymak is first supervisor for this project. His research is on fuzzy modeling, decision making and computational intelligence methods for advanced decision support. Second supervisor dr. Rui Jorge de Almeida is responsible for judging the final thesis, together with the first supervisor.

EY  This research project is performed during an internship at EY (formerly Ernst & Young) in Amsterdam, the Netherlands. EY is a multinational company delivering assurance, tax, transitions and advisory services. The internship is provided through the Data Analytics branch of the Financial Accounting Advisory Services (FAAS) department. This branch provides Data Analytics services for their clients (as their financial controller or as their advisor). Martijn van der Meijden is executive director of the Data Analytics department and main EY supervisor for this project.

Erik Vermeulen is director at the IT Transformation department. This department provides services that help measure, enable and improve IT effectiveness to increase the overall return on IT investment. These services include demand management, IT sourcing and outsourcing, application and infrastructure optimization and consolidation and solution architecture. Erik
is involved in several projects at Erasmus MC and provided the opportunity for this research project.

**Erasmus Medical Center** The research project takes place at the department of radiology at the Erasmus University Medical Center (Erasmus MC) in Rotterdam, the Netherlands. It is affiliated with Erasmus University Rotterdam and home to its faculty of medicine. Besides that it is the largest and one of the most authoritative scientific University Medical Centers in Europe.

The rest of this section elaborates on all people involved in this project (see figure 1.2). The department is lead by prof. G.P. Krestin and has approximately 300 employees. Besides being one of the largest radiology departments in Europe they also make a significant yearly contribution to research. Erasmus MC radiology has a high citation record, compared to the worldwide citation performance in the period 2008-2011, they were cited 1.88 times more frequently than an average paper in their field[24].

![Organogram of Erasmus MC people involved in the project](image)

**Figure 1.2:** Organogram of Erasmus MC people involved in the project

Dr. J.J. Visser is active as quality officer and radiologist, he is the main Erasmus MC supervisor for this project. Besides that he is chairman of the Integrated Diagnostics project team, the adrenal incidentaloma project is one of four running projects. J. Meijer is head of the unit research, education and training. She supports the supervision and provides feedback from an operational perspective. R. de Haan is a student intern that participates in the adrenal incidentaloma research project. Her role in the project handles descriptive analytics (diagnostic work-up analysis) and data collection.
Prof. M. Hunink is professor of clinical epidemiology and radiology at the Erasmus MC and adjunct professor of health decision sciences at Harvard School of Public Health. She participates in the Integrated Diagnostics project meetings and provides methodological feedback on the research project. E. Pons is a PhD student under prof. M. Hunink. His research is aimed at automatic construction of dynamic and setting specific prediction rules in Radiology. The project relies on data text-mined from electronic medical records and aspires to develop a working prototype for Clinical Decision Support. Natural Language Processing (NLP) algorithms developed by E. Pons are used for the automated identification of patients with adrenal incidentalomas.

1.7 Thesis Outline

In chapter 2 the AAES/AACE international guideline for management of adrenal incidentalomas is elaborated upon. This guideline represents the most recent consensus statement that serves as a basis for the Erasmus MC daily practice, which is discussed in the second part of the chapter.

Chapter 3 outlines the research methods for predictive modeling according to the CRISP-DM methodology in general. Methods used for data collection, data preparation, data transformation and data integration are discussed more extensively. In the end of the chapter methods for feature selection and the class imbalance problem are explained.

The data understanding phase focuses on familiarization with the data, i.e. data exploration. Chapter 4 covers descriptive statistics, univariate and bivariate data analysis. Additionally a missing value analysis to assess the quality of the data is performed. At this point all preparations for predictive modeling are done.

Chapter 5 covers a detailed explanation of the predictive modeling techniques, experimental design of the models and how their performance is evaluated. Then the actual models are built and interpreted.

Next, chapter 6 evaluates the results obtained by the predictive modeling process in two parts. Firstly the results of feature selection are derived resulting in the subset of variables that have most predictive power. Secondly the performance of the prediction models based on these variables is assessed and compared.

Based on the prediction models the project moves towards development of a prescriptive model.
in chapter 7. The prediction models are assessed on their usefulness in clinical practice. Then a novel prescriptive model is introduced for patient-specific management of adrenal incidentalomas. Followed by its results, i.e. the number of patients saved from unnecessary work-up.

Final chapter 8 discusses the results of the predictive and prescriptive models, answers the research questions and identifies directions for future research. Including an exploration of possible ways to deploy the prescriptive model.

1.8 Contribution to Existing Research

There have been attempts to make the current management algorithm more specific by incorporating a risk stratification algorithm to leave less room for subjective decision making[25]. Birsen and colleagues used tumor size and Hounsfield units (HU) as parameter for decision making based on a literature review. This project contributes to existing research by using a prediction model for decision making during AI work-up. The prediction model uses statistical and machine learning algorithms which have been applied successfully in various fields[26]. Applying predictive analytics in the context of AI management based on patient-specific variables has never been done before. Besides that, a prescriptive model incorporating machine learning and medical decision making techniques in the way presented here shows how to make the translation from prediction to prescription. This creates value in the form of relevance for clinical practice.
Chapter 2

Management of Adrenal Incidentalomas

This chapter discusses the management strategy for management based on a consensus statement from 2009 which is referred to as AACE/AAES international guideline. This guideline represents the most recent consensus statement that serves as a basis for the Erasmus MC daily practice, which is discussed in the second part of the chapter.

2.1 AACE/AAES International Guideline

The American Association for Clinical Endocrinologists (AACE)/American Association of Endocrine Surgeons (AAES) medical guidelines for the management of adrenal incidentaloma are as follows[18]:

History and Physical Examination

Patients that have an AI need to undergo a clinical evaluation. The patient history examination is aimed at excluding a functional tumor. The following things are checked for:

- Hypercortisolism (Cushing’s syndrome (CS)) = substantial weight gain, easy bruisability, severe hypertension, diabetes, virilization, proximal muscle weakness or fatigue.
- Pheochromocytoma (PCC) = sudden or severe headaches, weight loss, anxiety attacks, sweating, cardiac arrhythmias or palpitations.
- Primary aldosteronism (PA) (Conn’s syndrome) = hypertension, fluid retention or a history of hypokalemia.
- History of malignancy = history of cancer, recent weight loss and a smoking history because an AI may be a metastatic lesion.
• Physical examination = blood pressure, assessment for evidence of central obesity, ecchymoses, striae, muscle wasting, hirutism or other signs of virilization.

Biochemical Evaluation

Unless the adrenal lesion is an obvious myelolipoma the patient has to undergo a biochemical evaluation. Adrenal myelolipomas are of low CT attenuation and contain fat (-10 to -20 HU). It is important to emphasize that imaging cannot distinguish between functioning and non-functioning adrenal adenomas. A hormonally hyperfunctioning adenoma needs to be surgically resected. Patients are screened for the following:

• Subclinical hypercortisolism (Subclinical Cushing’s syndrome (SCS)) = 1-mg overnight dexamethasone suppression test. SCS diagnose if serum cortisol level > 5µg/mL. A low level of ACTH or a low dehydroepiandrosterone sulfate concentration also support this. [1-mg dexa > 5µg/mL]

• Pheochromocytoma (PCC) = elevated plasma free metanephrine and normetanephrine levels and 24-hour total urinary metanephrines and fractionated catecholamines.

• Primary aldosteronism (PA) (Conn’s syndrome) = ratio of plasma aldosterone concentration (PAC) (ng/dL) to plasma renin activity (PRA) (ng/mL per hour) of > 20 while not taking spironolactone and mineralocorticoid receptor blockers. [PAC/PRA ratio > 20]

Radiologic Imaging

The primary goal here is to distinguish between: adrenal adenoma, adrenal carcinoma, pheochromocytoma or metastatic lesions. To diagnose a lesion as adenoma the presence of intracellular lipid should be checked. This can be done through density measurements on: noncontrast CT or in-phase and out-of-phase MRI. The following can be stated:

• Adenoma = [HU < 10, wash-out > 50%][27]. Lesions that have an attenuation value < 10 HU on non-contrast CT. CT scans done 60 seconds after intravenous administration of a contrast agent and then again after a 10- to 15-minute delay. Benign adrenal lesions will commonly enhance up to 80 to 90 HU and wash out more than 50% on the delayed scan, whereas other entities do not.

• Pheochromocytoma = [HU > 100 HU] This diagnostically distinguishes them from adenomas.

Some benign adrenal lesions do not have HU less than 10 and may have values of 20 to 40 HU. This result is found in lipid-poor adenomas. In these cases, a washout of > 50% often allows the diagnosis of an adenoma to be made.
Follow-up of Patients With Non-functioning Adrenal Incidentaloma

The following work-up is advised for patients with an AI smaller than 4 cm identified as a benign adenoma[18]:

- Radiographic reevaluation at 3 to 6 months and then annually for 1 to 2 years.
- Hormonal evaluation annually for 5 years.
- If lesion grows $\geq 1$ cm in one year, adrenalectomy should be performed.

The risk of the mass enlarging during 1, 2, and 5 years is 6%, 14%, and 29%, respectively, and the risk of the mass becoming hormonally active during those time periods is 17%, 29%, and 47%, respectively[18]. Figure 2.1 illustrates the guideline for surgical resection proposed by the AACE/AAES.

Figure 2.1: AACE/AAES algorithm for the management of an adrenal incidentaloma

The * in figure 2.1 = Re-image in 3 to 6 months and annually for 1 to 2 years. Repeat biochemical screening annually for 5 years. If mass grows $\geq 1$ cm or becomes hormonally active, then adrenalectomy is recommended.
2.2 Erasmus MC Daily Practice

This section describes the daily practice for the management of adrenal incidentalomas at Erasmus MC. It has been acquired through a qualitative interview with endocrinologist dr. R. Feelders.

The Erasmus MC daily practice for the management of patients with adrenal incidentalomas is based on the international guidelines as proposed by the AACE/AAES in 2009[18]. It was modified according to the experience of internal medical experts and more recent medical research. Since Erasmus MC is an academic hospital, sometimes a different work-up is preferred for research purposes. These extraordinary cases are considered out of scope for this research. The daily practice, depicted in figure 2.2 is modeled using Business Process Model and Notation 2.0 (BPMN 2.0). The primary goal of BPMN is to provide a notation that is readily understandable by all business users. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation. Note that every swim-lane refers to a different hospital department. The processes are color coded:

- Green process: AI work-up process that requires presence of the patient.
- Red process: AI work-up process linked to a clinically relevant outcome that requires presence of the patient.
- Yellow process: AI work-up process performed internally that doesn’t require presence of the patient.

The process start with a request for abdomen or thorax CT scan by the referring department. Note that scans requested because of symptoms indicating adrenal disease are excluded since they cannot be incidentalomas. After the CT scan is performed the finding is recorded in the report and an advise for follow-up is given. The referring department which requested the scan reads about the adrenal mass incidentally found and requests a visit to the endocrinologist. The main goal of the visit and subsequent biochemical screening is to determine if the adrenal lesion is hormonally active. During the visit a physical examination is performed aimed at identifying a functional tumor. The history and physical examination is conform the AACE/AAES guideline[18]. Patients at Erasmus MC are biochemically screened using the following endocrine lab tests:

- Hypercortisolism (Cushing’s Syndrome(CS)) = 1-mg overnight dexamethasone suppression test.
- Pheochromocytoma(PCC) = metanephrine and normetanephrine levels, metanephrine-to-creatinine ratio, normetanephrine-to-creatinine ratio, 24-hour total urinary metanephrines and 24-hour total urinary normetanephrines.
Primary aldosteronism (PA) (Conn’s syndrome) = plasma aldosterone concentration while not taking anti-hypertensive drugs spironolactone and mineralocorticoid receptor blockers. For a list of reference values per test see Appendix 9.5. Primary aldosteronism is only tested for if the patient has hypertension or hypokalemia, the other tests should always be performed. When the lab tests are done, a new visit to the endocrinologist is planned to discuss the test results with the patient. If the adrenal has a hormonal hyperfunction or if the lesion is >4-6 cm adrenalectomy should be performed. There is no clear consensus in literature about the cut-off values for adrenalectomy, therefore a fuzzy region between 4 and 6 cm is used where the clinician makes the final decision. When a patient of 40 years or more, has hypertension and a hormonally hyperfunctioning lesion adrenal vein sampling has to be performed before adrenalectomy. If the adrenal lesion is not hormonally active a non-contrast CT is performed in 3-6 months after the AI finding. Note that all remaining imaging follow-ups contain escalation handles that interrupt the ordinary work-up process when a case becomes clinically relevant due to growth or malignant characteristics of the adrenal mass. The primary goal of CT imaging in the work-up of incidentalomas is to distinguish between adrenal adenoma, carcinoma, pheochromocytoma, and metastatic lesions. Note that when a relevant diagnosis is made, the patient exits AI work-up. The diagnosis of a benign adenoma depends on the identification of lipid in the adrenal lesion, which can either be assessed on non-contrast CT or in-phase and out-of-phase magnetic resonance imaging (MRI). At Erasmus MC a non-contrast CT was used routinely. If the lesion has benign characteristics, i.e. homogeneous, regular borders and HU<10 on non-contrast CT, then it receives follow-up in the form of repeated CT scans. It should be re-imaged annually for 2 years to check if the lesion grows. If either growth of >1cm in one year or a malignant histology of the lesion are observed then the follow-up process is escalated and adrenalectomy is performed.
Figure 2.2: Erasmus MC daily practice
Chapter 3

Research Methods for Predictive Modeling

This chapter first outlines the research methods for predictive modeling according to the CRISP-DM methodology in slightly more detail. Methods used for data collection, data preparation, data transformation and data integration are discussed more extensively. In the end of the chapter methods for feature selection and the class imbalance problem are explained.

3.1 Methodology for Predictive Modeling: CRISP-DM

The CRISP-DM methodology can be considered a hierarchical process model, consisting of four levels of abstraction from general to specific (see figure 3.1):

![CRISP-DM hierarchical breakdown](image)

*Figure 3.1: CRISP-DM hierarchical breakdown*
**Phases**: The 6 six phases of the CRISP-DM methodology.

**Generic Tasks**: Every phase consists of a set of generic tasks. These are general enough to cover every data mining situation.

**Specialized Tasks**: Describes how actions in the generic tasks should be carried out in specific situations. (i.e. from generic to specific)

**Process Instances**: A record of the actions, decisions and results of a data mining action.

It can be observed that there is a distinct mapping in the middle from generic to specialized. This mapping is driven by the data mining context[19]. The framework in figure 3.2 shows this mapping from generic to specialized. To make sure that there is a clear distinction between methods and results a separate chapter 4 about data exploration and missing value analysis was created. Feature selection (section 3.6) and the class imbalance problem (section 3.6) play a prominent role in the modeling process. Note that the CRISP-DM methodology is used only for predictive modeling, therefore the predictive models are deployed in the form of a prescriptive model. Chapter 7 describes the steps towards a prescriptive model.
<table>
<thead>
<tr>
<th>Generalized Task</th>
<th>Specialized Task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Phases</strong></td>
<td><strong>Figure 3.2:</strong> CRISP-DM mapping from generalized to specialized tasks</td>
</tr>
<tr>
<td><strong>Business Understanding</strong></td>
<td><strong>Data Understanding</strong></td>
</tr>
<tr>
<td>Problem Statement (Ch 1)</td>
<td>Data Collection (Ch 3)</td>
</tr>
<tr>
<td>Need for patient-specific management of adrenal incidentalomas.</td>
<td>Patient Identification: Natural Language Processing</td>
</tr>
<tr>
<td><strong>Generalized Task</strong></td>
<td><strong>Specialized Task</strong></td>
</tr>
<tr>
<td>Research Objective (Ch 1)</td>
<td>Data Exploration (Ch 4)</td>
</tr>
<tr>
<td>Develop prediction models</td>
<td>Univariate analysis: DESCRIPTIVES, box plots, variable distributions</td>
</tr>
<tr>
<td>Management of AI (Ch 2)</td>
<td>Data Quality Assessment (Ch 4)</td>
</tr>
<tr>
<td>AACE/AAES International guideline Erasmus MC daily practice</td>
<td>Missing value analysis Data correctness</td>
</tr>
<tr>
<td>Data Reduction (Ch 3)</td>
<td>Feature selection: Lasso logistic regression, random forest</td>
</tr>
<tr>
<td>Class Imbalance Problem (Ch 3)</td>
<td>Over-sampling: synthetic minority over-sampling technique (SMOTE)</td>
</tr>
</tbody>
</table>
3.2 Data Collection

To answer the research questions and carry out this project that uses data mining it is mandatory that there is sufficient data available. Head of the radiology department prof. G.P. Krestin approved the use of all data concerning patients with (potential) adrenal incidentalomas. The time horizon reaches from 01-01-2000 till 01-07-2014. First the patient identification using natural language processing process will be described then the data acquisition requests for these patients are presented.

3.2.1 Patient Identification Using Natural Language Processing

In order to identify which patients should be included for this project natural language processing (NLP) was used. Using NLP for cohort building of a retrospective study is a common application of data mining techniques[28]. Patients with potential adrenal incidentalomas were automatically identified by applying NLP to unstructured radiology reports. This was done using a NLP pipeline as proposed by E. Pons[28]. For his next publication application of the NLP pipeline for identification of incidental adrenal mass findings will be used as a use-case.

Automated classification of radiology reports using natural language processing and machine learning techniques can effectively facilitate the identification of suitable cases for retrospective research in radiology[29]. Similar to the work of Danforth and colleagues first the reports were filtered on procedural codes then a machine learning algorithm for classification was used[30]. The following sequential steps were used for filtering the reports:

1. Relevant protocol codes matching all clinical scenarios of occurrence are selected.

2. Reports that contain key sentences are selected; key sentences mention the essential anatomical descriptor for the finding (e.g., “adrenal”).

3. Non-relevant anatomic identifiers are removed from key sentences, frequent negative patterns are subsequently filtered (e.g., “normal appearance of [ANAT_LOC], adrenals and [ANAT_LOC]”).

4. A random sample of remaining key sentences are labeled by radiologists via consensus (e.g., “potential incidentaloma”).

5. A machine learning selection algorithm is trained on these examples and applied as final filter to identify potential AI cases.
Figure 3.3 illustrates the steps involved in the process pipeline from total number of CT’s to a subset of all potential adrenal incidentaloma cases.

![Figure 3.3: Automated patient identification pipeline](image)

The reports were acquired from a legacy database called TOPIC. The TOPIC database dump that was made available contained all radiology reports for the period 2010 till 2012. This made it possible to identify all patients with adrenal incidentalomas that were found or received follow-up in this period. A total of 528,127 CT’s were performed. From this total population a relevant subset of 32,298 was selected based on the protocol code enclosed in the report. These protocol codes are: 
- CT ABD (CT Abdomen)
- CT ABD EL (CT Abdomen)
- CT H-TH-ABD (CT Thorax-Abdomen)
- CT TH-ABD (CT Thorax-Abdomen)
- CTA ABD (CT Abdomen)

The second and third step applied simple rule-based filters to select relevant sentences and remove non-relevant atomic objects. This use of positive and keywords has been successfully applied in previous research[30]. The most frequently found patterns can be found in Table 3.1.

The anatomic locations in the sentences are not relevant for identifying potential incidentalomas. Most of the frequently found patterns consist of terms that indicate the adrenal being in good condition. Examples of terms used in sentences that should not be included:

- normaal aspect bijnieren
- normale bijnieren
- slanke bijnieren
- normaal aspect en bijnieren
- normale en bijnieren
- aan bijnieren en geen bijzonderheden
- een afwijkingen bijnieren
- normaal beeld bijnieren
- normaal beeld en bijnieren

So the sentences containing these terms are filtered out. An annotation corpus for learning was constructed with sentences related to *bijnier* for 500 randomized cases in step 4. These sentences were annotated by a medical student and checked by radiologist J.J. Visser. In step 5
a machine learning algorithm was used to classify a report as a potential adrenal incidentaloma (positive) or not (negative). Sentence-level classification was chosen instead of a bag-of-words model for the whole document because most of the time just a few sentences describe the adrenal. This makes the rest of the report obsolete. The concept hierarchy for the adrenal incidentaloma class can be found in Table 3.2.

The subclass Indistinct refers to an adrenal mass that’s described in a vague way. E.g., based on the report, it can’t be classified as an adrenal incidentaloma (mass > 1cm) but instead the radiologist calls it “plomp” (cumbersome). This implies that the adrenal shows abnormalities
but the mass is not large enough to be an adrenal incidentaloma. The model features were N-grams where one feature consists of N sequential words. A total of three classifier were tested on this dataset: Naive Bayes, RIPPER and Support Vector Machine (SVM). The Support Vector Machine (SVM) was selected for classification since it yielded the best performance using 10-fold cross-validation. The evaluation metrics used were sensitivity and specificity. The SVM classifier was used to predict the main class of a report based on the aggregated decision probabilities of individual sentences. The classifier identified 3,516 reports with potential incidentalomas.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.994</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.818</td>
</tr>
</tbody>
</table>

Table 3.3: SVM classifier performance for patient identification

The performance metrics in Table 3.3 indicate that 99.4% of all potential AI cases were correctly classified. It can be concluded from the performance metrics that the classifier was very accurate. At this time the dataset contained all potential adrenal incidentaloma (AI) cases. The reason they are called potential is that their AI class remains to be checked manually later during diagnostic work-up data collection.

3.2.2 Data Acquisition

For all patients potentially having adrenal incidentalomas the following data sources were requested:

**Laboratory Test Data**: Provided by the business intelligence department of Erasmus MC. Includes all measurements, routine lab tests, and specific lab test ever performed in the history of the patient.

**Patient Data**: Provided by the business intelligence department of Erasmus MC. Includes personal patient data including date of birth, date of death (if applicable), gender and geographical data.

**International Classification of Diseases (ICD) Data**: Provided by the business intelligence department of Erasmus MC. Includes all medical treatment that has been invoiced to health insurance companies.

**Medication Data**: Provided by the pharmacy department of Erasmus MC. This data includes all medication that a patient received.
**Measurements Data**: Provided by the business intelligence department of Erasmus MC. Contains all routine measurements including blood pressure, weight, length and pulse.

**Diagnostic Work-up Data**: This data had to be collected manually from the Electronic Health Record (EHR) named Elpado. Elpado’s implementation started in 2008. Data migration from the radiology department was finished in 2009. Work-up data collection was done by Romy de Haan and Job Visser.

The process started with verifying the class of the radiology report which was classified by the NLP algorithm \texttt{nlp\_class}. This class was either finding, indistinct or metastasis. Verification can prove the report to be a finding or follow-up report. Therefore, first the finding report had to be searched for in the radiology report history. If the finding report couldn’t be found we had to check the folder containing examination performed externally. If the AI was found in another hospital the finding class was external (EXT) and data collection stopped. External cases were excluded from the project because we don’t have full transparency in the work-up performed elsewhere. Needless to say, if the finding report couldn’t be found in this folder data collection also stopped. If the finding report was found in the reporting history the finding class \texttt{find\_class} had to be determined. If a patient came to Erasmus MC with complaints related to the abdomen \texttt{find\_class} is negative (NEG) since it is not an incidental finding and therefore not an AI. If the adrenal mass was found but described in a vague way, like for example ‘cumbersome’, then the \texttt{find\_class} is indistinct (IND). The radiologist didn’t consider the mass big enough or didn’t consider the AI clinically relevant. IND, NEG and EXT cases are not included in this research and therefore no further data was collected.

If the \texttt{find\_class} is positive (POS) and the finding date is not before the year 2000, data collection proceeded. All variables collected can be found in Table 3.5. First variables from the finding report were collected, after that the reporting history was inspected in search of follow-up reports. If a follow-up report was identified which was performed for the AI then its variables were collected. After that the patient letters were screened to extract variables related to the patient history. In the end the pathology reports were looked at to see if adrenalectomy or a biopt was performed (most of the time this was already clear from reading the radiology reports). If this was the case, then diagnosis could be verified too. The workflow of diagnostic work-up data can be found in figure 3.4.
Problems Encountered During Data Collection

The following problems were encountered during data collection:

- Only the radiologic reports originating from the period 2010 till 2012 were used in this research. The reason is that EHR data had to be sufficiently complete for retrieval of patient history and reports before and after the finding. EHR implementation was successful in 2009, that’s why the time horizon 2010 till 2012 was chosen. The EHR, Elpado, does contain patient information and reports pre 2009, but these could only be accessed one patient at a time.

- Patient letters are unstructured text files that file the patient history to inform other clinical practitioners about the patient. Text mining could be a valuable tool to extract variables from these letters (e.g. hypertension, hypokalemia and history of malignancy). Unfortunately these couldn’t be extracted through query-based retrieval by the business intelligence department.
3.2.3 Data Description

This section describes the raw data sources that was acquired during data collection. All data, except for the work-up, was provided in a comma-separated value (csv) format. Table 3.4 shows the metadata per data source. These sources contain all data for every requested patients ever recorded. The reports data source was used for the automated case identification.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Description</th>
<th>Record/row</th>
<th>From</th>
<th>Till</th>
<th>No. Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>pat-data</td>
<td>Patient data</td>
<td>patient</td>
<td>-</td>
<td>-</td>
<td>2,122</td>
</tr>
<tr>
<td>icd9</td>
<td>ICD-9 codes</td>
<td>treatment</td>
<td>Jan-00</td>
<td>Jun-13</td>
<td>35,713</td>
</tr>
<tr>
<td>icd10</td>
<td>ICD-10 codes</td>
<td>treatment</td>
<td>Jan-14</td>
<td>Jul-15</td>
<td>5,703</td>
</tr>
<tr>
<td>dbc</td>
<td>DBC codes</td>
<td>treatment</td>
<td>Mar-00</td>
<td>Aug-15</td>
<td>10,341</td>
</tr>
<tr>
<td>measurements</td>
<td>Routine measurements</td>
<td>measurement</td>
<td>Feb-02</td>
<td>Aug-15</td>
<td>324,341</td>
</tr>
<tr>
<td>blood-type</td>
<td>Blood type + rhesus</td>
<td>type or rhesus</td>
<td>Jan-00</td>
<td>Jun-15</td>
<td>17,102</td>
</tr>
<tr>
<td>medication</td>
<td>Medication</td>
<td>medication</td>
<td>Jan-04</td>
<td>Dec-14</td>
<td>201,129</td>
</tr>
<tr>
<td>labzis00 - labzis14</td>
<td>Laboratory tests</td>
<td>test</td>
<td>Jan-00</td>
<td>Dec-14</td>
<td>1,743,920</td>
</tr>
<tr>
<td>work-up</td>
<td>AI work-up</td>
<td>patient</td>
<td>-</td>
<td>-</td>
<td>2,122</td>
</tr>
<tr>
<td>reports</td>
<td>Radiology reports</td>
<td>report</td>
<td>Jan-10</td>
<td>Dec-12</td>
<td>3,516</td>
</tr>
</tbody>
</table>

Patient Data

Personal data was requested from EHR for every patient identified as a potential AI case. Figure 3.5 shows which variables the dataset contains. We are especially interested in the variables: gender, date of birth and date of death.

<table>
<thead>
<tr>
<th>1 pid</th>
<th>2 gender</th>
<th>3 dob</th>
<th>4 mar_stat</th>
<th>5 mult_births</th>
<th>6 dead</th>
<th>7 dod</th>
<th>8 street</th>
<th>9 no</th>
<th>10 no_supp</th>
<th>11 postal_code</th>
<th>12 city</th>
</tr>
</thead>
<tbody>
<tr>
<td>'F'</td>
<td>15-Dec-1929 00:00</td>
<td>'F'</td>
<td>'N'</td>
<td>'J'</td>
<td>15-Nov-2010 00:00</td>
<td>39''</td>
<td>NaT</td>
<td>25''</td>
<td>NaT</td>
<td>25''</td>
<td></td>
</tr>
<tr>
<td>'M'</td>
<td>05-Mar-1974 00:00</td>
<td>'M'</td>
<td>'N'</td>
<td>'J'</td>
<td>06-Mar-1974 00:00</td>
<td>60''</td>
<td>NaT</td>
<td>25''</td>
<td>NaT</td>
<td>25''</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.5: Example of the patient dataset

ICD Data

The international statistical classification of diseases and related health problems (ICD) is a medical classification list by the world health organization (WHO). It contains codes for diseases,
signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or diseases. It is used to monitor the incidence and prevalence of diseases and other health problems. Besides that ICD is used for reimbursement and resource allocation decision making by countries. The dataset contains ICD-9 and ICD-10 codes belonging to the 9th and 10th revision. Figure 3.6 shows which variables the dataset contains. Using these codes it is possible to find out which patient was diagnosed with which disease.

Measurements Data

The measurements dataset contains all routine measurements for 830 unique patients. Figure 3.7 shows which variables the dataset contains. The type variable has six distinct values: adp (arterial diastolic blood pressure), asp (arterial systolic blood pressure), length, weight, puls and temp (temperature).

Blood Type Data

This data contains all the blood types and rhesus factors for 1682 unique patients including the date at which the test was performed.
Medication Data

The medication data is a record of all medication ever subscribed to a patient at Erasmus MC. Figure 3.9 shows which variables the dataset contains. A total of 3210 unique descriptions were found. The medication data can be used to find out if a patient has a certain disease of interest. (e.g. anti-hypertensive drugs could indicate that a patient suffers from hypertension)

<table>
<thead>
<tr>
<th>3 pid</th>
<th>2 start_date</th>
<th>3 step_date</th>
<th>4 type</th>
<th>5 descr</th>
<th>6 dose</th>
<th>7 freq</th>
<th>8 per_dose</th>
<th>9 unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>05-Jan-2004 12:08...08-Jan-2004 15:56...’Klinisch’</td>
<td>MITOTAANTABL500MG</td>
<td>500</td>
<td>’0.75’</td>
<td>’500’</td>
<td>’mg’</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-Jan-2004 15:33...21-Jan-2004 15:39...’Klinisch’</td>
<td>ENALAPRILWATERSTOFMALEAATTABL5MG</td>
<td>5</td>
<td>’0.75’</td>
<td>’5’</td>
<td>’mg’</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-Jan-2004 16:56...22-Jan-2004 15:28...’Klinisch’</td>
<td>INSULINESOFARINJJE100IE ML</td>
<td>100</td>
<td>’NULL’</td>
<td>’ogr gluco...’</td>
<td>’IE’</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28-Jan-2004 01:14...03-Feb-2004 13:00...’Klinisch’</td>
<td>VITAMINBCOMPLEXDRAGEE</td>
<td>0</td>
<td>’0.25’</td>
<td>’1’</td>
<td>’stk’</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.9: Example of the medication dataset

Laboratory Data

The lab test data of 1968 unique patients consisted of 1180 unique types of tests. Figure 3.10 shows which variables the dataset contains.

<table>
<thead>
<tr>
<th>1 pid</th>
<th>2 material</th>
<th>3 date</th>
<th>4 val</th>
<th>5 unit</th>
<th>6 test</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL Bloed</td>
<td>10–May–2006 00...0.600000...’I/l’</td>
<td>'HTHematocrit'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BL Bloed</td>
<td>10–May–2006 00...1.150000...’mmol/l’</td>
<td>'ICACeion Ca'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BL Bloed</td>
<td>10–May–2006 00...1.160000...’mmol/l’</td>
<td>'CA74CCa__7.4'</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.10: Example of the lab test dataset

Work-up Data

The variables collected during work-up data collection including their descriptions can be found in Table 3.5. Please note that some variables like discrepancy in reporting timeline rep_discr are meant for quality assessment. These aid the descriptive analysis of AI work-up, not the dataset for the prediction model. Variable growth gr indicates if a decrease of increase in size of the lesion was mentioned. Growth is not considered clinically relevant until it is $\geq 1$ cm in a time-frame of one year, i.e. significant growth gr_sign.
<table>
<thead>
<tr>
<th>Variable Group</th>
<th>Name</th>
<th>Dataset Name</th>
<th>Values/Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finding Report</td>
<td>AI Finding Class</td>
<td>find_class</td>
<td>[POS, META, IND, NEG, EXT]</td>
<td>Nominal - Adrenal incidentaloma class label at the moment of finding.</td>
</tr>
<tr>
<td></td>
<td>Referring Department</td>
<td>find_refdept</td>
<td></td>
<td>Nominal - Applicant that requested the imaging test.</td>
</tr>
<tr>
<td></td>
<td>Date of Finding</td>
<td>find_date</td>
<td></td>
<td>Datetime - Date of the adrenal incidentaloma finding.</td>
</tr>
<tr>
<td></td>
<td>Type</td>
<td>find_type</td>
<td>[CT, ECHO, MRI, PET]</td>
<td>Nominal - Type of imaging test the patient received.</td>
</tr>
<tr>
<td></td>
<td>Advise for Follow-Up</td>
<td>advise</td>
<td>[0, 1]</td>
<td>Nominal - Advise for follow-up (FU) of the adrenal is given.</td>
</tr>
<tr>
<td></td>
<td>Finding in Conclusion</td>
<td>find_concl</td>
<td>[0, 1]</td>
<td>Nominal - The finding was mentioned in the conclusion.</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>loc</td>
<td>[UNI, BI]</td>
<td>Continuous - Diameter in its longest axis of the largest adrenal mass.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nominal - Location of the adrenal mass(es) found.</td>
</tr>
<tr>
<td>Work-up</td>
<td>Previous Reporting</td>
<td>prev_rep</td>
<td>[0, 1]</td>
<td>Nominal - There was previous reporting on the adrenal mass.</td>
</tr>
<tr>
<td></td>
<td>Unrelated Follow-up</td>
<td>fu_unrel</td>
<td>[0, 1]</td>
<td>Nominal - Subsequent nonrelated reports evaluate the adrenal mass.</td>
</tr>
<tr>
<td></td>
<td>Growth</td>
<td>gr</td>
<td>[0, 1, 2]</td>
<td>Nominal - Indicates if a change in size was mentioned in one of the reports.</td>
</tr>
<tr>
<td></td>
<td>FU Reffering Department</td>
<td>fu_refdept</td>
<td></td>
<td>Nominal - Applicant that requested the follow-up treatment.</td>
</tr>
<tr>
<td></td>
<td>FU Date</td>
<td>fu_date</td>
<td></td>
<td>Datetime - The date of the follow-up treatment.</td>
</tr>
<tr>
<td></td>
<td>FU Type</td>
<td>fu_type</td>
<td>[CT, MRI, BIOPT]</td>
<td>Nominal - The type of treatment the patient received.</td>
</tr>
<tr>
<td></td>
<td>Discrepancy in Reports</td>
<td>rep_discr</td>
<td>[0, 1, 2, 3, 4, 5]</td>
<td>Nominal - Type of discrepancy in the radiology report history.</td>
</tr>
<tr>
<td>Patient Letters</td>
<td>Hypertension</td>
<td>hypertension</td>
<td>[0, 1]</td>
<td>Nominal - Hypertension at some point in the patients history.</td>
</tr>
<tr>
<td></td>
<td>History of Malignancy</td>
<td>hist_malign</td>
<td>[0, 1]</td>
<td>Nominal - Patient has a history of malignancy.</td>
</tr>
<tr>
<td>Clinically Relevant</td>
<td>Diagnosis</td>
<td>diag</td>
<td>[CS, SCS, PA, FEO, META, ACC, OTHER]</td>
<td>Nominal - Which relevant diagnosis the patient received.</td>
</tr>
<tr>
<td></td>
<td>Adrenalectomy</td>
<td>adrenalectomy</td>
<td>[0, 1]</td>
<td>Nominal - Indicates if adrenalectomy for performed.</td>
</tr>
<tr>
<td></td>
<td>Significant Growth</td>
<td>gr_sign</td>
<td>[0, 1]</td>
<td>Nominal - Growth of more than 1 cm in one year since the finding.</td>
</tr>
</tbody>
</table>

Table 3.5: Work-up variables
3.3 Data Cleaning

This step includes data cleaning, missing values imputation and outlier analysis. The cleaning process involves all steps taken in order for the software package MATLAB to be able to perform analysis on the data. The following cleaning steps were performed on the datasets:

**Medication data** Transform medication dose from string to double (numeric).

**Measurement data** Transform separate fields date and time to the datetime data type and concatenate them into one date variable.

**Measurement data** Recode the description variable into a nominal variable.

**Measurement data** Delete all measurements that have a missing measurement value.

**Lab data** Transform the value variable from string to double. This involved replacing comma’s with periods. Furthermore some values contained other characters that have no numerical meaning e.g. < or >. These were removed from the expression after knowing that the tests probably have no value for the project.

**Lab data** Recode the test variable into a nominal variable.

**Work-up data** Recode one case where the lesion location loc was ‘Bi’ instead of ’BI’.

3.3.1 Missing Value Imputation

For missing value imputation the KNNimpute algorithm was used[31]. This algorithm has been successfully applied for estimation of missing values in DNA micro-arrays. Troyanskaya and colleagues showed that KNNimpute provides a more robust and sensitive method for missing value estimation than SVDimpute, and both SVDimpute and KNNimpute surpass the commonly used row average method (as well as filling missing values with zeros).

The KNN imputation method selects patients with a similar input vector as the patient of interest to impute missing values. If a patient $P_i$ with input vector $X_i$ has one missing value for size, the algorithm would find $K$ other patients $P_k$ where $k = 1, \ldots, K$, which have a value present for size. These size values of $K$ patients with input vectors $X_k$ most similar to $X_i$ are used to calculate the size of patient $P_i$. This calculation is a weighted average where the contribution of each patient is weighted by similarity of its input vector $X_k$ to $X_i$.

Previous studies indicate that tumor size is a strong predictor for malignancy[16][14]. Therefore for configuration of the KNNimpute algorithm the variable size was used, it has 54% missing values. After experimenting with all different types of distance measures and values of $K$, the euclidean distance with $K=6$ delivered best results (see figure 3.11).
This setting differs slightly from the findings of the original authors of the algorithms. They concluded that the euclidean distance was sufficiently accurate using $K=5$[31].

### 3.3.2 Addressing the Outlier Problem

Outliers are defined as values at least 3 times the inter quartile range (IQR) above the third quartile or at least 3 times the IQR below the first quartile[32]. In this work we consider outliers as any values that potentially have a very large influence in a classification model. When dealing with outliers it has to be considered if the value is realistic. Questions like this often require expert knowledge. For example a systolic blood pressure of 250 mmHg is biologically plausible in the acute care situation for traumatic brain injury patients, but may not be plausible in an ambulatory care situation. Based on the fact that the data we received was used for final judgment in clinical practice it is highly unlikely that they include data entry errors.

There were some extremely large values for size. For biologically possible values, various statistical approaches are subsequently possible. To reduce the influence on the regression coefficients, we may consider to transform the variable $X$ by "truncation". Very high and very low values are shifted to truncation points[33]:

$$\begin{align*}
&IF \ X > X_{\text{max}} \ THEN \ X = X_{\text{max}} \\
&ELSE IF \ X < X_{\text{min}} \ THEN \ X = X_{\text{min}} \\
&ELSE \ X = X
\end{align*}$$

This technique potentially has a big influence on the prediction models, since they need to detect the few clinically relevant cases. By truncating the variable size the performance might decrease. Therefore we should be wary before applying it. After inspection of the outliers for
variable size it was chosen to not perform truncation for the time being because they are biologically possible.

3.4 Data Transformation

The task of data transformation refers to transforming and consolidating data into forms appropriate for mining. Firstly the process of unstacking the datasets is elaborated upon. Secondly normalization of the variables is discussed and in the end all constructed features are listed.

3.4.1 Unstacking the Datasets

In order to integrate the datasets from different sources they first need to be in the same format. It was decided that a patient per row format is most convenient for this project. Therefore datasets like the lab test data (test per row) need to be unstacked. This means that data is unstacked from a single variable into multiple variables. Another interpretation could be from a tall form (test per row with many rows) into a wide form (patient per row with many columns or variables).

If for example we would like to unstack the biochemical test data called labEND with variables pid, material, date, val, unit and test. The variable to unstack is val or the value of the test where test is the indicator variable in tall that determines which variable in wide each test is unstacked into. The unstack function treats the remaining variables in tall as grouping variables. Each unique combination of their values defines a group of observations in tall that will be unstacked into a single observation in wide. This means that for every patient, for every unique date on which tests were performed, for every test the values are unstacked into a single value in wide. This is done using an aggregation function that takes the mean of the test values for that date. This mean is later on used for feature construction.

3.4.2 Normalization

The unit of measurement of a variable can affect the data analysis. In general expressing a variable in smaller units will lead to a larger range of values and thus tends to give such a variable greater effect or 'weight'[26]. For this reason normalization techniques can be applied. Normalization involves transforming the data to fall within a smaller or common range such as [0,1].

In this project min-max normalization is used, which performs a linear transformation on the original data. Suppose that variable $A$ represents the metanephrine levels where $min_A$ and
min_A are the minimum and maximum values. Min-max normalization maps a value, \( v_i \) of \( A \) to \( v'_i \) in the range \([\min_A^{\text{new}}, \min_A^{\text{new}}]\) by computing:

\[
v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\max_A^{\text{new}} - \min_A^{\text{new}}) + \min_A^{\text{new}}
\]

(3.2)

### 3.4.3 Feature Construction

Below the features can be found that were constructed using the dataset variables. Code used for feature construction can be found in Appendix 9.6.

**Age**  The numeric (integer) variable \( \text{age} \) is constructed by subtracting the date of birth \( \text{dob} \) from the date of finding \( \text{find_date} \). The finding date was extracted from the radiology report for all positive (POS) AI cases, for all other classes in \( \text{find_class} \) the date 01-01-2011 was used. This is the date halfway the time horizon 2010 – 2012 that was used for identification of AI cases.

**Number of days dead after finding**  The numeric variable \( \text{find_dead} \) is only calculated when there is a date of death \( \text{dod} \) available. It is calculated by subtracting \( \text{find_date} \) from \( \text{dod} \).

**Dead in six months**  The nominal variable \( \text{death}_{\text{6mon}} \) indicates if a patient died within a period of six months (182 days) after the AI finding.

**Visit to the endocrinologist**  The nominal variable \( \text{endo} \) indicates if a patient visited the endocrinologist, i.e. if a patient had biochemical screening. These two terms are used interchangeably in this thesis, since a patient that was send to the endocrinologist because of AI should always receive biochemical screening. For every patient it was checked if test values for the following lab tests were present:

- \( \text{metanephrine} \): old or new test.
- \( \text{normetanephrine} \): old or new test.
- \( \text{metanephrine-to-creatinine ratio} \): old or new test.
- \( \text{normetanephrine-to-creatinine ratio} \): old or new test.

These tests are used for diagnosing pheochromocytoma, which has the highest priority.

**Aldosterone**  The nominal variable \( \text{aldos} \) indicates if a patient’s aldosterone was tested. Aldosterone for the diagnosis of primary aldosteronism is tested only if the patient has hypertension, therefore it was not used for constructing \( \text{endo} \).
24-hour total urinary normetanephrines and metanephrines The nominal variable nor_meta_24 checks if either one of these tests were performed. 24-hour total urinary normetanephrine and metanephrine levels weren’t consistently tested and therefore not used for constructing endo.

Date of the biochemical screening Variable date_endo uses the date on which the metanephrines were tested after the finding date. It was agreed upon with endocrinologist dr. R. Feelders that this date is representative for a visit to the endocrinologist, since metanephrine levels should always be checked. Note that this date is only extracted for patients that received biochemical evaluation.

Interval of biochemical screening Variable endo_int gives the number of days between find_date and endo_date.

Interval of imaging follow-up 1 Variable fu1_int gives the number of days between date_endo and fu1_date.

Interval of imaging follow-up 2 Variable fu2_int gives the number of days between fu1_date and fu2_date.

Interval of imaging follow-up 3 Variable fu3_int gives the number of days between fu2_date and fu3_date.

Anti-hypertensive drugs used during finding Nominal variable hyp_med indicates if the patient uses any hypertension medications at the time of AI finding. For a list of 141 unique anti-hypertensive drugs we tested if the find_date was in between the medication start_date and stop_date. This resulted in 61 unique drugs that were used by patients at the moment of finding. The list can be found in Appendix 9.3.

Clinically relevant Nominal variable clin_rel is the target variable for the prediction models. For a patient to be a clinically relevant case it either: (1) receives a relevant diagnosis, (2) has a growing lesion (> 1 cm in one year time) or (3) undergoes adrenalectomy for the AI. The reasoning behind this variable is elaborated upon in section 1.2.

Biochemical screening lab tests In order to use the information from the endocrine lab test data it has to be unstacked. For every patient where endo == 1 the following features were constructed based on the lab test values performed after the AI finding:

- Mean (mean)
• Standard deviation (\textit{std})
• Minimum (\textit{min})
• Maximum (\textit{max})

**Routine lab tests**  Following the same logic, the following feature were constructed based on the routine lab test values:

• Mean (\textit{mean})
• Standard deviation (\textit{std})
• Minimum (\textit{min})
• Maximum (\textit{max})

### 3.5 Data Integration

Now the final datasets are derived by merging the data from all different sources. Since we already unstacked all datasets into a patient per row format we are now able to merge or ‘join’ them together. Therefore the \texttt{join} function is used in MATLAB which performs an outerjoin where it matched the patient id or \texttt{pid} variable from both datasets.

At this point it should be stated that from the ‘master’ dataset two datasets are generated to feed the prediction models. This is best explained using a simplified timeline of the AI work-up (see figure 3.12). Note that this timeline doesn’t account for patients leaving the work-up, this is deliberately done to focus on where the events take place in time. The prediction models can provide decision support throughout the diagnostic work-up. The models predict the likelihood of being \textit{clinically relevant} for an individual patient so that a decision can be made accordingly.

For every patient additional information is generated for every follow-up that is performed. Using this notion we would like to run the prediction model at the moment of the AI finding \((t=0 \text{ months})\) which is referred to as Moment 1 (M1). The second Moment (M2) is right after the biochemical screening has been performed \((t=1 \text{ month})\). Hence dataset M1 and dataset M2 are constructed.

![Figure 3.12: Adrenal incidentaloma work-up event timeline](image-url)
3.5.1 Dataset M1: Adrenal Incidentaloma Found

Size of the population $N = 643$ patients. Number of patients $\text{clin}_\text{rel} = 34$ (5.3%). Variables included:

- categorical: gender, hyp_med, loc, hist_malign
- numerical: age, size

3.5.2 Dataset M2: Biochemical Screening Done

Size of the population $N = 111$ patients. Number of patients $\text{clin}_\text{rel} = 17$ (15.3%). Variables included:

- categorical: gender, hyp_med, hist_malign, aldos, loc
- numerical: age, size
- biochemical screening lab tests (mean, std, min, max): normetanephrines, normetanephrine-to-creatinine ratio, metanephrines, metanephrine-to-creatinine ratio

Due to the large amount of missing values the following biochemical screening lab tests are excluded: ALD Aldosterone, MN Meta_24u, MN meta_24u, MFKR Meta_Kreat, NMD Normeta_24u, NFKR Norm_Kreat, NMET Normetanefr_. Four patient, one of which was clinically relevant, were removed from dataset M2 because they received the new biochemical lab tests.

3.5.3 Dataset Description

Below all definite variables used in datasets M1 and M2 are tabulated including their description. Please note that for all biochemical screening tests the mean, standard deviation, minimum and maximum are included.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Name</th>
<th>Variable</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 – M2</td>
<td>Age</td>
<td>age</td>
<td></td>
<td>Discrete - Age of the patient in years</td>
</tr>
<tr>
<td>M1 – M2</td>
<td>Gender</td>
<td>gender</td>
<td>[M, F]</td>
<td>Nominal - Location of the adrenal mass(es) found.</td>
</tr>
<tr>
<td>M1 – M2</td>
<td>Diameter</td>
<td>size</td>
<td>mm</td>
<td>Continuous - Diameter in its longest axis of the largest adrenal mass.</td>
</tr>
<tr>
<td>M1 – M2</td>
<td>Location</td>
<td>loc</td>
<td>[UNI, BI]</td>
<td>Nominal - Location of the adrenal mass(es) found.</td>
</tr>
<tr>
<td>M1 – M2</td>
<td>History of Malignancy</td>
<td>hist_malign</td>
<td>[0, 1]</td>
<td>Nominal - Patient has a history of malignancy.</td>
</tr>
<tr>
<td>M1 – M2</td>
<td>Hypertension Medications</td>
<td>hyp_med</td>
<td>[0, 1]</td>
<td>Nominal - Hypertension drugs at the moment of finding.</td>
</tr>
<tr>
<td>M2</td>
<td>Aldosterone</td>
<td>aldos</td>
<td></td>
<td>Nominal - Patient has aldosterone tested</td>
</tr>
<tr>
<td>M2</td>
<td>Metanephrine</td>
<td>MNMetanef_oud_</td>
<td>[0, 1]</td>
<td>Continuous - Normetanephrine level in the urine.</td>
</tr>
<tr>
<td>M2</td>
<td>Metanephrine-to-Creatinine Ratio</td>
<td>MNKRMeta_KR_oud_</td>
<td></td>
<td>Continuous - Ratio of urinary metanephrine and creatinine levels.</td>
</tr>
<tr>
<td>M2</td>
<td>Normetanephrine</td>
<td>NMNormeta_oud_</td>
<td></td>
<td>Continuous - Normetanephrine level in the urine.</td>
</tr>
<tr>
<td>M2</td>
<td>Normetanephrine-to-Creatinine Ratio</td>
<td>NMKRNorm_KR_oud_</td>
<td></td>
<td>Continuous - Ratio of urinary metanephrine to creatinine levels.</td>
</tr>
</tbody>
</table>

Table 3.6: Variable descriptions datasets M1 and M2
3.6 Feature Selection

Feature selection aims at reducing the dimensionality of data, i.e. reducing the number of variables. Two forms can be distinguished: (a) selecting relevant variables (or features) among the set of original ones and (b) ranking variables according to their individual predictive power. This project focusses on constructing and selecting subsets of features that are useful to build a good prediction model (option a). Feature selection also helps to better understand the data by discovering which are the important features and how they are related to each other[34]. Besides that it reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

There are three main directions to take when applying best subset feature selection. They divide into (1) wrappers, (2) filters and (3) embedded methods. The wrapper approach consists of using the prediction performance of the prediction model to assess the relative usefulness of subsets of variables. The filter approach uses measurements as evaluation criteria to evaluate the quality of feature subsets. Filters select subsets of variables as a preprocessing step, independently of the chosen predictor. Embedded methods perform feature selection in the process of training and are usually specific to given learning algorithms[35]. We focus on wrappers and embedded methods.

Wrapper methods use the learning machine of interest as a black-box to score subsets of features according to their predictive power. Depending on the model and performance metrics chosen an exhaustive search can fast become computationally intensive. This problem is known to be NP-hard and the searches become computationally intractable[36]. Wrappers can apply greedy search strategies to overcome the computational burden and at the same time be robust against overfitting. Two types of categories are forward selection and backward selection. Embedded methods incorporate variable selection as part of the training process which may be more efficient because: (1) they don’t need to split the data into a training and testing set and (2) they reach a solution faster by avoiding retraining a predictor from scratch for every variable subset investigated[35]. For this project two types of embedded methods are used: Lasso regularization and the random forest algorithm.

3.6.1 Lasso Regularization

The most common type of embedded feature selection method are regularization techniques. These are often called penalization methods that introduce additional constraints into the optimization of a predictive algorithm, in our case Lasso logistic regression (see subsection 5.1.1).
By applying the $L_1$ constraint both shrinkage and feature selection are done simultaneously. In this project we apply the algorithm for the sole purpose of feature selection since logistic regression usually outperforms Lasso logistic regression on the reduced feature set[32].

3.6.2 Random Forest

Random forests are a combination of tree predictors such that each tree depends on the values of a random subset of variables sampled independently and with the same distribution for all trees in the forest[37]. Please see section 5.1 for a brief explanation on how the algorithm works. A random forest makes it able to estimate the importance of a variable. This property is used for feature selection in this project. Variable importance is estimated in the following way: for every tree $k$, count the number of votes for the correct class. Then permute the values of variable $d$ in the out-of-bag (OOB) cases and run the tree on them. Now take the number of votes for the correct class in the original OOB data and subtract from them the number of correct votes in the variable-$d$-permuted OOB data. The average of this number over all the trees is the raw importance of variable $d$.

3.7 Addressing the Class Imbalance Problem

For this project there exists a so called class imbalance problem which refers to the large group of 607 (94.7%) patients that is not clinically relevant compared to the small group of 34 (5.3%) that is. A well balanced dataset is very important for training a good prediction model. An imbalanced dataset causes the prediction model to lose generalization, i.e. not able to correctly classify new cases as clinically relevant. Most existing classification methods tend to perform poorly on minority class samples when the dataset is extremely imbalanced. This is because they aim to optimize the overall accuracy without considering the relative distribution of each class[38].

There are two common approaches to tackle the problem of extremely imbalanced data. One is based on cost sensitive learning: assigning a high cost to misclassification of the minority class[39]. The other approach is to use a sampling technique: Either down-sampling the majority class or over-sampling the minority class, or both. For this project an over-sampling approach called SMOTE is used which has been applied extensively in the medical field[40][41]. SMOTE stands for Synthetic Minority Over-sampling TEchnique in which the minority class is over-sampled by creating 'synthetic' examples rather than by over-sampling with duplicated real data entries. Depending upon the amount of over-sampling required, neighbours from the $K$ nearest neighbours of a case are randomly chosen. The main drawback is that it can
damage the performance of data mining algorithms due to increased computational load[38]. Because the dataset of this study is small, this is not a problem. Feature selection should be performed before SMOTE is used because it has hardly any effect on most classifiers trained on high-dimensional data[40].
Chapter 4

Data Analysis

This chapter aims for familiarization with the data. Univariate and bivariate analyses are performed to summarize the large and complex datasets both numerically and visually. This conveys the essence of the data and allows for further processing. The exploratory analysis focuses on the variables that are expected to have some value for the descriptive analysis of the diagnostic work-up or the prediction model. Relevant variables included are:

**Numerical**
- Patient data: age, find_dead
- Lab data: Adrenal incidentaloma biochemical screening lab tests (section 2.2), endo_int
- Work-up data: size, fu1_int, fu2_int, fu3_int

**Categorical**
- Patient data: gender, dead
- Lab data: endo
- Work-up data: find_class, find_type, advise, find_concl, loc, prev_img, fu_unrel, gr_sign, hypertension, diag, adrl, hist_malign, death_unrel, clin_rel

After exploration of the data its quality is assessed in terms of correctness followed by a missing value analysis.

### 4.1 Data Exploration: Univariate Analysis

The univariate analysis concerns the descriptive statistics of the variables. For categorical variables a tabulation or cross-tabulation is given. For numeric variables the following descriptive
statistics are calculated: mean, standard deviation, minimum and maximum. Graphs to visually inspect the data are provided in the form of histograms and bar charts.

### 4.1.1 Descriptive Statistics

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXT</td>
<td>370</td>
<td>17.44</td>
</tr>
<tr>
<td>IND</td>
<td>711</td>
<td>33.51</td>
</tr>
<tr>
<td>META</td>
<td>179</td>
<td>8.44</td>
</tr>
<tr>
<td>NEG</td>
<td>219</td>
<td>10.32</td>
</tr>
<tr>
<td>POS</td>
<td>643</td>
<td>30.30</td>
</tr>
</tbody>
</table>

**Table 4.1:** Frequency table of `find_class`

Table 4.1 shows the actual class distribution of the `find_class` variable. This can be interpreted as the actual adrenal incidentaloma class based on the finding report. These labels are assigned to patients identified as potential adrenal incidentalomas. This research focuses on the positive (POS) class only (n=643), therefore all other classes are excluded from the population until further notice.

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>528</td>
<td>82.12</td>
</tr>
<tr>
<td>1</td>
<td>115</td>
<td>17.88</td>
</tr>
</tbody>
</table>

**Table 4.2:** Frequency table of `endo`

<table>
<thead>
<tr>
<th></th>
<th>noAldos</th>
<th>Aldos</th>
</tr>
</thead>
<tbody>
<tr>
<td>noEndo</td>
<td>528</td>
<td>0</td>
</tr>
<tr>
<td>Endo</td>
<td>25</td>
<td>90</td>
</tr>
</tbody>
</table>

**Table 4.3:** Cross-tabulation of `aldos` and `endo`

Table 4.2 shows the percentage of AI cases that received biochemical screening. 17.9% is very low since, according to both the daily practice and international guideline all AI cases should go through biochemical screening. This is the only way to know if a lesion is hormonally active or not. This finding is of concern to the project and should be inspected further. Of these 115
patients 90 received aldosterone screening therefore these patients are suspected to have had hypertension at the moment of finding. (see Table 4.3)

<table>
<thead>
<tr>
<th></th>
<th>noAldos</th>
<th>Aldos</th>
</tr>
</thead>
<tbody>
<tr>
<td>HypMed</td>
<td>8.0</td>
<td>20.0</td>
</tr>
<tr>
<td>noHypMed</td>
<td>17.0</td>
<td>70.0</td>
</tr>
</tbody>
</table>

Table 4.4: Cross-tabulation of hyp_med and aldos for dataset 2

Just 20 patients use hypertension medications at the moment of finding received aldosterone screening. (see table 4.4) This could indicate two things: (1) that hypertension was diagnosed during the visit at the endocrinologist or (2) that the patient was using antihypertensive drugs that were prescribed by another instance than Erasmus MC.

<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.95</td>
</tr>
<tr>
<td>0</td>
<td>97.05</td>
</tr>
</tbody>
</table>

Table 4.5: Frequency table of gr_sign

Table 4.5 shows the number of cases where significant growth was observed. This means that the adrenal mass has grown $\geq 1$ cm in one year.

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>3.00</td>
<td>0.47</td>
</tr>
<tr>
<td>CS</td>
<td>3.00</td>
<td>0.47</td>
</tr>
<tr>
<td>FEO</td>
<td>2.00</td>
<td>0.31</td>
</tr>
<tr>
<td>META</td>
<td>3.00</td>
<td>0.47</td>
</tr>
<tr>
<td>OTHER</td>
<td>3.00</td>
<td>0.47</td>
</tr>
<tr>
<td>PA</td>
<td>1.00</td>
<td>0.16</td>
</tr>
<tr>
<td>SCS</td>
<td>4.00</td>
<td>0.62</td>
</tr>
<tr>
<td>Unknown</td>
<td>624.00</td>
<td>97.05</td>
</tr>
</tbody>
</table>

Table 4.6: Frequency table of diag

Table 4.6 shows the distribution of diagnoses among patients. The relevant diagnosis include: adrenocortical carcinoma (ACC), hypercortisolism (Cushing’s syndrome(CS)), pheochromocy-
toma(FEO), metastatic disease (META), primary aldosteronism (PA) or subclinical Cushing’s syndrome (SCS). The class OTHER refers to rare cases where the diagnosis was unclear or not possible to make. The Unknown class refers to all other cases where the lesion was proven, or strongly suspected, to be benign and non-functioning.

Table 4.7 shows the distribution of clinically relevant clin_rel outcomes among the 643 patients. As discussed earlier clin_rel is the outcome variable for the prediction model. For a patient to be a clinically relevant case it either: (1) carries a relevant diagnosis, (2) has a growing lesion (≥ 1 cm in one year) or (3) undergoes adrenalectomy for the AI. The reasoning behind this variable is elaborated upon in section 1.2.

<table>
<thead>
<tr>
<th>Clinically Relevant</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>diagnosis, growth, adrenalectomy</td>
<td>2</td>
<td>0.31</td>
</tr>
<tr>
<td>diagnosis, adrenalectomy</td>
<td>6</td>
<td>0.94</td>
</tr>
<tr>
<td>diagnosis, growth</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>growth, adrenalectomy</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>diagnosis</td>
<td>5</td>
<td>0.78</td>
</tr>
<tr>
<td>growth</td>
<td>17</td>
<td>2.65</td>
</tr>
<tr>
<td>adrenalectomy</td>
<td>4</td>
<td>0.62</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>34</strong></td>
<td><strong>5.30</strong></td>
</tr>
</tbody>
</table>

Table 4.7: Frequency table of clin_rel

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>507</td>
<td>78.85</td>
</tr>
<tr>
<td>1</td>
<td>136</td>
<td>21.15</td>
</tr>
</tbody>
</table>

Table 4.8: Frequency table of death_6mon

Earlier we found out that just $\frac{115}{643} = 17.9\%$ of all positive AI cases received biochemical screening, which is concerning. Table 4.8 shows that $\frac{136}{643} = 21\%$ of the patients died within 6 months after the AI finding. It can be considered highly unlikely that an AI patient dies because of the adrenal tumor within 6 months. This variable was constructed to get to know why a large fraction of AI patients didn’t receive proper follow-up. A sample of these cases could be investigated to see what their cause of death was to verify if the consideration is correct.

Table 4.9 shows that the average age of a patient at the finding was 62.7 with a reasonable
Table 4.9: Descriptive statistics of continuous variables

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
<th>Mean</th>
<th>Std</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>643</td>
<td>100.0</td>
<td>62.7</td>
<td>11.1</td>
<td>16.0</td>
<td>93.0</td>
</tr>
<tr>
<td>find_dead</td>
<td>210</td>
<td>32.7</td>
<td>144.8</td>
<td>105.6</td>
<td>0.0</td>
<td>362.0</td>
</tr>
<tr>
<td>size</td>
<td>349</td>
<td>54.3</td>
<td>22.8</td>
<td>14.1</td>
<td>10.0</td>
<td>150.0</td>
</tr>
<tr>
<td>endo_int</td>
<td>115</td>
<td>17.9</td>
<td>280.8</td>
<td>660.4</td>
<td>1.0</td>
<td>4110.0</td>
</tr>
<tr>
<td>fu1_int</td>
<td>180</td>
<td>28.0</td>
<td>233.6</td>
<td>421.3</td>
<td>1.0</td>
<td>3343.0</td>
</tr>
<tr>
<td>fu2_int</td>
<td>71</td>
<td>11.0</td>
<td>309.9</td>
<td>208.5</td>
<td>2.0</td>
<td>1002.0</td>
</tr>
<tr>
<td>fu3_int</td>
<td>24</td>
<td>3.7</td>
<td>342.1</td>
<td>293.7</td>
<td>0.0</td>
<td>1306.0</td>
</tr>
</tbody>
</table>

standard deviation of 11. This indicates that the patient population is quite old which is in line with literature which states that the likelihood of AI increases with age[4]. The maximum size of 150 mm is almost tenfold higher than the average. This is physically possible in case of ACC, but might have too much impact on the prediction models. A large fraction of $\frac{210}{643} = 32.7\%$ of the patients died, find_dead states how many days after the AI finding a patient died. endo_int, fu1_int, fu2_int and fu3_int reflect the number of days between subsequent follow-up activities. It can be observed that 180 patients did receive imaging follow-up and just 115 patients underwent biochemical screening. This is again not in line with daily practice and may indicate that the referring department would rather request an extra imaging examination instead of referring the patient to the endocrinologist.

4.1.2 Box Plots

Figure 4.1 indicates that there are two extreme outliers in variable size. These cases were diagnosed with ACC and indeed the tumor was that big. The age variable shows no extraordinary findings except for some younger patients.

The follow-up intervals in figure 4.2 show how many days were in between subsequent follow-up activities. these can be interpreted as follows: (1) from finding to endo, (2) from endo to fu1, (3) from fu1 to fu2 and (4) from fu2 to fu3. It can be observed that there are large fluctuations in the beginning of the work-up. For some of the outliers in the imaging follow-up one might argue that these were performed not solely for the AI but rather mentioned the adrenal together with other indications in the report.

The laboratory test dataset is structured in a test per row format and contains a total of 1180 unique tests. Table 4.10 shows all tests performed during biochemical screening in line with the daily practice presented in section 2.2. The descriptive statistics are based on all test values
Figure 4.1: Box plots of size and age

Figure 4.2: Box plots of endo_int, fu1_int, fu2_int, fu3_int

(Test Count) per lab test. The following things can be observed:

- At some point a new type of test was introduced for measuring the metanephrines, normetanephrines and their creatinine ratios. Inspection of the data indicates that the new tests were introduced in April 2014.
- The normetanephrine-to-creatinine ratio has an exceptionally high maximum value.

What follows is a conversion list for all biochemical screening lab tests performed at Erasmus MC.
Table 4.10: Descriptive statistics of biochemical screening tests

- **ALDSAldosteron** = This is the ratio of plasma aldosterone concentration (PAC) (ng/dL) to plasma renin activity (PRA) for the diagnosis of primary hyperaldosteronism.
- **MNKRMeta_KR_ou_** and **MFKRMeta_Kreat** = metanephrine-to-creatinine ratio for the diagnosis of pheochromocytoma.
- **MNMetanef_ou_** and **MNEMetanefrine** = metanephrine levels for the diagnosis of pheochromocytoma.
- **MNDMeta_24u** = 24-hour total urinary metanephrines for the diagnosis of pheochromocytoma.
- **NFKRNorm_Kreat** and **NMKRNorm_KR_ou_** = normetanephrine-to-creatinine ratio for the diagnosis of pheochromocytoma.
- **NMDNormeta_24u** = 24-hour total urinary normetanephrine levels for the diagnosis of pheochromocytoma.
- **NMNormeta_ou_** and **NMETNormetanefr_** = normetanephrine levels for the diagnosis of pheochromocytoma.

Table 9.15 in Appendix 9.1 shows the descriptive statistics of 'routine' lab tests. These were identified by evaluating the percentage of missing values using a cutoff of 5%. Figure 9.4 shows the correlation heatmap. These tests turned out to have no predictive power.
4.1.3 Distribution of Variables

The size variable clearly is not normally distributed. The red line in figure 4.3 is a fitted lognormal distribution. Some parametric classification algorithms like logistic regression work better when continuous predictor variables are normally distributed. This requires transformation of the size variable which is accounted for in the data transformation phase (section 3.4).

The distribution of age in figure 4.4 shows normal distribution in red which produces a good fit.
4.2 Data Exploration: Bivariate Analysis

Prediction modelers of medical data should assess non-linearity and correlation of continuous variables. Some parameterized data mining techniques require that the correlation among predictor variables is not too high. Therefore a correlation heatmap is constructed. In the end a scatter-plot matrix is made based on the continuous variables grouped by the target variable clin.rel. Based on the missing value analysis in section 4.4 only the variables that sufficiently complete are included.

4.2.1 Correlation Heatmap

![Correlation Heatmap of Continuous Variables](image)

From figure 4.5 we can see that NMNormeta_oud_ and NMKRNorm_KR_oud_ are highly correlated with each other. This is no surprise since the normetanephrine-to-creatinine ratio is directly related to normetanephrine levels in the blood. The same logic to a lesser extend applies to metanephrine levels and ratios. No significant correlations with age and log_size were found. It is important to note that this analysis is done using only for the 115 positive AI cases that received biochemical screening and uses the average of the lab test values per patient. The remaining screening lab tests were excluded because they had more than 90% missing values. (see section 4.3)
4.2.2 Scatter Plots

Figure 4.6 shows that the distribution of age for clinically relevant cases is quite similar to the distribution of non-clinically relevant patients. Note that this visualization uses all 643 positive cases. The clinically relevant patients have a distribution for size which is more flat and shifted to the right.

Figure 4.7 applies to the patient subset of 115 that received biochemical screening. It can be concluded from log_size that clinically relevant patients clearly have bigger lesions in general. Furthermore the younger patients seem to be clinically relevant more often which is remarkable. The outliers that clearly stand out from the rest are the pheochromocytoma cases.

4.3 Verify Data Quality: Correctness of the Data

This section assesses correctness of the data. In order to do this the individual datasets need to be unstacked. This means that every dataset has to be converted to a patient per row format (this is explained in section 3.5) The following issues arose concerning correctness of the data:

**Patient removal** One patient appeared in the work-up dataset which was not present in the
data acquisition patient list. Therefore its data is missing and the patient is removed from the dataset.

**New and old biochemical screening tests** The endocrine lab tests performed during biochemical screening have been renewed in 2014. Since the reference values differ, the values can’t be merged with those of their predecessor.

**Nominal values** All nominal variables in the work-up dataset that have missing values should
be '0' instead of 'NaN'. Since the scope of this project assumes that everything we can extract from EPD is everything there is to know about the patient we can recode the missing values to '0'.

**1-mg dexamethasone suppression test** During the 1-mg dexamethasone suppression test the patient will receive dexamethasone. Afterwards blood is drawn so that the cortisol level can be measured. Cortisol measurements are performed for a wide range of indications. The way to identify if the cortisol test value relates to a 1-mg dexamethasone suppression test is to check the note in EHR. The database export didn’t include the note field and therefore this test can’t be identified. This issue is addressed at indications for future research.

### 4.4 Verify Data Quality: Missing Value Analysis

Variable size was extracted from the finding report. The report is based on the images, but because the size, interpreted from the CT image, is not reported consequently there are 45% missing values.(see figure 9.7) This insight is troublesome since literature indicates that size is a significant predictor for malignancy[16][42].

Figure 4.8: Missing value analysis biochemical screening data

Figure 4.8 shows that, among the 115 patients that received biochemical screening, the new tests have approximately 95% missing values. This is no surprise since a new type of metanephrine
and normetanephrine test was introduced in April 2014. It is surprising that the 24 hour urinary metanephrine and normetanephrine levels have that many missing values too. This indicates that these weren’t part of the regular biochemical screening for a long time which contradicts with what the endocrinologist said.

![Missing value analysis measurements data](image)

**Figure 4.9:** Missing value analysis measurements data

The routine measurements data in figure 4.9 show that variables `puls`, `weight` and `length` can be excluded from further analysis because of the high percentage of missing values. `temp`, `asp` and `adp` could be used for further analysis but their usefulness is doubted. This is because many of these blood pressure measurements originate from a monitoring device recorded during surgery. Therefore it is not possible to for example draw conclusion about hypertension based on this data.
Chapter 5

Predictive Modeling

By now the data has been collected, pre-processed and explored meaning we can start the actual modeling. This chapter first covers the predictive modeling techniques in detail. Secondly the experimental design is discussed building upon the M1 and M2 datasets introduced in section 3.5. Thirdly the metrics are introduced meant for evaluation of the model performance. In the end the actual models are build and interpreted.

5.1 Modeling Techniques

This section documents the data mining techniques used for classification. The selected algorithms are logistics regression, fuzzy inference system and random forest.

5.1.1 Logistic Regression

Logistic regression measures the relationship between the categorical target variable and the predictor variables using a logistic function. The model is flexible in that it can incorporate categorical and continuous predictors, non-linear transformations, and interaction terms. Many of the principles of linear regression also apply for logistic regression. The binary outcome $y$ is linked to a linear combination of a set of predictors $x$ and regression coefficients $\beta$. It can be seen as a generalized linear model just like the linear regression model except for the fact that the conditional distribution is a Bernoulli distribution instead of a Gaussian distribution[43]. This is because the target variable is binary. Besides this the predicted output values are likelihoods in the range $[0, 1]$, where 1 equals *clinically relevant* and 0 equals *not clinically relevant*. To make sure the that the predictions are restricted to this interval a logistic link function is used. Logistic regression predicts the *odds* of being *clinically relevant* based on the
predictor variables $x$. Where odds refers to the conditional probability that a specific outcome is a clinically relevant $P(y = 1|x)$ divided by the conditional probability that it is not clinically relevant $P(y = 0|x)$. More specifically the model is written as a linear function inside a logistic function, or logit. So for clarification:

$$\text{logit}(P(y = 1)) = \log(\text{odds}(y = 1)) = \log \left( \frac{P(y = 1|x)}{P(y = 0|x)} \right)$$

(5.1)

The logit is the logarithm of the odds of the target variable $y$ expressed as a function of predictor variables, $x$, and a constant term:

$$\text{logit}(P(y = 1)) = \beta_0 + \beta x^T$$

(5.2)

Where $\beta_0$ is the intercept and $\beta = (\beta_1, ..., \beta_p)$ denotes the linear coefficients for a total of $p$ variables. The logit gives the odds of a clinically relevant case, however if we want to convert this into likelihoods, we need to use the transformed equation below:

$$P(y = 1) = \frac{1}{1 + e^{-\beta_0 - \beta x^T}}$$

(5.3)

The quantity $e^{-\beta_0 - \beta x^T}$ is the odds of the outcome. The logistic function has a characteristic sigmoid shape, or S shape. Interpretability to a clinical audience is usually essential. Logistic regression models can transparently be presented, with insight in the relative effects of predictors by odds ratios. A maximum likelihood approach is commonly used in calculating the coefficients and the log-likelihood can be written:

$$l(\beta_0, \beta) = \sum_{i=1}^{n} (y_i \log(P(y = 1; \beta)) + (1 - y_i) \log(1 - P(y = 1; \beta)))$$

(5.4)

$$= \sum_{i=1}^{n} \left( y_i (\beta_0 + \beta x_i^T) - \log(1 + e^{\beta_0 + \beta x_i^T}) \right)$$

(5.5)

Logistic regression is used for predictive modeling.

**Lasso Logistic Regression**

The logistic regression model can be extended into a Lasso logistic regression model by imposing a $L_1$ constraint on the $\beta$ coefficients. These are often called penalization methods that bias the model towards less coefficients. The Lasso logistic regression minimizes the negative log-likelihood function with a penalty term:[32]:

$$\min_{\beta_0, \beta} \left( \sum_{i=1}^{n} \log(1 + e^{\beta_0 + \beta x_i^T}) - y_i (\beta_0 + \beta x_i^T) \right) + \lambda \sum_{j=1}^{p} |\beta_j|$$

(5.6)
Here $n$ is the number of patients, $y_i$ the response and $x_i$ the input vector for patient $i$. $\lambda$ is a non-negative regularization parameter corresponding to one value of lambda. As $\lambda$ increases, the number of nonzero components of $\beta$ (vector of $\beta_i$ coefficients for $i = 1, ..., p$) decreases. In other words the effect of non-relevant variables diminishes. In this way, both shrinkage and feature selection are done simultaneously and it is also this property that makes Lasso generally much easy to interpret and a very popular algorithm. Note that the Lasso regularization algorithm can’t handle categorical predictors. Therefore it is used only for feature selection in section 3.6.

5.1.2 Fuzzy Inference System

Fuzzy inference is a method that interprets the values in the input vector and, based on some set of rules, assigns values to the output vector. In fuzzy logic, the truth of any statement becomes a matter of degree. First a short introduction to fuzzy logic is given. Then the fuzzy inference system of our choosing, Adaptive Neuro-Fuzzy Inference Sytem (ANFIS), is explained.

Fuzzy Logic

Fuzzy logic is synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with unsharp boundaries in which membership is a matter of degree. The basic idea of fuzzy logic refers to the concept of a linguistic variable, i.e. a variable whose values are words, not (crisp) numbers. Words are of course less precise then numbers, but their use lies much closer to human intuition. Another basic concept is that of the fuzzy if-then rule, or fuzzy rule, where the IF part is called the antecedent and the THEN part is called the consequent. (e.g. IF temperature is cold THEN heather is high) These can be applied to rule based systems and provide a way for dealing with fuzzy consequents and fuzzy antecedents. Fuzzy logic falls under the branch of soft computing. Soft computing is a collection of methodologies, which aim to exploit tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost.

Fuzzy inference is the process of formulating a mapping from a given input to an output using fuzzy logic. Fuzzy inference system (FIS) here refers to a fuzzy rule based classifier where where fuzzy IF-THEN rules provide a basis for making decisions. A fuzzy inference process consists of five steps[44]:

Fuzzification of the input variables takes the crisp inputs and determines the degree to which they belong to each of the appropriate fuzzy sets via membership functions.
Application of the fuzzy operator (AND or OR) in the antecedent If the antecedent of the rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent of that rule. This number is then applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables and the output a single truth value.

Implication from the antecedent to the consequent The input of the implication process is a single crisp number given by the antecedent

Aggregation of the consequents across the rules Decisions are based on the testing of all of the rules in a FIS. Therefore the rules must be aggregated. This is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set.

Defuzzification The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single crisp number. Fuzziness helps the rule evaluation during the intermediate steps but the final desired output for each variable is generally a single value.

Adaptive Neuro-Fuzzy Inference System

An artificial neural network (ANN) tries to imitate the neurological function of the brain. ANN’s consist of interconnected nodes that have the same function as neurons do in the brain. They are connected through links with adjustable weights. These weights are adjusted as the ANN is being trained. A perceptron is a feed-forward NN where the input moves in one direction and there are no loops in the topology[45]. The most powerful and widely used NN is the multi-layer perceptron with back-propagation which makes backward feedback through the model topology possible by adjusting the weights in order to produce an output that better resembles the original output. Multi-layer refers to the number of hidden layers of nodes in between the input and ouput layer which make it possible to construct nonlinear relationships between input variables.

ANN’s require crisp and precise numeric input data in order to function correctly. Real world data often is not as crisp and precise as we would like it to be. Fuzzy logic offers a flexible way to deal with different aspects of uncertainty or incompleteness in real life situations. In a FIS the features are associated with a degree of membership to different classes. This linguistic transparency provided by fuzzy classifiers is an important argument for using them for this project, since it facilitates communication with domain experts using the membership functions.
Both ANNs and FIS are very adaptable in estimating input – output relationships. Neural networks deal with numeric and quantitative data while fuzzy systems can handle linguistic and qualitative data. By combining both techniques into a neuro-fuzzy hybrid we obtain a system which unites fuzzy reasoning logic with the learning capability of neural networks. The hybrid we’re interested in is an Adaptive neuro-fuzzy inference system (ANFIS) proposed by Jang that utilizes a first order Takagi–Sugeno fuzzy model to produce fuzzy IF-THEN learning rules[44]. Figure 5.1 illustrates the AFNIS architecture. Layer 1 is the fuzzification layer which passes the crisp inputs $x$ and $y$ to determine the membership for each input in the form of a gaussian fuzzy membership function. $\mu_{A_j}(x)$ represents the antecedent membership function which has the form:

$$\mu_{A_j}(x) = \exp \left( - \frac{(x - v_j)^2}{\sigma_j^2} \right)$$ \hspace{1cm} (5.7)

Where parameters $v_j$ and $\sigma_j$ of the antecedent membership function need to be estimated. These parameters indicate, respectively, the center and the width of the membership function. In this work a unsupervised clustering algorithm called fuzzy c-means (FCM) was used for estimating the parameters[46]. In layer 2, the rule layer, the firing strength of the rule is calculated by taking the product of the membership grades. Layer 3 ’normalizes’ the firing strengths. Every node in this layer receives input from all nodes in layer 2 and calculates the ratio of firing strengths of all rules. Layer 4 is called the defuzzification layer which produces the consequent for each rule based on the fuzzy antecedents from layer 3. In layer 5 a single node calculates the overall output of ANFIS by aggregating all incoming defuzzified signals.
The algorithm uses a combination of the least-squares and back-propagation gradient descent methods for training the model. The main benefit of such a hybrid approach is that it converges much faster, since it reduces the search space dimensions of the back-propagation method used in neural networks[44]. In the medical context, fuzzy approaches have been used successfully in many topics, including in the prediction of patients’ survival rate[47], prognosis[48], cancer relapse[49] and mortality in septic shock patients[50].

5.1.3 Random Forest

A random forest grows many decision trees, i.e. they are an ensemble of trees. Therefore first a decision tree is shortly introduced and after that the random forest algorithm, which is an ensemble of decision trees.

One Decision Tree

Tree models where the target variable takes a finite set of values (i.e. clinically relevant or not clinically relevant) are called classification trees. In this work, when we use ‘decision tree’ we actually refer to a ‘classification tree’. The name decision tree arose from the easy to understand visual appearance of recursive partitioning algorithms. This flowchart-like tree structure is constructed in a top-down, recursive, divide-and-conquer, manner[51]. Figure 5.2 shows an example where we want to predict if the day is suitable to go golfing. This prediction is made based on the weather outlook for the specific day. Splitting golf data on the outlook attribute yields three subsets or branches. The leaf nodes visualize the class imbalance for the target variable indicating if the day is suitable to go golfing. The middle and right branches may be split further.

The Attribute Selection Method (ASM) is the key process in the construction of a decision tree. The ASM selects a splitting criterion (attribute) that best splits the given sample into each of the class labels. Decision trees assume that the decision boundaries are parallel to the axes, e.g. if we have two features \((x_1, x_2)\) then it can only create rules such as \(x_1 >= 4.5, x_2 >= 6.5\) etc. which we can visualize as lines parallel to the axis. This is illustrated in figure 5.3.
So decision trees chop up the feature space into rectangles (or in higher dimensions, hyper-rectangles). Decision trees can be interpreted in the form of IF-THEN rules if you interpret the branches and nodes top-down. In other words, constructing a decision tree is the training step of classification[26].

If in some situation two variables offer an equally good purity of sub-samples. For their split the algorithm chooses one randomly. This could have consequences though for the final solution and structure of the decision tree. This specific split may yield a locally optimal solution for this data sample, and not a globally optimal solution for the entire data universe. Thus a tree will be only one of many possible trees that could be built, some of which may (or may not) provide better predictive accuracy. One algorithm that deals with these issues in an efficient manner is a random forest.

**An Ensemble of Decision Trees**

A Random Forest grows many classification trees, i.e. they are an ensemble of decision trees proposed by Breiman[52]. Every tree in the forest classifies if a patient is *clinically relevant* based on input vector $X_D$ where $D$ is the number of variables. We could say that every tree votes for a class $Y$. The forest chooses the classification having the most votes (over all the trees in the forest). This specific implementation is also known as the *TreeBagger* algorithm in MATLAB. *TreeBagger* refers to bootstrap aggregation for an ensemble of decision trees. Each tree is grown as follows:

1. If the number of patients in the training set is $N$, sample $N$ patients at random with replacement, from the original data. The sampled will be used for training the tree.
2. If there are $D$ input variables, a number $d << D$ is specified such that at each node, $d$ variables are selected at random out of the $D$ and the best split on these $d$ is used to split the node. The value of $d$ is held constant during the forest growing.

3. Each tree is grown to the largest extent possible. So there is no pruning.

Every tree $k$ in the forest $K$ is trained using two third of the cases for training and one third for testing. The training set is sampled with replacement. This process is repeated for every tree and is referred to as bootstrapping. The testing set is called out-of-bag (OOB) data and used to get an estimate of the classification error in the process of adding more trees to the forest.

**OOB classification error**  Every OOB case is ran down tree $k$ after training to get a prediction. This way a test set classification is obtained in approximately one third of the trees. At the end of the run, when the forest is constructed take $c$ to be the class that got most votes every time case $n$ was OOB. The OOB classification error is the proportion of times that $c$ is not equal to the actual class of case $n$ averaged over all cases.

After each tree is built, all the data is run down the tree, and proximities are computed for each pair of cases. If two cases occupy the same terminal node, their proximity is increased by one. After the forest has run, the proximities are normalized by dividing by the number of trees. Proximities can also be used for replacing missing data and locating outliers[52].

**Variable importance**  A random forest makes it able to estimate the importance of a variable. For every tree $k$, count the number of votes for the correct class. Then randomly permute the values of variable $d$ in the OOB cases and run the tree on them. Now take the number of votes for the correct class in the original OOB data and subtract from them the number of correct votes in the variable-$d$-permuted OOB data. The average of this number over all the trees is the raw importance of variable $d$.

**Random Forest and Imbalanced Data**  As discussed in section 3.7 SMOTE is used for oversampling the minority class. The random forest algorithm has a way of dealing with imbalanced data internally through the use of a cost matrix for misclassification and over-sampling techniques. This combination was first proposed by Chen et al.[53]. If a cost matrix is used the algorithm generates in-bag samples by oversampling classes with large misclassification costs and under-sampling classes with small misclassification costs. This results in a balanced in-bag training set and an imbalanced OOB test set where the clinically relevant class carries high misclassification costs. In our case, when using a small dataset, we can expect high variance in
the OOB classification error. This is because the random forest is trying to minimize the overall error rate. Because our dataset is small, highly unbalanced and should have a highly skewed cost matrix (i.e., a false negative is way worse than a false positive) it was chosen not to use this functionality. By using an over-sampled dataset without misclassification costs the same data can be used by all three algorithms which is more consistent for the experimental design of this research. Besides that the performance of the different models can be safely compared this way.

5.2 Experimental Design

Now the intended plan for training, testing and validating the prediction models is outlined. Recall that two datasets were constructed since we would like to run the prediction model at the moment of AI finding (M1) and at the moment after biochemical evaluation (M2). Firstly for every dataset a validation set was created using a holdout of 15% of the original unbalanced data. The validation set is used to validate if the prediction model, trained on oversampled data will also perform well on the original data. The remaining data was used for training and testing the model.

As discussed earlier, in order to overcome the problem of imbalanced data, we use the synthetic minority over-sampling technique (SMOTE)[11]. For every dataset cases were generated until the dataset was balanced. For dataset M1 we used 10-fold cross-validation to better estimate
the performance of the models, so that 90% of the over-sampled data were used for training and 10% for testing in each fold. For dataset M2 the same design was applied. The experimental design can be found in figure 5.4. Per dataset the same train, test and validation sets are used for all three classification models. In the end the performance, averaged over all folds, of the model on the original validation set and of the model on the over-sampled test set is evaluated separately.

5.3 Performance Metrics

The prediction models in this project provide predictions for a binary outcome. The outcome is either clinically relevant or not. This situation calls for a models that accurately distinguishes those at high risk (i.e. clinically relevant) from those at low risk (i.e. not clinically relevant). Steyerberg and colleagues state that overall performance measures can be categorized into two groups: discrimination metrics and calibration metrics[54]. Discrimination refers to the degree of discrimination between those with and without the outcome. Calibration refers to the agreement between observed outcomes and predicted outcomes. The predictive models are evaluated based on discriminative performance metrics.

A classifier is typically evaluated using a confusion matrix as illustrated in figure 5.5. The columns in this matrix represent the actual or real class and the rows are the predicted class. In the confusion matrix, TN is the number of negative examples correctly classified (true negatives), FP is the number of negative examples incorrectly classified as positive (false positives), FN is the number of positive examples incorrectly classified as negative (false negatives) and TP is the number of positive examples correctly classified as positive (true positives).

<table>
<thead>
<tr>
<th></th>
<th>Actual Positive</th>
<th>Actual Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted Positive</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td>Predicted Negative</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

Figure 5.5: Confusion matrix example

From the confusion matrix we can obtain the fraction of truly positive cases \( p \) through \(((TP + FN)/Total)\), \( n \) is the fraction truly negative \(((TN + FP)/Total)\), the fraction of predicted positive \( \hat{p} \) through \(((TP + FP)/Total)\) and the fraction of predicted negative \( \hat{n} \) through \(((TN + FN)/Total)\). The discriminative metrics that used for evaluation:
**Accuracy** \(\frac{(TP + TN)}{Total}\): The percentage of patients that were classified correctly.

**Sensitivity** \(\frac{TP}{(TP+FN)}\): Also called the true positive rate (TPR), or the recall in some fields. The percentage of clinically relevant patients that are correctly classified as such.

**Specificity** \(\frac{TN}{(FP+TN)}\): Also called the true negative rate (TNR). The percentage of patients not clinically relevant that are correctly classified as such.

**Precision** \(\frac{TP}{(TP+FN)}\): Also called the positive predictive value (PPV). The fraction of retrieved instances that are relevant.

A specific model has a small number of false negatives, while a sensitive model predicts a large number of true positives. Ideally, the model should be both sensitive and specific, but these two goals often conflict, so one needs to make a trade-off between the two objectives. Additionally the following metrics are used:

**Receiver Operating Characteristic (ROC) Curve**: Plots the sensitivity (true positive rate) against 1-specificity (false positive rate). When a classifier computes a continuous valued output (i.e. estimated likelihood), it can be turned into a binary one by applying a threshold to the output. If the output value is above the threshold, the pattern is classified to the positive class. Otherwise, it is labeled as belonging to the negative class. The ROC curve can be computed by varying the value of the threshold and recording the false positive and true positive rates for all possible thresholds. When a ROC curve of classifier A dominates the one of B (i.e. it has higher true positive values for all values of the false positive rate), one can conclude that A performs better than B.

**Area Under the Curve (AUC)**: Area under the ROC curve. The AUC of a random classifier is 0.5, because its ROC curve is the straight line. Since we are typically interested in classifiers which have some positive added value over a random guess, the AUC range which is of interest [0.5-1] can be linearly transformed to the range [0-1]. The AUC has some attractive features, most notably its objectivity. Given a testing data set and a classifier, every researcher will get the same AUC[55].

**Kappa statistic** \(\kappa\) or Cohen’s kappa is a statistic which measures inter-rater agreement for qualitative (categorical) items. It is generally thought to be a more robust measure than simple percent agreement calculation, since \(\kappa\) takes into account the agreement occurring by chance. Cohen’s Kappa is a scalar evaluation metric defined as:

\[
\kappa = \frac{a - p_c}{1 - p_c} = \frac{accuracy - (\hat{p} + \hat{n})}{1 - (\hat{p} + \hat{n})} \]  

(5.8)
Here $a$ is the observed agreement among classifiers (i.e. accuracy) and $p_c$ is defined as the probability of predicting the right class due to chance. $(p\hat{p} + n\hat{n})$. This means that it takes into account the class priors, a property not shared with the AUC\cite{55}. Kappa ranges from 1 to -1. Random guessing results in a $\kappa$ of 0. For example, a $\kappa = 0.5$ shows that the model performs 50% better than if the classification was done only by chance. The most interesting property of $\kappa$ is the fact that it favors correct classifications of the minority class over that of the majority class. This is considered useful when dealing with an imbalanced dataset where it is more important to correctly predict the minority class.

**Area Under Kappa (AUK):** Plot Kappa against the false positive rate as a way of analyzing the performance of a classifier and to assess its performance over all thresholds by computing the Area Under the Kappa (AUK) curve.

### 5.4 Model Building: Three Models for Datasets M1 and M2

Now the models are built using datasets M1 and M2. According to the modeling technique certain parameter configurations have to be selected, these are documented. As explained in section 5.2 every model is be trained and tested using 10-fold cross-validation on the over-sampled dataset. Then the model is validated on a 15% holdout sample of the original data. For both the testing and validation step a ROC- and AUK-curve is constructed for every model. Every curve carries their respective performance metrics averaged over the folds. Then in the end the performance of all models is assessed in terms of best average performance and standard deviation over all folds. Note that for the discriminative performance metrics based on a threshold to map the predicted likelihoods to 0 or 1 a default cut-off of 0.5 is used. This does not influence the scalar AUC and AUK metrics though.

**Predictive Models: Moment of Finding M1**

Here the models are built using dataset M1 which contains variables known at the moment of the AI finding. The reduced set of features that resulted from the feature selection process are: gender and size.

**Logistic Regression Model M1**

The model found in table 5.1 is constructed using the over-sampled training set of the first fold. Note that there are 10 models in total, one for every fold.
As suspected judging from literature size is a significant predictor \([16][14]\). Besides that its coefficient estimate \(\beta_{\text{size}}\) is 3.24 which indicates that it has a lot of influence on the log-odds ratio. Variable gender doesn’t have much predictive power. it is far from significant and the coefficient \(\beta_{\text{gender}}\) of 0.13 is quite small. The \(R^2\) or coefficient of determination of 47% can be interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable.

**Random Forest Model M1**

The random forest model was build using the over-sampled training data. Figure 5.6 shows the classification error for a forest of 200 trees grown on the reduced feature set in the first fold. At around 55 trees the error drops to 9% and from 70 till 200 trees it remains constant at 10%. The minimum leaf size was set to 1 which is default for classification trees.

**Fuzzy Inference System M1**

The fuzzy rules in the fuzzy inference system provide an interpretable representation of the general behavior of the model, which facilitates communication with domain experts. As discussed earlier we use a first-order Takagi–Sugeno (TS) fuzzy inference system (FIS) for this problem. To build the model, we used fuzzy c-means clustering [46] and the ANFIS (adaptive neuro-fuzzy inference system) learning method[44]. Each cluster corresponds to a rule in the FIS. The FIS was trained for different numbers of clusters \(c = 1, \ldots, 10\). It turned out that two clusters yielded the best performance for the over-sampled dataset. These clusters were used to initialize the rule base of a first-order TS model, the parameters of which where optimized using ANFIS training.

The FIS that was trained in the first fold is used to analyze the membership functions. The membership functions in figure 5.7 show two distinct rules for \(\log_{\text{size}}\). The centers correspond to a size of 11.3 mm for the 'small' group and 33.2 mm for the 'big' group.
membership functions of male and female are very wide, but still it’s clear to which gender type they belong. The following two fuzzy rules results from the model:

1. **IF** size is large **AND** male is no **AND** female is yes **THEN** clinically relevant is yes

2. **IF** size is small **AND** male is yes **AND** female is no **THEN** clinically relevant is no
Predictive Models: Biochemical Evaluation Done M2

Here the models are built using dataset M2 which contains variables known at the moment after biochemical screening. The reduced set of features that resulted from the feature selection process are: age, log_size, MNKRMeta_KR_oud_, STD_MNKRMeta_KR_oud_, MAX_NMKRNorm_KR_oud_ and MAX_MNKRMeta_KR_oud_.

Logistic Regression Model M2

The model found in table 5.2 is constructed using the over-sampled training set of the first fold. Note that there are 10 models in total, one for every fold. There are no significant deviations in performance between the models (see section 6.3).

<table>
<thead>
<tr>
<th></th>
<th>( \beta )</th>
<th>(S.E.)</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(intercept)</td>
<td>-7.406</td>
<td>(2.619)</td>
<td>0.004</td>
</tr>
<tr>
<td>age</td>
<td>-0.077</td>
<td>(0.028)</td>
<td>0.006*</td>
</tr>
<tr>
<td>size</td>
<td>3.859</td>
<td>(0.740)</td>
<td>1.88\times10^{-7}*</td>
</tr>
<tr>
<td>MNKRMeta_KR_oud_</td>
<td>0.040</td>
<td>(0.033)</td>
<td>0.221</td>
</tr>
<tr>
<td>STD_MNKRMeta_KR_oud_</td>
<td>0.181</td>
<td>(0.062)</td>
<td>0.003*</td>
</tr>
<tr>
<td>MAX_NMKRNorm_KR_oud_</td>
<td>0.004</td>
<td>(0.001)</td>
<td>0.036*</td>
</tr>
<tr>
<td>MAX_MNKRMeta_KR_oud_</td>
<td>-0.079</td>
<td>(0.035)</td>
<td>0.027*</td>
</tr>
<tr>
<td>( N )</td>
<td>153</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( R^2 )</td>
<td>0.51</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*\( p \leq 0.05 \)

Table 5.2: Logistic regression model M2

In logistic regression model M2 all variables except for MNKRMeta_KR_oud_ are significant predictors. Coefficient estimate \( \beta_{size} \) is 3.85 which indicates that it has a lot of influence on the log-odds ratio. The coefficient of variable STD_MNKRMeta_KR_oud_ is 0.18 and therefore has some influence. Coefficients of the MAX variables don’t have a lot of influence on the outcome but they are both significant. The \( R^2 \) or coefficient of determination of 51% can be interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable. In comparison to logistic regression model M1 an increase in \( R^2 \) of 4% is realized by adding the features from biochemical screening.

Random Forest Model M2

The random forest model was build using the over-sampled training data. Figure 5.8 shows the classification error for a forest of 200 trees grown on the reduced feature set in the first fold.
The estimated out-of-bag error has a large variance, therefore it is more difficult to interpret than at M1. This has to do with the fact that the dataset is quite small \((N=153)\). At around 70 trees the error drops to 16% and from 80 till 200 trees it varies between 16% and 19.5%. The minimum leaf size was set to 1 which is default for classification trees.

![Classification error random forest M2](image)

**Figure 5.8:** Classification error random forest M2

**Fuzzy Inference System M2**

For fuzzy inference system M2 it turned out that two clusters were sufficient. These clusters were used to initialize the rule base of a first-order TS model, the parameters of which where optimized using ANFIS training. The FIS that was trained in the second fold is used to analyze the membership functions since that model is easier to interpret. The membership functions in figure 5.9 generally show one clear and narrow rule and one very wide rule. This applies for age, log_size, MNKRMeta_KR_oud_, MAX_MNKRNNorm_KR_oud_ and MAX_MNKRMeta_KR_oud_. The membership functions of STD_MNKRMeta_KR_oud_ show proper dissimilarity between the clustered rules. Other membership functions are less explanatory since they overlap, i.e. are more or less similar. Similarity is defined as the degree to which the fuzzy sets are equal. Fuzzy models, especially if acquired from real data, may contain redundant information in the form of similarity between fuzzy sets. Linguistic interpretation of these rules is difficult as it is not trivial to assign qualitatively meaningful labels to highly similar fuzzy sets. Therefore using rule base simplification is argued. Rule base simplification is a way to deal with two or more similar fuzzy sets by merging them to create a new fuzzy set representative of the merged sets. By substituting this new fuzzy set for the ones merged in the rule base, the number of fuzzy rules
The resulting fuzzy rules are:

1. **IF** age is old **AND** size is small **AND** meta-creat ratio is low **AND** STD-meta-creat ratio is low **AND** MAX-normeta-creat ratio is low **AND** MAX-meta-creat ratio is low **THEN** clinically relevant is no

2. **IF** age is young **AND** size is large **AND** meta-creat ratio is high **AND** STD-meta-creat ratio is high **AND** MAX-normeta-creat ratio is high **AND** MAX-meta-creat ratio is high **THEN** clinically relevant is yes

*Figure 5.9: FIS membership functions M2*
Chapter 6

Results of Predictive Modeling

The results obtained by the predictive modeling process are evaluated in two parts. Firstly the results of feature selection are derived resulting in the subsets of variables that have most predictive power. Secondly the performance of the prediction models based on these variables is assessed and compared.

6.1 Results of Feature Selection

Based on the results from Lasso logistic regularization and the random forest the following variables are selected for the reduced feature set of dataset M1: (1) gender and (2) log_size.

The results of Lasso regularization, discussed first, are in line with the subsequent results of the random forest algorithm.

Select Features M1

Lasso Regularization M1
First dataset M1 is selected and missing values are imputed using the KNNimpute algorithm (euclidean distance with $K=6$). Then the lassoglm function in MATLAB was used to create a cross-validated fit. All input variables were standardized and the number of lambda values for which a regression model has to be calculated was set to 25. The Lasso uses 10-fold cross-validation to estimate the deviance. Deviance refers to the quality of the fitted model applied to the data used to perform the fit. The Alpha sets the weight of lasso ($L_1$) versus Ridge ($L_2$) optimization. Alpha = 1 represents Lasso regression, other versions like ridge optimization and elastic nets are not considered since we are only interest in shrinkage of the coefficients.
The values for deviance in figure 6.1 represent the estimated expected deviance of the model applied to new data calculated using cross validation. It can be observed that the interval of the cross-validated output is not very wide which indicates stable results. The plot identifies the minimum-deviance point with a green circle and dashed line as a function of the regularization parameter Lambda. The blue circled point has minimum deviance plus no more than one standard deviation.

The trace plot in figure 6.2 shows nonzero model coefficients as a function of the regularization parameter $\lambda$. It can be observed that as $\lambda$ increases to the left, \texttt{lasso}lm first sets the age coefficients to zero, then size. The plot identifies the minimum-deviance point with a green dashed line where regularization parameter $\lambda$ is 0.00046. At this value the following coefficients were found:

- constant :-5.9864
- age :-1.5718
- size :9.0313

age was not selected though because is was not selected for minimum deviance plus one standard deviation. The suggestion is that the choice of one standard error is entirely heuristic, based on the sense that one standard error typically is not large relative to the range of $\Lambda$ values[32].
Random Forest M1
First dataset M1 is selected and missing values are imputed using the KNNimpute algorithm (euclidean distance with $K=6$). Then the TreeBagger function in MATLAB was used to create a random forest. A total of 300 trees were build. For classification, it is best to set the minimal leaf size $\text{MinLeafSize}$ to 1 and select the square root of the total number of features for each decision split at random. These settings are default for TreeBagger if applied to classification problems. Furthermore the forest should calculate the Out-Of-Bag (OOB) variable importance, this is used for feature selection. The Cost matrix where $C(i,j)$ is the cost of classifying a point into class $j$ if its true class is $i$ (i.e. the rows correspond to the true class and the columns correspond to the predicted class) has been considered but was not used. If Cost is highly skewed, then, for in-bag samples, MATLAB over-samples patients that are clinically relevant. This was avoided because the sole purpose of using a random forest here is to estimate feature importance of the original dataset. If we select features based on the over-sampled set these don’t necessarily do well on the original imbalanced set.

It can be observed from figure 6.3 that the OOB error was at its lowest when the number of trees was around 170. The classification error was approximately 6% in this area. Another thing that occurs it that there is no need to grow more then 200 trees in total since the error doesn’t become smaller.
For each feature, the feature importance measure is the difference between the number of raised margins (errors) and the number of lowered margins if the values of that variable are permuted across the OOB observations. This measure is computed for every tree, then averaged over the entire ensemble and divided by the standard deviation over the entire ensemble. In MATLAB this variable importance measure is referred to as OOBPermutedVarDeltaError. Values of OOBPermutedVarDeltaError can occasionally get negative. In this case they are usually small in magnitude (a fraction of one) and represent a small fraction of all predictors[52]. This effect gets worse if you have very wide data (lots of predictors and just a few observations). A large fraction of negative values indicates that data is noise and the random forest likely is not learning much. Figure 6.4 shows that log_size has a lot of predictive power which is in line with the results from Lasso regularization. If we chose an arbitrary cut-off of 0.2 based on the results in figure 6.4 the reduced dataset includes variables log_size and gender.

**Select Features M2**

Based on the results the random forest reduced subset of variables was selected because it yielded best performance. This conclusion was drawn based on the poor performance of the Lasso regularization model at minimum-deviance plus one standard deviation. The following variables are selected for the reduced feature set of dataset M2: age, log_size, MNKRMeta_KR_oud_, STD_MNKRMeta_KR_oud_, MAX_NMKNRNorm_KR_oud_ and MAX_MNKRMeta_KR_oud_. First the results of Lasso regularization are discussed followed by the random forest algorithm.
Lasso Regularization M2

First dataset M2 is selected and missing values are imputed using the KNNimpute algorithm (euclidean distance with $K=6$). The `lasso1m` function in MATLAB was used in the same way as for dataset M1.
The values for deviance in figure 6.5 show that the interval of the cross-validated deviance is very wide for smaller values of $\lambda$. The plot identifies the minimum-deviance point with a green dashed line where regularization parameter $\lambda$ is 0.035. In this situation we go with the minimum deviance point because at minimum deviance plus one standard deviation all coefficients are zero. This is against the heuristic of minimum-deviance plus one standard deviation in the error mentioned earlier but at least it leaves us with results which we can interpret.

The trace plot shows nonzero model coefficients as a function of the regularization parameter $\lambda$. It can be observed from figure 6.6 that as $\lambda$ increases to the left all variable coefficients shrink to zero. The following coefficients were found at the minimum deviance point:

- constant: -3.9367
- age: -1.3775
- size: 6.2300
- NMKNorm_KR_oud_: 0.0466
- MIN_MNMetanef_oud_: -0.0456

Figure 6.6: Trace plot of $\lambda$ for dataset M2

90
Random Forest M2

First dataset M2 is selected and missing values are imputed using the KNNimpute algorithm (euclidean distance with \( K = 6 \)). The TreeBagger function was used to create a random forest with the same settings as for dataset M1.

![Figure 6.7: Out-of-bag classification error for dataset M2](image)

If can be observed from figure 6.7 that the OOB error becomes stable at 100 trees. The classification error is approximately 0.125 in this area. Another thing that occurs is that there was no need to grow more than 100 trees in total. Figure 6.8 shows that again \( \log\_\text{size} \) has a lot of predictive power. If we chose an arbitrary cut-off of 0.2 based on the results in figure 6.8 the reduced dataset includes the 6 most important variables. These are: \( \text{age}, \log\_\text{size}, \text{MNKRMeta}_\text{KR_oud}, \text{STD}_\text{MNKRMeta}_\text{KR_oud}, \text{MAX}_\text{NMKRNorm}_\text{KR_oud} \) and \( \text{MAX}_\text{MNKRMeta}_\text{KR_oud} \).
Figure 6.8: Feature importance for dataset M2
6.2 Results of Predictive Models

The results of the clinical prediction models are interpreted for all three models at the moment of finding M1 and after biochemical evaluation M2.

Results of Models M1

Logistic Regression M1

![Averaged ROC curve for model: LR M1 SMOTE](image1)

![Averaged Kappa curve for model: LR M1 SMOTE](image2)

Figure 6.9: M1: Results for the logistic regression model on the over-sampled test set

Results for the logistic regression model on the over-sampled dataset in figure 6.9 are very good considering that there are just two predictor variables. The area under the curve (AUC) can be interpreted as the probability (89%) that in a pair of individuals, one who is and one who is not clinically relevant, the individual who is had the higher predicted probability. It can be observed that the percentage of clinically relevant patients that are correctly classified as such, i.e. the sensitivity is 82%. Sensitivity quantifies the avoiding of false negatives, as specificity does for false positives. Most of the time a trade-off between sensitivity (82%) and specificity (79%) exists. Since we care less for false positive values it is best when sensitivity is higher. The Cohen’s kappa ($\kappa$) of 0.61 shows that the model performs 61% better than if the classification was done only by chance which is in line with other results. Overall the results on the over-sampled data are good.
Inspecting the results on the unbalanced (original) validation set in figure 6.10 we observe a tremendous drop in performance. The main reason for this lies in the class imbalance, the validation set contains only 5 clinically relevant patients. This is the reason why both graphs are jagged. A sensitivity of 40% indicates that the logistic regression model managed to correctly predict 2 out of 5 to be clinically relevant. The Cohen’s kappa ($\kappa$) shows that the model performs just 5% better than if the classification was done only by chance which is very poor.

**Random Forest M1**

Results of the random forest on the over-sampled dataset in figure 6.11 are excellent. The AUC of 0.94 and AUK of 0.44 demonstrate good performance over the full range of thresholds. It can be observed that the percentage of clinically relevant patients correctly classified as such is 86%. The specificity indicates that 94% of the negative cases are correctly classified as such. Since the class balance is approximately 50 - 50 here it can be concluded that the random forest model better detects true negative cases than true positive cases. The Cohen’s kappa ($\kappa$) shows that the model performs 80% better than if the classification was done only by chance which is very good.

Results on the unbalanced (original) validation set in figure 6.12 again show a performance drop. A sensitivity of 40% indicates that the random forest model managed to correctly predict 2 out of 5 to be clinically relevant. The Cohen’s kappa ($\kappa$) shows that the model performs 21% better than if the classification was done only by chance. This is quite good given kappa’s
<table>
<thead>
<tr>
<th>False positive rate</th>
<th>True positive rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 6.11:** M1: Results for the random forest model on the over-sampled test set

**Figure 6.12:** M1: Results for the random forest model on the unbalanced validation set

ability to account for class imbalance. An AUK of 0.06 is not very good. Although the results on the over-sampled dataset looked promising, the random forest doesn’t perform excellent on the validation set in terms of true positives.

**Fuzzy Inference System M1**

Results of the fuzzy inference system on the over-sampled dataset in figure 6.13 are good. The AUC of 0.89 and AUK of 0.39 demonstrate good performance over the full range of thresholds.
Figure 6.13: M1: Results for the fuzzy inference system model on the over-sampled test set

It can be observed that the percentage of clinically relevant patients correctly classified as such is 85%. The specificity indicates that 75% of the negative cases are correctly classified as such. The Cohen’s kappa ($\kappa$) shows that the model performs approximately 60% better than if the classification was done only by chance which is good.

Figure 6.14: M1: Results for the fuzzy inference system model on the unbalanced validation set

Results on the unbalanced (original) validation set in figure 6.14 present a sensitivity of 56%. This indicates that the fuzzy inference system most of the time managed to correctly predict 3
of 5 to be clinically relevant. The Cohen’s kappa ($\kappa$) shows that the model performs just 6% better than if the classification was done only by chance. This is quite quite poor. An AUK of 0.08 is quite poor. Although the results on the over-sampled dataset weren’t as high as for example the random forest model, the fuzzy inference system outperforms on the validation set in terms of true positives.

**Results of Models M2**

**Logistic Regression M2**

![Averaged ROC curve for model: LR M2 SMOTE](image)

![Averaged Kappa curve for model: LR M2 SMOTE](image)

Figure 6.15: M2: Results of the logistic regression model on the over-sampled test set

Results of the logistic regression model on the over-sampled dataset in figure 6.15 are very good. The AUC of 0.88 and AUK of 0.38 demonstrate good performance over the full range of thresholds. It can be observed that the percentage of clinically relevant patients correctly classified as such is 84%. The specificity indicates that 76% of the negative cases are correctly classified as such. As discussed earlier a higher sensitivity is favorable. The Cohens kappa ($\kappa$) shows that the model performs 61% better than if the classification was done only by chance which is very good.

Inspecting the results on the unbalanced (original) validation set in figure 6.16 we observe that performance is very high. The validation set contains only 2 clinically relevant patients this time so performance of all models on the validation set is highly variable. The logistic regression
model managed to predict both of them correctly though. The Cohen’s kappa ($\kappa$) shows that the model performs 45% better than if the classification was done only by chance which is good. AUK is 0.26 which underlines good performance.

**Random Forest M2**

Results of the random forest on the over-sampled dataset in figure 6.17 are excellent. The AUC
of 0.89 and AUK of 0.39 demonstrate good performance over the full range of thresholds. It can be observed that the percentage of clinically relevant patients correctly classified as such is 83% which is high. The specificity indicates that 75% of the negative cases are correctly classified as such. This indicates that random forest model M2 better detects true positive cases than true negative cases as opposed to random forest model M1. The Cohen’s kappa ($\kappa$) shows that the model performs 59% better than if the classification was done only by chance which is good.

![Averaged ROC curve for model: RF M2 ORI](image1)

![Averaged Kappa curve for model: RF M2 ORI](image2)

**Figure 6.18:** M2: Results for the random forest model on the unbalanced validation set

The results on the unbalanced (original) validation set in figure 6.18 are again extraordinary high. A sensitivity of 100% indicates that the random forest model correctly predicted both clinically relevant patients. The Cohen’s kappa ($\kappa$) shows that the model performs 51% better than if the classification was done only by chance. An AUK of 0.23 is very good. The results on the over-sampled dataset looked promising and the random forest performed very good on the validation set.

**Fuzzy Inference System M2**

Results of the fuzzy inference system on the over-sampled dataset in figure 6.19 are very good. The AUC of 0.89 and AUK of 0.39 demonstrate good performance over the full range of thresholds. It can be observed that the percentage of clinically relevant patients correctly classified as such is 81%. The specificity indicates that 76% of the negative cases are correctly classified as such. The Cohens kappa ($\kappa$) shows that the model performs approximately 57% better than if the classification was done only by chance which is good.
Figure 6.19: M2: Results for the fuzzy inference system model on the over-sampled test set

Figure 6.20: M2: Results for the fuzzy inference system model on the unbalanced validation set

Results on the unbalanced (original) validation set in figure 6.20 present a sensitivity of just 65%. The negative Cohen's kappa ($\kappa$) shows that the model performs 29% better than if the classification was done only by chance. An AUK of 0.18 is reasonable. The FIS didn’t perform as good on the unbalanced validation set as other prediction models.
6.3 Model Ranking

Now the performance of the classification models can be compared. For the models at M1 and M2 the performance on the over-sampled test data and original validation data is compared. So a total of four comparison tables is given. The highest performance metrics for every model is highlighted. Please note that the results on the validation dataset should be interpreted with care, especially for the predictions after biochemical screening. The reason for this is the relatively small size of the dataset (n=111), and therefore the results are highly dependent on which patients are in the holdout sample.

**Table 6.1:** M1: Model comparison on the over-sampled test data

<table>
<thead>
<tr>
<th></th>
<th>LR M1</th>
<th></th>
<th>RF M1</th>
<th></th>
<th>FIS M1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>µ</td>
<td>σ</td>
<td>µ</td>
<td>σ</td>
<td>µ</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.80</td>
<td>0.04</td>
<td>0.90</td>
<td>0.03</td>
<td>0.80</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.82</td>
<td>0.05</td>
<td>0.86</td>
<td>0.05</td>
<td>0.85</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.79</td>
<td>0.05</td>
<td>0.94</td>
<td>0.04</td>
<td>0.75</td>
</tr>
<tr>
<td>Precision</td>
<td>0.80</td>
<td>0.04</td>
<td>0.93</td>
<td>0.04</td>
<td>0.77</td>
</tr>
<tr>
<td>AUC</td>
<td>0.89</td>
<td>0.03</td>
<td>0.94</td>
<td>0.03</td>
<td>0.89</td>
</tr>
<tr>
<td>Kappa</td>
<td>0.61</td>
<td>0.09</td>
<td>0.80</td>
<td>0.06</td>
<td>0.60</td>
</tr>
<tr>
<td>AUK</td>
<td>0.39</td>
<td>0.03</td>
<td>0.44</td>
<td>0.03</td>
<td>0.39</td>
</tr>
</tbody>
</table>

**Table 6.2:** M1: Model comparison on the unbalanced validation data

<table>
<thead>
<tr>
<th></th>
<th>LR M1</th>
<th></th>
<th>RF M1</th>
<th></th>
<th>FIS M1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>µ</td>
<td>σ</td>
<td>µ</td>
<td>σ</td>
<td>µ</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.73</td>
<td>0.00</td>
<td>0.88</td>
<td>0.01</td>
<td>0.66</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.40</td>
<td>0.00</td>
<td>0.40</td>
<td>0.00</td>
<td>0.56</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.75</td>
<td>0.00</td>
<td>0.91</td>
<td>0.01</td>
<td>0.66</td>
</tr>
<tr>
<td>Precision</td>
<td>0.08</td>
<td>0.00</td>
<td>0.20</td>
<td>0.02</td>
<td>0.08</td>
</tr>
<tr>
<td>AUC</td>
<td>0.72</td>
<td>0.00</td>
<td>0.64</td>
<td>0.02</td>
<td>0.72</td>
</tr>
<tr>
<td>Kappa</td>
<td>0.05</td>
<td>0.00</td>
<td>0.21</td>
<td>0.02</td>
<td>0.06</td>
</tr>
<tr>
<td>AUK</td>
<td>0.08</td>
<td>0.00</td>
<td>0.06</td>
<td>0.01</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Table 6.1 shows that the random forest (RF) outperforms logistic regression (LR) and the fuzzy inference system (FIS). LR and FIS show almost identical performance which is interesting given their difference and nature. The models perform very consistent over all folds given their low standard deviation.

The results in table 6.2 show that the RF was overfitting the over-sampled training data. FIS clearly performs best on the metrics that are of greatest interest for us. Sensitivity, AUC and AUK demonstrate reasonable generalization of the FIS.

<table>
<thead>
<tr>
<th></th>
<th>LR M2</th>
<th>RF M2</th>
<th>FIS M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.81</td>
<td>0.79</td>
<td>0.79</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.84</td>
<td>0.83</td>
<td>0.81</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.76</td>
<td>0.75</td>
<td>0.76</td>
</tr>
<tr>
<td>Precision</td>
<td>0.82</td>
<td>0.81</td>
<td>0.82</td>
</tr>
<tr>
<td>AUC</td>
<td>0.88</td>
<td>0.89</td>
<td>0.89</td>
</tr>
<tr>
<td>Kappa</td>
<td>0.61</td>
<td>0.59</td>
<td>0.57</td>
</tr>
<tr>
<td>AUK</td>
<td>0.38</td>
<td>0.39</td>
<td>0.39</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>LR M2</th>
<th>RF M2</th>
<th>FIS M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.79</td>
<td>0.83</td>
<td>0.78</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>1.00</td>
<td>1.00</td>
<td>0.65</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.76</td>
<td>0.81</td>
<td>0.79</td>
</tr>
<tr>
<td>Precision</td>
<td>0.38</td>
<td>0.43</td>
<td>0.31</td>
</tr>
<tr>
<td>AUC</td>
<td>0.96</td>
<td>0.93</td>
<td>0.84</td>
</tr>
<tr>
<td>Kappa</td>
<td>0.45</td>
<td>0.51</td>
<td>0.29</td>
</tr>
<tr>
<td>AUK</td>
<td>0.26</td>
<td>0.23</td>
<td>0.18</td>
</tr>
</tbody>
</table>
When comparing the results in table 6.3 it seems that the difference in performance is much smaller. LR performs slightly better on the smaller dataset M2 which is higher in dimensionality. This time RF and FIS are comparable in terms of average results but the standard deviation are also higher. This indicates that having just 111 observations results in more variance among the folds. The Cohen’s kappa of LR is one metric in which it truly differentiates from FIS.

The results in table 6.4 that again the performance on the over-sampled data doesn’t tell much about the performance on the unbalanced data. This time RF and LR managed to correctly predict both clinically relevant patients in the validation set. AUC and AUK of LR demonstrate good generalization.
Chapter 7

Towards a Prescriptive Model

Based on the prediction models the project moves towards development of a prescriptive model for patient-specific management of adrenal incidentalomas. The fundamental question here is: how can we integrate predictive analytics into a healthcare delivery system? In layman’s terms, the prediction model gives us the estimated likelihood that an individual patient will get a *clinically relevant* outcome. So it tells us what is likely to happen. But what should a clinician do with this information? He’s not looking for an estimated likelihood but wants to know what to do next. So not a prediction but a prescription is the final goal.

In the following section we first evaluate the prediction models in terms of clinical usefulness using decision curve analysis. Secondly the prescriptive model for management of patients with adrenal incidentalomas is introduced. The prescriptive model integrates the prediction models into the Erasmus MC daily practice. Additionally modifications to the Erasmus MC daily practice are effectuated in order to shorten throughput time and make it more cost efficient. In the end the results of the model are discussed yielding the number of patients saved from unnecessary work-up.

### 7.1 Clinical Usefulness: Decision Curve Analysis

There are two moments at which we want to predict what the risk is of a patient being *clinically relevant*. Based on the predicted likelihood of an individual patient a decision is made to stop or proceed with the diagnostic work-up. All prediction models constructed in the modeling phase (see chapter 5) output some form of estimated likelihood for the target variable *clinically relevant*. First some light is shed on how the models construct these outputs:

- Logistic regression: For every observation (i.e. every individual patient) it is the likelihood of this observation originating from this class.
Random forest: For every observation the score generated by each tree is the likelihood of this observation originating from this class computed as the fraction of observations of this class in a tree leaf. These scores are averaged over all trees in the ensemble.

Fuzzy inference system: For every observation the score is an aggregation of the consequents to produce a crisp output for the associated class.

So although the clear differences it can be viewed as that for every individual patient in the validation set the prediction models predicts the likelihood of him being clinically relevant. During model assessment we evaluated the performance of the models by applying them to the patients in the test/validation set and comparing the predictions with the actual outcomes. The performance metrics that were used focus on discrimination but cannot tell if the model is worth using in clinical practice. This is because no information on the consequences is incorporated in the evaluation process. Besides that none of the performance metrics used account for a false negative being much more harmful than a false positive. An example of how this can be misleading is when a model that has a much greater specificity but slightly lower sensitivity than another would have a higher AUC, but would be a poorer choice for clinical use.

Vickers and colleague propose a method called decision curve analysis for evaluating prediction models that incorporates consequences and therefore can tell if they can be used in clinical practice[57]. In this research there are two moments at which we want to predict what the likelihood is of a patient being clinically relevant. Based on the predicted likelihood a decision is made whether to stop the work-up or continue. The application of the decision curve analysis is explained using the prediction at M1, the moment of finding.

If the likelihood of being clinically relevant for an individual patient is near one, then additional follow-up is needed to determine the diagnosis. If the likelihood is near zero, the patient can exit the work-up. At some probability between 0 and 1, the endocrinologist is unsure whether or not to continue. This threshold likelihood $P_t$ for an individual patient is where the expected benefit of treatment is equal to the expected benefit of avoiding treatment. Treatment in this situation refers to the biochemical evaluation. Figure 7.1 shows a decision tree based on the threshold probability $P_t$, and $a$, $b$, $c$ and $d$ as the value for each outcome. This can be interpreted in terms of quality-adjusted life-years. At the likelihood threshold $P_t$ we can solve the following equation:

$$P_t a + (1 - P_t)b = P_t c + (1 - P_t)d$$

$$\frac{a - c}{d - b} = \frac{1 - P_t}{P_t}$$

(7.1)

(7.2)
$d - b$ is the harm associated with a false positive (FP) result compared to a true negative (TN) result. $a - c$ is the harm associated with a false negative (FN) result compared to a true positive (TP) result. From the above equation we can see how the likelihood threshold $P_t$ for an individual patient tells something about how the endocrinologist weighs the relative harms of a FP and FN results. An unstructured interview with endocrinologist dr. R. Feelders, prof. M. Hunink and radiologist dr. J.J. Visser was performed in order to set the parameters needed for the decision curve analysis. It was decided that at the moment of finding a FN would be 99 times worse then unnecessary biochemical screening and CT scans. e.g. sending someone with adrenocortical carcinoma (ACC) home compared to testing a healthy person for hormonal hyperfunction. By equation 7.2 it can be concluded that this corresponds with a likelihood threshold $P_t$ of 1% for every individual patient.

The relationship between the weight of FN compared to FP outcome and threshold likelihood $P_t$ is used to assess the clinical usefulness of a prediction model. This can be done by calculating the net benefit which uses the value of a TP and FP results. First it fixes the value of a TP ($a - c$) at 1. Then it solves the equation 7.2 to retrieve the outcome value of a FP ($b - d$).
\[
\frac{1}{d - b} = \frac{1 - P_t}{P_t}
\] (7.3)

\[(d - b)(1 - P_t) = P_t\] (7.4)

\[(-d + b)(-1 + P_t) = -P_t\] (7.5)

\[b - d = \frac{-P_t}{1 - P_t}\] (7.6)

This is used to calculate the net benefit at threshold likelihood \(P_t\)[58]:

\[\text{Net Benefit} = \frac{TP}{N} - \frac{FP}{N} \left( \frac{P_t}{1 - P_t} \right)\] (7.7)

Here \(TP\) and \(FP\) refer to the number of true positive and true negatives not the outcome value. \(N\) is the total number of patients. Plotting the net benefit for different threshold likelihoods results in a 'decision curve'. Decision curve analysis identifies the range of threshold likelihoods in which a model constitutes a high benefit. To know if the prediction model has any value at a \(P_t\) of 1% we have to compare it to two alternatives: (1) treat all patients and (2) treat none[57]. For the treat all strategy TP are all clinically relevant and FP all not clinically relevant patients. For the treat none strategy TP and FP are both zero hence the net benefit is zero. By plotting both strategies together with the decision curve of the model it can be assessed if the model has any clinical usefulness at a certain threshold likelihood or in a region of thresholds.

Choosing the right threshold likelihood \(P_t\) can be difficult in reality. Besides that clinicians, who make the final decision, differ as to how they rate possible consequences of treatment. Therefore we need to consider the likely range of \(P_t\) in the population, i.e. the typical threshold likelihoods at which clinicians opt for treatment. What follows is an application of decision curve analysis at M1 and M2.

**Decision Curve Analysis M1**

At the moment of finding an adrenal incidentaloma (M1) the prediction models can be used to decide whether to proceed with diagnostic work-up or not. If the models can reliably tell what the risk of being clinically relevant is, the decision curve can show if they are clinically useful at the given threshold likelihood \(P_{t1}\). The net benefit curves of prediction models logistic regression (LR), random forest (RF) and fuzzy inference system (FIS) are compared with strategies treat all (All) and treat none (None).
In figure 7.2 the curves of LR and FIS are only superior to the curve of ‘treat all’ (i.e. biochemical screening for all patients) for thresholds between 1% and 8%. RF clearly shows poorer performance for threshold values below 7% but from there it outperforms LF and FIS up to 11%. To interpret these results the threshold likelihood $P_{M1}^{\text{t}}$ is needed. Recall that it was decided that a FN is 99 times worse then unnecessary biochemical screening. e.g. sending someone with adrenocortical carcinoma (ACC) home compared to testing a healthy person for hormonal hyperfunction. This corresponds to a $P_{M1}^{\text{t}}$ of 1%. Missing ACC is obviously something we want to avoid, although it is not a fast growing cancer and it is unlikely that delaying the diagnosis for a couple of months leads to important harm. Besides that ACC make up for only 8.8% of clinically relevant cases compared to 50% growth cases of benign nodules. Therefore 1% is considered the lower limit and 3% the upper limit. So one estimate for the range of $P_{M1}^{\text{t}}$ based on a back-of-the-envelope session with medical experts is 1 – 3%. It can be concluded that at 1% there is no difference between using LR, FIS and treat all. In the interval 2 - 3% though the FIS has a clear advantage over the rest.
**Decision Curve Analysis M2**

After biochemical screening (M2) three CT scans will follow according to the daily practice. (1 non-contrast and 2 regular CT’s) There is no real consensus literature on the number of CT’s and the interval in between them[59]. Associated with a CT test are costs, hassle for the patient and test harm from radiation. As in many aspects of medicine, there are both benefits and risks associated with the use of CT for follow-up. The main risk is associated with the increased probability of cancer induction from x-ray exposure. The probability for absorbed x-rays to induce cancer is very hard to estimate mainly because estimates for the effective dose from a diagnostic CT procedure can vary a factor of 10 or more depending on the type of CT procedure, patient size and the CT system[60]. Zondervan and colleagues recently indicated that among young adult patients without a cancer diagnosis in whom only one or two scans were obtained, mortality and predicted risk of radiation-induced cancer-death were 1.9% and 0.1% for abdominopelvic CT[61]. They chose the age of 35 as an upper limit, as estimated risk of cancer from low-dose radiation in adults plateaus beyond this age. For this research it can be considered safe to use an estimated probability of 0.1% for an individual patient. Chances are that the actual probability is even lower since the average age of this population is 62. Nonetheless there should be a way to account for this test harm and decide on the number of CT follow-ups based on the risk of the patient being *clinically relevant*.

For now we focus only on the risk of inducing cancer on a healthy patient. There’s a way to account for this test harm when calculating the net benefit. A slightly different formulation of equation 7.7 can be used here:

\[
NetBenefit = \frac{TP}{N} - \frac{FP}{N} \left( \frac{P_t}{1 - P_t} \right) - TestHarm
\]

(7.8)

The harm from the test is a 'holistic' estimate of the negative consequence of having to take the test (cost, inconvenience, medical harms and so on) in the units of a true positive result[57]. At the moment of finding, there is no test harm involved since the tests are non-invasive and cheap. But if we focus on the moment after biochemical screening there is a chance of \(3 \times 0.1 = 0.3\%\) on inducing cancer for every individual patient. At this moment in time we only know that the adrenal mass doesn’t have a hormonal hyperfunction. Subsequent follow-up will answer the questions: (1) does it have malignant characteristics? and (2) does it grow? During the interview it was decided that inducing cancer on a healthy patient is two times worse than missing a *clinically relevant* patient. So the test harm would be rated as \(0.3\% \times 0.67 = 0.002\) when the unit of measure is a true positive result.

In figure 7.3 the curves of models LR and RF look a lot less smooth. This is because of the size
of the original validation set situation M2 (n=16). FIS appears the smoothest but results can fluctuate for different validation sets. LR jumps below the treat all strategy between threshold likelihoods 6 - 12% which means that the model performs worse than when we would perform CT test on all patients. RF is clearly better than FIS for thresholds between 8% and 16%. At 16% or higher the models produce nothing but random noise. It was decided to set the range of thresholds at Moment 2 (M2) a bit wider since hormonal hyperfunction has been ruled out so $P_{t}^{M2}$ ranges from 1% - 5%. From 1% - 5% LR is the best performing model.

### 7.2 Prescriptive Model

Prescriptive models say what is the best suggested course of action. To device a healthcare delivery system that will improve the quality of decisions made during AI work-up it should provide actionable information and suggestions, and not just predictions or predicted likelihoods[26]. A clinician does not simply predict likelihoods or outcomes to get to a patient’s diagnosis. Instead they make high quality optimized decisions automatically to improve patient outcomes. Therefore, a prescriptive analytics system is needed that integrates the prediction model with the Erasmus MC daily practice. In the end the daily practice should be considered a care pathway that gives every individual patient a treatment that suits him best.
To summarize the above points, we want to integrate rules to guide the decision-making process with model predictions for the following two reasons: (1) in order to associate statistical likelihood of certain conditions to specific prescriptions for the next ‘recommended’ action; and (2) in order to augment analytic models with specific domain knowledge that is not captured by the predictive model. The latter refers to incorporating the daily practice including proposed modifications.

The prescriptive model depicted in figure 7.4 is a new algorithm for patient-specific management of patients with adrenal incidentalomas. The process is modeled using Business Process Model and Notation 2.0 (BPMN 2.0). The primary goal of BPMN is to provide a notation that is readily understandable by all business users. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation. Note that every swim-lane refers to a different hospital department. The processes are color coded:

- Green process: AI work-up process that requires presence of the patient.
- Red process: AI work-up process linked to a *clinically relevant* outcome that requires presence of the patient.
- Yellow process: AI work-up process performed internally that doesn’t require presence of the patient.
- Blue process: Prediction model which decides whether to stop or continue AI work-up.

The process start with a request for abdomen or thorax CT scan by the referring department. Note that scans requested because of symptoms indicating adrenal disease are excluded since they can never be incidentalomas. The CT images are immediately inspected after the test. When an incidentaloma is found, the patient situation is assessed by the radiologist and logistic regression model M1 is used to decide if diagnostic work-up should be initiated. For the threshold likelihood $P_{M1}^t$ range between 1 - 3% the model is of added value over the strategy of treat all for an individual patient. For $P_{M1}^t$ lower then 1% the model is no better than strategy of treating all patients. On the other hand it is never worse, and because it is based on routinely collected data, it has no obvious downside. If the predicted likelihood of being *clinically relevant* is less than 1% the patient exits the work-up. The model could be of use for clinicians who, at least some of the time, would opt for biochemical screening if a patient has a predicted likelihood between 1 - 3%.

If AI work-up is initiated biochemical evaluation is performed which in the current daily practice consists of two visits to the endocrinologist with the lab tests in-between. Based on several brainstorming sessions with prof. M. Hunink and dr. J.J. Visser the idea arose to skip the first
visit to the endocrinologist. The reasoning behind this is that tests for ruling out pheochromocytoma (metanephrines, normetanephrines, metanephrine-to-creatinine ratio, normetanephrine-to-creatinine ratio, 24-hour urinary metanephrines, 24-hour urinary normetanephrines) and Cushing’s syndrome (1-mg dexamethasone suppression test) should always be requested. The test for primary aldosteronism (aldosterone) is only requested if the patient is hypertensive and when the patient doesn’t use specific anti-hypertensive drugs (spironolactone and mineralocorticoid receptor blockers). If the patient uses these drugs, he should temporarily stop using them in order for the aldosterone test to be reliable. This scenario is part of the first visit and therefore not modeled separately. If this is not the case aldosterone is requested right away and a new visit to the endocrinologist is planned. If the adrenal has a hormonal hyperfunction or if the lesion is $>4$-$6$ cm adrenalectomy is performed. In case of a patient older than 40 with hormonal hyperfunction and hypertension first adrenal vein sampling is done.

If the adrenal mass is non-functioning and small ($<4$ cm) then logistic regression model M2 is used to decide if 3 subsequent CT scans are needed. The primary goal of subsequent imaging in the work-up of incidentalomas is to distinguish between adrenal adenoma, carcinoma, pheochromocytoma, and metastatic lesions[4]. The diagnosis of an adenoma depends on the identification of lipid in the adrenal lesion, which can either be assessed with a non-contrast CT. If the lesion has benign characteristics, i.e. homogeneous, regular borders and $HU<10$ on non-contrast CT, then it receives follow-up in the form of repeated CT scans. More specifically it should be re-imaged annually for 2 years to check if the lesion grows. If either growth of $>1$ cm in one year or a malignant histology of the lesion are observed then the follow-up process is escalated and adrenalectomy is performed. For the likelihood threshold $P_{t}^{M2}$ range between 1 - 5% logistic regression model M2 is of added value over the strategy of treat all. If the predicted likelihood is less than 1% the patient exits the work-up. Again the model will be of use for clinicians who, at least some of the time, would opt for biochemical screening if a patients predicted likelihood is between 1 - 5%.
Figure 7.4: Prescriptive model for patient-specific management of AI
7.3 Results of Prescriptive Model

Now the newly proposed prescriptive model for patient-specific AI management can be tested based on the logistic regression models M1 and M2 since these were superior in terms of net benefit. By applying the likelihood thresholds to the predicted likelihoods for the patients in the original validation set it is possible to assess how well the models perform.

For every model we evaluate its performance using the lower limit as well as the upper limit threshold. Tables 7.1 and 7.2 show the confusion matrix when $P_{t}^{M1}$ and $P_{t}^{M2}$ are equal to 1%. It can be concluded that in both situations not a single clinically relevant case has been missed, i.e. no false negative cases. There are a lot of false positive cases though which is to be expected when the threshold is low. For the lowest threshold likelihood the number of true negative cases is 3 for M1 and M2 which implies that by implementing the prescriptive model a total of 6 patients would receive no further work-up at these moments. Note that the validation sets of M2 and M2 are randomly drawn from their respective population, so these are not the same patients.

<table>
<thead>
<tr>
<th>Act-Pos</th>
<th>Pred-Pos</th>
<th>Pred-Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act-Pos</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Act-Neg</td>
<td>88</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 7.1: Confusion matrix M1 at $P_{t}^{M1} = 1%$

<table>
<thead>
<tr>
<th>Act-Pos</th>
<th>Pred-Pos</th>
<th>Pred-Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act-Pos</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Act-Neg</td>
<td>11</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 7.2: Confusion matrix M2 at $P_{t}^{M2} = 1%$

Tables 7.3 and 7.4 show the confusion matrix when $P_{t}^{M1}$ is 3% and $P_{t}^{M2}$ is 5%. It is interesting to observe that although this is a riskier version of the prediction models, still no clinically relevant patients are missed. Model M1 saves a total of 15 patients from unnecessary follow-up and Model M2 saves 4. These results are promising and show an increase of 12 patients saved from unnecessary follow-up at M1 and 1 patient at M2.

<table>
<thead>
<tr>
<th>Act-Pos</th>
<th>Pred-Pos</th>
<th>Pred-Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act-Pos</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Act-Neg</td>
<td>76</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 7.3: Confusion matrix M1 at $P_{t}^{M1} = 3%$

<table>
<thead>
<tr>
<th>Act-Pos</th>
<th>Pred-Pos</th>
<th>Pred-Neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act-Pos</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Act-Neg</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 7.4: Confusion matrix M2 at $P_{t}^{M2} = 5%$
Chapter 8

Conclusion

There is demand for an improved management algorithm that limits subjective decision making[25]. A management approach tailored to the individual patient that offers multiple decision moments avoids unnecessary procedures, reduce risks and lead to monetary savings[10]. This research resulted in a patient-specific management approach that offers multiple moments at which we want to assess whether to stop or continue AI work-up. By its development we learned how to transition from predictive to prescriptive analytics in a concrete and standardized way. More specifically it outlines the development of clinical prediction models followed by an evaluation of their usefulness in clinical practice. This allows for the prediction models to be implemented in a standardized business process.

In this chapter we conclude in what way the main problem statement has been addressed by first answering the research questions. Secondly the discussion gives a more in-depth evaluation of the results. In the end an indication for future research is given.

8.1 Answering the Research Questions

We conclude on the results of this research by answering the research questions posed in section 1.3.

RQ 1 Are prediction models based on machine learning algorithms suitable for predicting a clinically relevant outcome of adrenal incidentalomas?

A total of three clinical prediction models were developed capable of predicting if an AI outcome will be clinically relevant at two different moments during work-up: (M1) at the moment of AI finding , and (M2) after biochemical screening. Results on the over-sampled test tests are very good and results on the original validation set are good yet tend to be highly fluctuating. (see section 6.3)
RSQ 1 Which data can be obtained from the Erasmus MC hospital databases to develop the clinical prediction model?
The data sources obtained contain patient data, lab test data, routine measurements data, medication data, international classification of diseases (ICD) data and radiology reports. Work-up data had to be extracted manually from EHR. (see section 3.2)

RSQ 2 Which features have good predictive power for the clinical prediction models?
For prediction it turns out that the size of the adrenal tumor is a highly significant predictor. Besides that several features constructed using the lab test data from biochemical screening were also important. (see section 6.1)

RSQ 3 Can a prescriptive model for patient-specific management of patients with adrenal incidentalomas be derived from this clinical prediction model?
In order to assess the clinical usefulness of the prediction models decision curve analysis is applied at M1 and M2. The most appropriate prediction model for use in clinical practice is integrated with the Erasmus MC daily practice for AI management. This results in a novel prescriptive model for patient-specific management of adrenal incidentalomas, that potentially saves patients that are not clinically relevant from unnecessary follow-up. (see section 7.2)
8.2 Discussion

AI Guideline Adherence  At Erasmus MC management of adrenal incidentalomas is suspected to be highly variable in terms of guideline adherence. Complementary to this previous studies indicate that there is demand for an improved management algorithm that limits subjective decision making[25]. This work supports that desire given that only 18% of all 643 patients with AI’s received biochemical screening, the first mandatory step in incidentaloma work-up. This might sounds like a bold statement because 47% received unrelated imaging follow-up and 13% received AI imaging follow-up only. But it does indicate that clinicians prefer additional imaging instead of referring a patient to the endocrinologist which is not in line with the AI management guideline. Additionally 21% of patients died within 6 months after the AI finding. This indicates that many AI patients have comorbidities or a history of malignancy (56%) since it is highly unlikely that these patients died because of an AI related disease. Despite these mitigating factors 270 out of 643 (42%) AI patients didn’t receive any form of follow-up while still being alive at this very moment. This indicates that many clinicians at Erasmus MC are not aware of the guideline or that they consider it to be an overkill of diagnostic evaluation.

Results of Predictive and Prescriptive Model  This research addresses the above problems by proposing a novel prescriptive model for patient-specific management of adrenal incidentalomas which aims at saving patients from unnecessary follow-up. The prescriptive model is based on prediction models that are used at the moment of finding M1 and after biochemical screening M2. It can be concluded that the performance of the prediction models proves to be high when trained and tested on a over-sampled dataset. Results on the imbalanced validation set are good but highly fluctuating. At the moment of AI finding M1 the random forest (RF) model performs best on the over-sampled dataset while the fuzzy inference system (FIS) yields best results on the unbalanced validation set. At the moment after biochemical screening M2 the logistic regression (LR) model performs best on the over-sampled dataset while the RF yields best results on the unbalanced validation set. Results of the prescriptive model on the M1 validation data (n=96) indicate that 15% of patients at M1 can be saved from unnecessary follow-up while correctly detecting all clinically relevant patients. At M2 25% of the validation set (n=16) are saved from unnecessary follow-up without omitting any clinically relevant patients.

Over-sampling and Cross-validation  With the use of k-fold cross-validation we try to understand how well the prediction models can generalize for new patients. In the experimental design (see section 5.2) we propose to assess the generalizibility based on a 15% hold-out
sample after feature selection. Then over-sampling is performed for training and testing the models. This has been a design decision but we have not investigated if it delivered the best results. There is an active discussion within the machine learning community about when feature selection and over-sampling should be performed. Some claim that it is a malpractice to perform feature selection before going into cross-validation, since it is something that should however be done during cross-validation, so that the selected features are only derived from training data, and not from pooled training and validation data[62]. Additionally over-sampling the minority class can result in overfitting problems if we over-sample before cross-validating. This can be concluded from the results of this research. An expert blog post however claims that this problem can be overcome by over-sampling inside the loop[63]. Altini claims that it makes no sense to create instances based on the current minority class and then exclude an instance for validation, pretending we didn’t generate it using data that is still in the training set. This matter could be inspected further to see what the difference in performance of the prediction models is.

**From Silos to Synergy**  For the proposed prescriptive model to work it relies on some changes in the way of working at Erasmus MC. In the current situation a radiologist performs an imaging examination and in case of AI he advises follow-up according to the adrenal protocol. This is communicated to the referring department or applicant which on its turn has to request a visit to the endocrinologist. The prescriptive model calls for integration of all business functions that provide diagnostic services. In the newly proposed situation the radiologist should take action directly upon find the AI. What would it mean if the diagnostics practice would be truly integrated? This requires different departments that deliver diagnostics to form one integrated business function instead of separate silos. Therefore the model demands for a paradigm shift from different departments delivering each other services to all departments facilitating the integrated AI care pathway. Then diagnostics would be truly integrated.

**Limitations of this Research**

**The Amount of Data**  The dataset that was used for this research has some flaws in terms of numerosity and quality. The first problem simply indicates that for the prediction model to perform better and more reliable a larger dataset is favorable. The models generate results with high variance on the imbalanced validation set. Therefore a larger dataset wouldn’t just be good for training the models, but also for validating them. This problem is rather the rule than the exception. For example, a review of 31 prognostic models in traumatic brain injury showed that 22 were based on samples with less than 500 patients[64]. The main challenges are hence with the development of a good prediction model with a relatively small study sample.
Data Quality  Secondly the quality of the data is a limitation especially in terms of missing values. The most significant variable for prediction is size and in the original dataset it contains approximately 45% missing values. This is due to the fact that it was collected from the AI finding report. Since size was not consistently mentioned in the report by the radiologist this causes the missing values. Another issue concerned the biochemical screening lab test data. For this project only the tests that screen for pheochromocytoma were used in prediction models M2. Cortisol couldn’t be used because the test notes that link it to a 1-mg dexamethasone suppression test were missing in the database export. Additionally there were too many missing values for aldosterone to include it as a numeric variable.

8.3 Future Research

Enrich the Data  One general advise for future research is to enrich the dataset. In order to overcome the data quality issue of missing size values we opt for re-measurement of the tumors on CT images. While doing that additional imaging variables can be collected. HU and washout, determined during non-contrast CT, are variables that indicate if the lesion is malignant. These would be very valuable to include in the prediction models. Fortunately the dataset is being enriched at this moment. Approaching the end of this project a medical student was found to remeasure the adrenal lesions using the original CT images. Doing so the following variables will be added to the dataset in the near future: size (remeasured), hu_before (HU before contrast), hu_after (HU after contrast) and washout. These variables can potentially provide an interesting performance boost.

External Validation  The models were developed and validated using internal data from Erasmus MC. In order to ensure general applicability of a prediction model external validation is essential. Where internal validation techniques are all characterized by random splitting of train and test sets, external validation considers patients that differ in some respect from the development patients\[33\]. Preferably the external validation set is from another institution.

Add Additional Prediction Moments  As discussed in the problem statement current guideline only offer one moment to decide whether to continue or stop work-up; at the moment of finding (M1). This decision is based on just the size of the masses, from that point on every patient is subjected to the same expensive cascade of tests and procedures\[9\][10]. This research added decision moment M2 after biochemical screening. For future research we advise to add additional prediction moments after subsequent imaging follow-ups. This would make the prescriptive model more patient-specific since it not only offers more decision moments, but also takes into account more data.
Bibliography


[8] T. J. Cawood, P. J. Hunt, D. O&apos;apos;apos;Shea, D. Cole, and S. Soule, “Recommended evaluation of adrenal incidentalomas is costly, has high false-positive rates and confers a risk of fatal cancer that is similar to the risk of the adrenal lesion becoming malignant; time for a rethink?,” *European Journal of Endocrinology*, vol. 161, pp. 513–527, 2009.


Chapter 9

Appendices

9.1 Appendix A: Descriptive Statistics

Table 9.1: Frequency table of find_type

<table>
<thead>
<tr>
<th>find_type</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
<td>618.00</td>
<td>97.78</td>
</tr>
<tr>
<td>ECHO</td>
<td>4.00</td>
<td>0.63</td>
</tr>
<tr>
<td>MRI</td>
<td>6.00</td>
<td>0.95</td>
</tr>
<tr>
<td>PET</td>
<td>4.00</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table 9.1 shows through what type of examination the adrenal incidentaloma was discovered. Since the TOPIC database only contained radiology reports from CT’s it is not surprising that they make up for 97% of the Al findings. Other Al’s were identified through one of their imaging follow-ups being a CT within the time horizon 2010 till 2012.

Table 9.2: Frequency table of gender

<table>
<thead>
<tr>
<th>Gender</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>339.00</td>
<td>52.72</td>
</tr>
<tr>
<td>V</td>
<td>304.00</td>
<td>47.28</td>
</tr>
</tbody>
</table>

Table 9.2 shows the gender distribution. The population consists of slightly more male patients.

Table 9.3 displays the quantity of patients that were diagnosed with hypertension at any time in their patient history. This variable was extracted from the clinical letters.
<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31.10</td>
</tr>
<tr>
<td>0</td>
<td>68.90</td>
</tr>
</tbody>
</table>

Table 9.3: Frequency table of hypertension

<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20.53</td>
</tr>
<tr>
<td>0</td>
<td>79.47</td>
</tr>
</tbody>
</table>

Table 9.4: Frequency table of hyp_med

<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>86.00</td>
</tr>
<tr>
<td>1</td>
<td>14.00</td>
</tr>
</tbody>
</table>

Table 9.5: Frequency table of aldos

<table>
<thead>
<tr>
<th></th>
<th>hypMedPre</th>
<th>nohypMedPre</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypMed</td>
<td>91.0</td>
<td>41.0</td>
</tr>
<tr>
<td>nohypMed</td>
<td>67.0</td>
<td>444.0</td>
</tr>
</tbody>
</table>

Table 9.6: Cross-tabulation of hyp_med and hyp_med_pre

<table>
<thead>
<tr>
<th></th>
<th>noFu1</th>
<th>Fu1</th>
</tr>
</thead>
<tbody>
<tr>
<td>noEndo</td>
<td>445.0</td>
<td>83.0</td>
</tr>
<tr>
<td>Endo</td>
<td>18.0</td>
<td>97.0</td>
</tr>
</tbody>
</table>

Table 9.7: Cross-tabulation of endo and fu1

<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>55.52</td>
</tr>
<tr>
<td>2</td>
<td>6.22</td>
</tr>
<tr>
<td>0</td>
<td>38.26</td>
</tr>
</tbody>
</table>

Table 9.8: Frequency table of hist_malign

Table 9.8 displays the quantity of patients that have a history of malignancy.

Table 9.9 shows the percentage of adrenal mass findings that were found unilateral and bilateral. (i.e. one or more lesions found in one of the adrenals or both)
Table 9.9: Frequency table of loc

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>BI</td>
<td>93.00</td>
<td>14.46</td>
</tr>
<tr>
<td>UNI</td>
<td>550.00</td>
<td>85.54</td>
</tr>
<tr>
<td>Unknown</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 9.10: Frequency table of advise

<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120.00</td>
</tr>
<tr>
<td>2</td>
<td>46.00</td>
</tr>
<tr>
<td>3</td>
<td>3.00</td>
</tr>
<tr>
<td>0</td>
<td>474.00</td>
</tr>
</tbody>
</table>

Table 9.10 shows the number of finding reports that contained any form of advise for AI follow-up.

Table 9.11: Frequency table of find_concl

<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>332.00</td>
</tr>
<tr>
<td>0</td>
<td>311.00</td>
</tr>
</tbody>
</table>

Table 9.11 shows the percentage of finding reports in which the finding was mentioned in the conclusion of the report. Note that this is obligatory when an AI is found.

Table 9.12: Frequency table of fu_unrel

<table>
<thead>
<tr>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>300.00</td>
</tr>
<tr>
<td>1</td>
<td>343.00</td>
</tr>
</tbody>
</table>

Table 9.12 shows the percentage of cases that received any form of imaging follow-up after the finding. For example when a patient suffers from lung cancer for which it receives frequent imaging follow-up. In this case the adrenal is also investigated and mentioned in the report.
The radiologist often observes if the lesion has grown. This is referred to as follow-up unrelated to the AI.

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>211.00</td>
<td>32.81</td>
</tr>
<tr>
<td>N</td>
<td>432.00</td>
<td>67.19</td>
</tr>
</tbody>
</table>

**Table 9.13:** Frequency table of dead

<table>
<thead>
<tr>
<th></th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.00</td>
<td>1.87</td>
</tr>
<tr>
<td>0</td>
<td>631.00</td>
<td>98.13</td>
</tr>
</tbody>
</table>

**Table 9.14:** Frequency table of adrl

<table>
<thead>
<tr>
<th>Test</th>
<th>Count</th>
<th>Mean</th>
<th>Std</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFRSKrea_Se_GFR</td>
<td>53543.0</td>
<td>68.5</td>
<td>23.3</td>
<td>3.0</td>
<td>144.0</td>
</tr>
<tr>
<td>HBHemoglobine</td>
<td>74555.0</td>
<td>6.9</td>
<td>1.5</td>
<td>0.0</td>
<td>13.6</td>
</tr>
<tr>
<td>KKalium</td>
<td>71528.0</td>
<td>4.4</td>
<td>2.6</td>
<td>1.2</td>
<td>149.0</td>
</tr>
<tr>
<td>KREAKreatinine</td>
<td>63692.0</td>
<td>105.8</td>
<td>139.5</td>
<td>0.1</td>
<td>5657.0</td>
</tr>
<tr>
<td>LEULeukocyten</td>
<td>54517.0</td>
<td>8.3</td>
<td>8.9</td>
<td>0.1</td>
<td>400.1</td>
</tr>
<tr>
<td>MCVMCV</td>
<td>56750.0</td>
<td>90.5</td>
<td>7.1</td>
<td>1.0</td>
<td>127.0</td>
</tr>
<tr>
<td>NANatrium</td>
<td>66104.0</td>
<td>139.4</td>
<td>8.2</td>
<td>5.0</td>
<td>242.0</td>
</tr>
<tr>
<td>RDWRDW</td>
<td>57323.0</td>
<td>15.5</td>
<td>2.5</td>
<td>9.9</td>
<td>66.8</td>
</tr>
<tr>
<td>TRTrombocyten</td>
<td>52777.0</td>
<td>229.0</td>
<td>136.3</td>
<td>2.0</td>
<td>1840.0</td>
</tr>
</tbody>
</table>

**Table 9.15:** Descriptive statistics of routine lab tests

### 9.2 Appendix B: Data Exploration Visualizations

The heatmap in figure 9.4 shows some that variables KKalium, HBHemoglobine and RDWRDW are highly correlated. The correlation were calculated based on the average of the lab test values per patient, i.e. one average value per patient for the positive (POS) AI class of 643. Figure 9.5 shows no extraordinary relationship for clinically relevant patients within the routine lab tests.
Figure 9.1: Box plots of find_dead

Figure 9.2: Box plots of endo_int grouped by find_class
Figure 9.3: Box plots of ful_int grouped by find_class

Figure 9.4: Correlation heatmap of routine lab tests
Figure 9.5: Scatter plot matrix of routine lab tests grouped by clin_rel
Figure 9.6: Missing value analysis lab data

Figure 9.7: Missing value analysis work-up and patient data
9.3 Appendix C: Anti-hypertensive Drugs

<table>
<thead>
<tr>
<th>Drug Name</th>
<th>Dosage</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMILORIDE HYDROCHLOORTHAZIDETABL</td>
<td>9.5mg</td>
<td>STUK</td>
</tr>
<tr>
<td>AMLODIPINE ALBESIL 5mg</td>
<td>STUK</td>
<td></td>
</tr>
<tr>
<td>AMLODIPINE ALSMALEAAT</td>
<td>50mg</td>
<td>STUK</td>
</tr>
<tr>
<td>ATENOLOL 100mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATENOLOL 50mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATENOLOL CHLOORTALIDON 100mg, 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BARNIDIPINE CAPS</td>
<td>10mg</td>
<td>STUK</td>
</tr>
<tr>
<td>BARNIDIPINE CAPS 20mg</td>
<td>STUK</td>
<td></td>
</tr>
<tr>
<td>BISOPROLOL FUMARATE</td>
<td>2.5mg</td>
<td>STUK</td>
</tr>
<tr>
<td>BISOPROLOL FUMARATE 10mg</td>
<td>STUK</td>
<td></td>
</tr>
<tr>
<td>BISOPROLOL FUMARATE 5mg</td>
<td>STUK</td>
<td></td>
</tr>
<tr>
<td>CANDESARTAN CILEXETIL 16mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAPTOPRIL 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARVEDILOL 10mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARVEDILOL 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARVEDILOL 5mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHLOORTALIDONE 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENALAPRIL WATERSTOFMALEAAT</td>
<td>10mg</td>
<td>STUK</td>
</tr>
<tr>
<td>ENALAPRIL WATERSTOFMALEAAT 20mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENALAPRIL 5mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENALAPRIL HYDROCHLOOR</td>
<td>12.5mg</td>
<td>STUK</td>
</tr>
<tr>
<td>1 FOSINOPRIL Natrium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HYDROCHLOORTHIAZIDETABL 12.5mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HYDROCHLOORTHIAZIDETABL 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HYDROCHLOORTHIAZIDETABL 50mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRBESARTAN 150mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRBESARTAN 300mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LISINOPRIL 10mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LISINOPRIL 5mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOSARTAN KALIUM 50mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLSUCCINAATTABLERETARD</td>
<td>60mg</td>
<td>STUK</td>
</tr>
<tr>
<td>METOPROLOLSUCCINAATTABLERETARD 30mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLSUCCINAATTABLERETARD 15mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLSUCCINAATTABL 95mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLSUCCINAATTABL 190mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLTARRAATTABL 100mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLTARRAATTABL 12.5mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLTARRAATTABL 50mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METOPROLOLTARRAATZEPIL 50mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEBIVOLOLTAB 5mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIFEDIPINERETARDTABL 20mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIFEDIPINERETARDTABL 60mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERINDOPRILBUMINE 8mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERINDOPRILBUMINETABL 2mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERINDOPRILBUMINETABL 4mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERINDOPRILINAPAMIDETABL 1, 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROPRANOLOLHCL 80mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROPRANOLOLHCTLTABL 10mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROPRANOLOLHCTLTABL 40mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUINAPRILALSHCL 10mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAMIPRIL 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAMIPRIL 5mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TELMISARANTABL 40mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIAMTEREEN 25mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIAMTEREEN 50mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALSARTAN 80mg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALSARTAN 160mg</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9.4 Appendix D: Work-up Data Collection Workflow

Figure 9.8: Workflow graph of work-up data collection
9.5 Appendix E: Biochemical Screening Reference Values

The following reference values were collected from the Erasmus MC quality information system.

<table>
<thead>
<tr>
<th>Test name</th>
<th>Material</th>
<th>Variable name</th>
<th>Min</th>
<th>Max</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortisol</td>
<td>Blood</td>
<td>COR Cortisol</td>
<td>0</td>
<td>850</td>
<td>IF 1mg-dexa old</td>
</tr>
<tr>
<td>Cortisol</td>
<td>Blood</td>
<td>COR Cortisol</td>
<td>5</td>
<td>133</td>
<td>IF 1mg-dexa new</td>
</tr>
<tr>
<td>Normetanephrine-to-Creatinine Ratio</td>
<td>Urine</td>
<td>NFKR Norm/Kreat</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normetanephrine Old</td>
<td>Urine</td>
<td>NM Normeta(oud)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 Hour Total Normetanephrine</td>
<td>Urine</td>
<td>NMD Normeta/24u</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normetanephrine</td>
<td>Urine</td>
<td>NMET Normetanefr.</td>
<td>75</td>
<td>325</td>
<td></td>
</tr>
<tr>
<td>Normetanephrine-to-Creatinine Ratio Old</td>
<td>Urine</td>
<td>NMKR Norm/KR(oud)</td>
<td>60</td>
<td>260</td>
<td></td>
</tr>
<tr>
<td>Metanephrine-to-Creatinine Ratio</td>
<td>Urine</td>
<td>MFKR Meta/Kreat</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metanephrine Old</td>
<td>Urine</td>
<td>MN Metanef(oud)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 Hour Total Metanephrine</td>
<td>Urine</td>
<td>MND Meta/24u</td>
<td>0</td>
<td>1.51</td>
<td></td>
</tr>
<tr>
<td>Metanephrine</td>
<td>Urine</td>
<td>MNE Metanefrine</td>
<td>40</td>
<td>171</td>
<td></td>
</tr>
<tr>
<td>Metanephrine-to-Creatinine Ratio Old</td>
<td>Urine</td>
<td>MNKR Meta/KR(oud)</td>
<td>35</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Aldosterone</td>
<td>Blood</td>
<td>ALDS Aldosteron</td>
<td>50</td>
<td>250</td>
<td></td>
</tr>
</tbody>
</table>
9.6 Appendix F: Code Base

9.6.1 Data Collection

% DATA UNDERSTANDING - Data Collection

% This script imports all data sources from excel and converts them to
% datasets

% INPUT: pat_data, labzis (2000 till 2015), measurements, blood_type,
% medication, icd9, icd10, dbc

% OUTPUT: patd, blood, meas, med, icd9, icd10, dbc, lab

% (c) J.B.R. Visser, September 2015.

%% Import flag for lab test data, warning TAKES LONG
import_labzis = 0;

%% Set root folder for import
cd(matlabroot)
cd('/Users/jobvisser/Desktop/ML/data') % Data location
cd('/Users/jobvisser/Dropbox/TUe/Graduation/3 Thesis/MATLAB') % Scripts location

%% Import data from Excel
patd = dataset('XlSFile', 'pat_data.xlsx');
blood = dataset('XlSFile', 'blood_type.xlsx');
meas = dataset('XlSFile', 'measurements.xlsx');
med = dataset('XlSFile', 'medication.xlsx');
icd9 = dataset('XlSFile', 'icd9.xlsx');
icd10 = dataset('XlSFile', 'icd10.xlsx');
dbc = dataset('XlSFile', 'dbc.xlsx');
wu_cl = dataset('XLSFile','ds_wu.xlsx');

if import_labzis == 1
    ds_ld0007 = dataset('XlSFile', 'labzis00-07.xlsx');
ds_ld0007.Properties.VarNames = {'pid','material','date','val','unit','test'};
ds_ld08 = dataset('XlSFile', 'labzis08.xlsx');
ds_ld08.Properties.VarNames = {'pid','material','date','val', 'unit', 'test'};
ds_ld09 = dataset('XlSFile', 'labzis09.xlsx');
ds_ld09.Properties.VarNames = {'pid','material','date','val', 'unit', 'test'};
ds_ld10 = dataset('XlSFile', 'labzis10.xlsx');
ds_ld10.Properties.VarNames = {'pid','material','date','val', 'unit', 'test'};
ds_ld11 = dataset('XlSFile', 'labzis11.xlsx');
ds_ld11.Var3 = [];% Delete empty column in export file
ds_ld11.Properties.VarNames = {'pid','material','date','val', 'unit', 'test'};
ds_ld11.pid = str2double(ds_ld11.pid);
ds_ld12 = dataset('XlSFile', 'labzis12.xlsx');
ds_ld12.Var3 = [];% Delete empty column in export file
ds_ld12.Properties.VarNames = {'pid','material','date','val', 'unit', 'test'};
ds_ld12.pid = str2double(ds_ld12.pid);
ds_ld13 = dataset('XlSFile', 'labzis13.xlsx');
ds_ld13.Var3 = [];% Delete empty column in export file
ds_ld13.Properties.VarNames = {'pid','material','date','val', 'unit', 'test'};
ds_ld14 = dataset('XlSFile', 'labzis14.xlsx');
ds_ld14.Var3 = [];% Delete empty column in export file
ds_ld14.Properties.VarNames = {'pid','material','date','val', 'unit', 'test'};
% Merge labzis data from all years
lab = [ds_ld0007; ds_ld08; ds_ld09; ds_ld10; ds_ld11; ...
       ds_ld12; ds_ld13; ds_ld14];
clear ds_ld0007 ds_ld08 ds_ld09 ds_ld10 ds_ld11 ds_ld12 ds_ld13 ds_ld14
end

% Rename dataset variables
patd.Properties.VarNames = {'pid','gender','dob','mar_stat','mult_births',...
   'dead', 'dod', 'street','no','no_suppl', 'postal_code','city','city2'};
blood.Properties.VarNames = {'pid','material','date','val', 'test'};
meas.Properties.VarNames = {'pid','val','unit', 'descr', 'time', 'date'};
icd9.Properties.VarNames = {'pid','date_disch','code','descr', 'dept'};
icd10.Properties.VarNames = {'pid','date','code','descr', 'dept_code', 'dept'};
dbc.Properties.VarNames = {'pid','dept', 'date_start','code'};
wu_cl.Properties.VarNames = {'pid','find_class','find_refdept',...
  'find_date', 'find_type','advise', 'find_concl', 'size',...
  'loc','prev_rep', 'fu_unrel','hu','washout', 'gr','report','gr_sign',...
  'adenoma','rep_discr', 'hypertension', 'diag', 'adr1', 'hist_malign',...
  'fu1_appl', 'fu1_date', 'fu1_type', 'fu2_appl', 'fu2_date', 'fu2_type',...
  'fu3_appl', 'fu3_date', 'fu3_type', 'fu4_appl', 'fu4_date', 'fu4_type', ...
  'fu5_appl', 'fu5_date', 'fu5_type', 'fu6_appl', 'fu6_date', 'fu6_type',...
  'fu7_appl', 'fu7_date', 'fu7_type'};

%% Convert dates to datetime type, make sure to set the excel date format to 'number'
% Patient data
patd.dob = datetime(patd.dob, 'ConvertFrom', 'excel');
patd.dod = datetime(patd.dod, 'ConvertFrom', 'excel');
% Blood type data
blood.date = datetime(blood.date, 'ConvertFrom', 'excel');
% Measurements data
meas.date = datetime(meas.date, 'ConvertFrom', 'excel');
% ICD data
icd9.date_disch = datetime(icd9.date_disch, 'ConvertFrom', 'excel');
icd10.date = datetime(icd10.date, 'ConvertFrom', 'excel');
dbc.date_start = datetime(dbc.date_start, 'ConvertFrom', 'excel');
% Lab test data
lab.date = datetime(lab.date, 'ConvertFrom', 'excel');
% Medication data
med.Creatiedatum = datetime(med.Creatiedatum, 'ConvertFrom', 'excel');
med.Startdatum = datetime(med.Startdatum, 'ConvertFrom', 'excel');
med.Stopdatum = datetime(str2double(med.Stopdatum), 'ConvertFrom', 'excel');
% Work-up data
wu_cl.find_date = datetime(wu_cl.find_date, 'ConvertFrom', 'excel');
wu_cl.fu1_date = str2double(wu_cl.fu1_date);
wu_cl.fu1_date = datetime(wu_cl.fu1_date, 'ConvertFrom', 'excel');
wu_cl.fu2_date = str2double(wu_cl.fu2_date);
wu_cl.fu2_date = datetime(wu_cl.fu2_date, 'ConvertFrom', 'excel');
wu_cl.fu3_date = str2double(wu_cl.fu3_date);
wu_cl.fu3_date = datetime(wu_cl.fu3_date, 'ConvertFrom', 'excel');
wu_cl.fu4_date = str2double(wu_cl.fu4_date);
9.6.2 Data Exploration

%%% DATA UNDERSTANDING - Data Exploration

%%% UNIVARIATE ANALYSIS

 Num: Descriptive Statistics (mean, stdev, min, max)
 Histogram
 Box plot
 %
 Cat: Tabulation

%%% BIVARIATE ANALYSIS:

 Num: Grouped scatter-plot matrix
 Correlation heatmap
 Scatter-histogram

% INPUT: ds_tr

% (c) J.B.R. Visser, October 2015.

%%% Prepare data for Data Exploration

 Categorical for POS class
 ds_cat = ds_tr(ds_tr.find_class=='POS',
   {'gender','dead','endo','aldos','...
    'meta_normeta_24','find_class','find_type','advise','find_concl','loc','...
    'prev_rep','fu_unrel','fu1','gr_sign','rep_discr','hypertension','...
    'hyp_med','hyp_med_pre','diag','adrl','hist_malign','death_6mon','clin_rel'});
 NamesC = ds_cat.Properties.VarNames;
 NamesC = strrep(NamesC, '_', '\_'); % Replace _ for \_ to avoid label errors

 Numerical for POS class
 ds_num = ds_tr(ds_tr.find_class=='POS',
   {'age','find_dead','size','log_size','...
    'endo_int','fu1_int','fu2_int','fu3_int'});
 NamesN = ds_num.Properties.VarNames;
 NamesN = strrep(NamesN, '_', '\_'); % Replace _ for \_ to avoid label errors

%%% Endocrine (biochemical screening) lab test data
X_end =
    double(ds_tr(ds_tr.find_class=='POS' & ds_tr.endo=='1',
        {'NMNormeta_oud_','NMKRNorm_KR_oud_','MNMetanef_oud_','MNKRMeta_KR_oud_'}));

ds_end = ds_tr(ds_tr.find_class=='POS' & ds_tr.endo=='1',
    {'NMNormeta_oud_','NMKRNorm_KR_oud_','MNMetanef_oud_','MNKRMeta_KR_oud_'}));

% Remove 4 patients with new endo lab tests
ds_end = ds_end(~ismissing(ds_end(:,1)),:);

NamesEND =
    {'NMNormeta_oud_','NMKRNorm_KR_oud_','MNMetanef_oud_','MNKRMeta_KR_oud_'};
NamesEND = strrep(NamesEND, '_','\_'); % Replace _ for \_ to avoid label errors

% Routine lab test data
X_lab = double(ds_tr(ds_tr.find_class=='POS',76:111)); %76:84 are tests
ds_lab = ds_tr(ds_tr.find_class=='POS',[76:111 128]);
NamesLAB = ds_tr.Properties.VarNames(76:111);
NamesLAB = strrep(NamesLAB, '_','\_'); % Replace _ for \_ to avoid label errors

%% UNIVARIATE - Categorical

%% tabulate find_class
a = tabulate(ds_tr.find_class);
saveTABLE(a,'find_class');
clear a

%% tabulate find_type
a = tabulate(DSm1.find_type);
saveTABLE(a,'find_type');
clear a

%% tabulate gender
a = tabulate(DSm1.gender);
saveTABLE(a,'gender');
clear a

%% tabulate endo
a = tabulate(DSm1.endo);
saveTABLE(a,'endo');
clear a

%% tabulate advise
a = tabulate(DSm1.advise);
saveTABLE(a,'advise');
clear a
%% tabulate find_concl
a = tabulate(DSm1.find_concl);
saveTABLE(a,'find_concl');
clear a
%% tabulate loc
a = tabulate(DSm1.loc);
saveTABLE(a,'loc');
clear a
%% tabulate fu_unrel
a = tabulate(DSm1.fu_unrel);
saveTABLE(a,'fu_unrel');
clear a
%% tabulate fu_type
a = tabulate(DSm1.fu1_type);
saveTABLE(a,'fu_type');
clear a
%% tabulate gr_sign
a = tabulate(DSm1.gr_sign);
saveTABLE(a,'gr_sign');
clear a
%% tabulate rep_discr
a = tabulate(DSm1.rep_discr);
saveTABLE(a,'rep_discr');
clear a
%% tabulate hypertension
a = tabulate(DSm1.hypertension);
saveTABLE(a,'hypertension');
clear a
%% tabulate hyp_med
a = tabulate(DSm1.hyp_med);
saveTABLE(a,'hyp_med');
clear a
%% tabulate hyp_med_pre
a = tabulate(DSm1.hyp_med_pre);
saveTABLE(a,'hyp_med_pre');
clear a
%% tabulate aldos
a = tabulate(DSm1.aldos);
saveTABLE(a,'aldos');
clear a
% tabulate meta_normeta_24
a = tabulate(DSm1.meta_normeta_24);
saveTABLE(a,'meta_normeta_24');
clear a
% tabulate hist_malign
a = tabulate(DSm1.hist_malign);
saveTABLE(a,'hist_malign');
clear a
% tabulate dead
a = tabulate(DSm1.dead);
saveTABLE(a,'dead');
clear a
% tabulate death_unrel
a = tabulate(DSm1.death_6mon);
saveTABLE(a,'death_6mon');
clear a
% tabulate diag
a = tabulate(DSm1.diag);
saveTABLE(a,'diag');
clear a
% tabulate adrl
a = tabulate(DSm1.adrl);
saveTABLE(a,'adrl');
clear a
% tabulate clin_rel
a = tabulate(DSm1.clin_rel);
saveTABLE(a,'clin_rel');
clear a
% UNIVARIATE - Numeric

% Descriptive statistics of relevant numeric variables
stats_num = zeros(max(size(NamesN)), 1);
i=1;
% For every test in test_names
for ii=1:max(size(NamesN))
    var_array = double(DSm1(:,ii));
    % Sum the number of NaN values per variable
test_stats(i,1) = sum(~isnan(var_array));
% Calculate the percentage of NaN values

% Calculate the mean (ignoring missing) per variable

% Calculate the std (ignoring missing) per variable

% Calculate the min (ignoring missing) per variable

% Calculate the max (ignoring missing) per variable

test_stats(i,6) = nanmax(var_array);
i = i+1;

end

clear vars i ii var_array miss

%% Put descriptive statistics in dataset: stats_num
stats_num = mat2dataset(test_stats, 'VarNames', {'Count', 'Percent'....

% Create LaTeX table

stats_num = dataset2table(stats_num);

input.data = stats_num;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Descriptive statistics of continuous variables';
input.tableLabel = 'descr_cont';
input.dataFormat = {'%.1f'}; % one digit precision
latex(input,'descr_cont')
clear test_stats input

%% Descriptive statistics of endo_int grouped by find_class

stats_endo_int = grpstats(ds_tr, 'find_class', {'mean', 'min', 'max'},

'DataVars','endo_int');

stats_fu_int = grpstats(ds_tr, 'find_class', {'mean', 'min', 'max'},

'DataVars','fu1_int');

hyp_xtab = crosstab(DSm1.hypertension, DSm1.hyp_med);
hyp_pre_xtab = crosstab(DSm1.hypertension, DSm1.hyp_med_pre);

%% Create Latex table for endo - aldos cross-tabulation

[a, ~, ~, labels] = crosstab(DSm1.endo, DSm1.aldos);
labels
a = array2table(a,'VariableNames',
    {'noAldos','Aldos'},'RowNames',{noEndo,'Endo'});
input.data = a;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Cross-tabulation of aldos and endo';
input.tableLabel = 'cross_aldos_endo';
input.dataFormat = {'%.1f'}; % one digit precision
latex(input,'cross_aldos_endo')
clear a input

%% Create Latex table for hyp_med - hyp_med_pre cross-tabulation
[a, ~, ~, labels] = crosstab(DSm1.hyp_med, DSm1.hyp_med_pre);
labels
a = array2table(a,'VariableNames',{hypMedPre,'nohypMedPre'},'RowNames',{hypMed,'nohypMed'});
input.data = a;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Cross-tabulation of hypMed and hypMedPre';
input.tableLabel = 'cross_hyp_med';
input.dataFormat = {'%.1f'}; % one digit precision
latex(input,'cross_hyp_med')
clear a input

%% Create Latex table for endo - aldos cross-tabulation
[a, ~, ~, labels] = crosstab(DSm1.endo, DSm1.fu1);
labels
a = array2table(a,'VariableNames',
    {'noFu1','Fu1'},'RowNames',{noEndo,'Endo'});
input.data = a;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Cross-tabulation of endo and fu1';
input.tableLabel = 'cross_endo_fu1';
input.dataFormat = {'%.1f'};  % one digit precision
latex(input,'cross_endo_fu1')
clear a input

%% Create Latex table for hyp_med - aldos cross-tabulation
[a, ~, ~, labels] = crosstab(DSm2.hyp_med, DSm2.aldos);
labels
a = array2table(a,'VariableNames',
    {'noAldos','Aldos'},'RowNames',{'HypMed','noHypMed'});
input.data = a;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Cross-tabulation of hypMed and Aldos for dataset 2';
input.tableLabel = 'cross_hyp_med_aldos_m2';
input.dataFormat = {'%.1f'};  % one digit precision
latex(input,'cross_hyp_med_aldos_m2')
clear a input

%% UNIVARIATE - Visualization

%% Box plot per numeric variable
% Size and Age can be grouped since their range is comparable
figure(1)
boxplot(double(DSm1(:,{'size','age'})),'labels',{'size','age'})
saveEPS( figure(1), 'box_size_age.eps' )
% Death after finding in months
figure(2)
boxplot(double(DSm1(:,{'find_dead'})),'labels',{'find_dead'})
saveEPS( figure(2), 'box_find_dead.eps' )
% Days after subsequent follow-ups in days
figure(3)
boxplot(double(DSm1(:,{'endo_int','fu1_int','fu2_int','fu3_int'})),'labels',{'endo_int','fu1_int','fu2_int','fu3_int'})
saveEPS( figure(3), 'box_fu_int.eps' )
% Follow-up 1 interval in days grouped by find_class
figure(44)
boxplot(double(ds_tr.fu1_int), ds_tr.find_class)
saveEPS( figure(44), 'box_fu_int_gr.eps' )
% Endo interval in days grouped on find_class
figure(45)
boxplot(double(ds_tr.endo_int), ds_tr.find_class)
saveEPS( figure(45), 'box_endo_int_gr.eps' )

%%% Histograms

% Size with lognormal distribution fit
figure(4)
histfit(DSm1.size,50,'lognormal')
saveEPS( figure(4), 'hist_size.eps' )

% Age with normal distribution fit
figure(5)
histfit(DSm1.age,25,'normal')
saveEPS( figure(5), 'hist_age.eps' )

%%% Normal probability plots

% Lognormal probability plot for size
figure(6)
probplot( 'lognormal',DSm1.size)
saveEPS( figure(6), 'prob_size.eps' )

% Normal probability plot for age
figure(7)
probplot( 'normal',DSm1.age)
saveEPS( figure(7), 'prob_age.eps' )

%%% BIVARIATE - Visualization

%%% Correlation Heatmap - continues variables
X = double(ds_end);  % For endo lab tests
X(:,end+1) = DSm2.age;  % Add Age variable
X(:,end+1) = DSm2.log_size;  % Add log_size variable
Names = {'NMNormeta_oud_','NMKRNorm_KR_oud_','MNMetanef_oud_','...
         'MNKRMeta_KR_oud_','age','log_size'};

% Impute missing values with KNNimpute: euclidean distance K=6
X = knnimpute(X,6,'Distance','euclidean');

% Create correlation matrix
corrmat = corr(X,X);

%%% Generate correlation heatmap
figure(8)
n = size(X, 2);
imagesc(corrmat);
ax = gca;
ax.XTickLabelRotation = 25; % Set X-label rotation
set(gca,'clim',[-1,1])
set(gca,'XTick',1:n);
set(gca,'YTick',1:n);
set(gca,'XTickLabel',Names);
set(gca,'YTickLabel',Names);
axis([0 n+1 0 n+1]);
grid;
colorbar;
caxis([0,1])

%% Save heatmap to LaTeX folder
clear corrmat X Names ax n
saveEPS( figure(8), 'corr_end.eps' )

%% Correlation Matrix - lab tests
% For routine lab tests
X = horzcat(X_lab(:,1:9));
Names = NamesLAB(1:9);
% Impute missing values with KNNimpute: euclidean distance K=6
X = knnimpute(X,6,'Distance','euclidean');
% Create correlation matrix
corrmat = corr(X,X);
% Generate correlation heatmap
figure(8)
n = size(X, 2);
imagesc(corrmat);
ax = gca;
ax.XTickLabelRotation = 25;
set(gca,'clim',[-1,1])
set(gca,'XTick',1:n);
set(gca,'YTick',1:n);
set(gca,'XTickLabel',Names);
set(gca,'YTickLabel',Names);
axis([0 n+1 0 n+1]);
grid;
colorbar;
caxis([0,1])

%% Save heatmap to LaTeX folder
clear corrmat X Names ax n
saveEPS( figure(8), 'corr_lab.eps' )

%% Scatter plot matrix
% They are used to identify relationships among variables. Grouped versions
% of these plots use different plotting symbols to indicate group membership.

% Prepare data for scatter plots
X = X_end;  % Endocrine lab tests
Y = Ym2;    % Target variable clin_rel for dataset M2
Names = NamesEND;

%% Routine Lab Tests
% figure(11)
% plotmatrix(X,'ob')  % Useless for big set
% saveEPS( figure(11), 'scat_end.eps' )

%% Grouped Scatter Matrix: ENDO
X = X_end;  % Biochemical screening lab tests
Y = DSm2.clin_rel;
Names = NamesEND;

figure(9)
gplotmatrix(X,X, Y, 'gr','oX',8,'on',[],Names,Names);
% saveEPS( figure(9), 'scat_end.eps')
clear X Y Names

%% Grouped Scatter Matrix: LAB
X = X_lab(:,[1:9]);  % Routine lab tests
Y = ds_lab.clin_rel;
Names = NamesLAB([1:9]);

figure(9)
gplotmatrix(X,X, Y, 'bg','xo',8,'on',[],Names,Names);
ax = gca;
ax.XTickLabelRotation = 25;  % Rotate X-labels
% saveEPS( figure(9), 'scat_lab.eps' )
clear X Y Names

%% Scatter Histogram: Grouped by clin_rel
figure(10)
scatterhist(DSm1.size,DSm1.age,'Group',Ym1,'Marker','o+','MarkerSize',[4,8])
saveEPS( figure(10), 'scat_hist.eps' )

9.6.3 Data Quality

%% DATA UNDERSTANDING - Data Quality
% Quality assessment or missing value analysis
% INPUT: Xm1, X_lab, X_end, X_meas
% (c) J.B.R. Visser, November 2015.

%% Prepare data
X = Xm1; % Dataset M1
Xl = X_lab(:,1:9); % Routine lab tests
Xe = X_end(:,1:9); % Biochemical screening tests
Xm = X_meas; % Routine measurements

%% Inspect work-up and patient data for missing values.
figure(1);
    barh(sum(isnan(X),1)/size(X,1));
    h = gca;
    h.YTick = 1:numel(NamesM1);
    h.YTickLabel = NamesM1(1:end);
    ylabel 'Predictor';
    xlabel 'Fraction of missing values';
    saveEPS( figure(1), 'miss_wu.eps' )

%% Inspect endo data for missing values.
figure(2);
    barh(sum(isnan(Xe),1)/size(Xe,1));
    h = gca;
    Names = strrep(NamesEND, '_', '\_'); % Replace _ for \_ to avoid label errors
h.YTick = 1:numel(Names(1:11));
h.YTickLabel = Names(1:11);
ylabel 'Predictor';
xlabel 'Fraction of missing values';
saveEPS( figure(2), 'miss_end.eps' )
clear Names

%%% Inspect lab data for missing values.
figure(3);
   barh(sum(isnan(Xl),1)/size(Xl,1));
   h = gca;
   h.YTick = 1:numel(NamesLAB(1:9));
   h.YTickLabel = NamesLAB(1:9);
ylabel 'Predictor';
xlabel 'Fraction of missing values';
saveEPS( figure(3), 'miss_lab.eps' )

%%% Inspect measurements data for missing values.
figure(4);
   barh(sum(isnan(Xm),1)/size(Xm,1));
   h = gca;
   h.YTick = 1:numel(NamesMEAS);
   h.YTickLabel = NamesMEAS(1:end);
ylabel 'Predictor';
xlabel 'Fraction of missing values';
saveEPS( figure(4), 'miss_meas.eps' )

% Clear variables
clear X Xl Xe Xm
9.6.4 Data Cleaning

Data Cleaning: Lab Test Data

% DATA PREPARATION - Data Cleaning

% This script cleans the labzis data for further analysis

% INPUT: lab
% OUTPUT: lab_cl

% (c) J.B.R. Visser, September 2015.

%% Prepare data
trans_vals = 0; % Flag to clean lab test values
lab_cl = lab; % Cleaned lab test dataset

if trans_vals == 1
    % Transform the value variable
    value = cellstr(lab_cl.val);
    num_val = zeros(max(size(value)),1);
    % For every value in lab
    for i=1:max(size(lab_cl(:,4)))
        str_val = char(value(i,1));
        % from , to . decimal seperator
        str_val = strrep(str_val, ',', '.');
        % Remove < or > chars
        str_val = strrep(str_val, '<', ' ');
        str_val = strrep(str_val, '>', ' ');
        % Construct double for the left over values
        num_val(i,1) = str2double(str_val);
    end
    % Add column with cleaned vals to ds
    lab_cl.val = num_val;
else
end

% make material and tests nominal
lab_cl.test = nominal(lab_cl.test);
lab_cl.material = nominal(lab_cl.material);

% Unstack dataset to wide format (patient per row)
pat_test = lab_cl(:,{'pid', 'val', 'test'});
pat_test.test = cellstr(pat_test.test);
pat_test = unstack(pat_test, 'val', 'test', 'AggregationFun', @nanmean);

Data Cleaning: Measurements Data

% DATA PREPARATION - Data Cleaning
%
% This script cleans and transforms the measurements dataset
%
% INPUT:     meas
% OUTPUT:    pat_meas, meas_cl, meas_wide
%
% (c) J.B.R. Visser, September 2015.

% Concatenate time and date columns
conc_date = 0;
if conc_date==1
    % Concatenate time and date columns
    time = datetime(meas.time, 'InputFormat','HH:mm');
    date = meas.date;
    date_full = date + timeofday(time);
    meas.date = date_full;
    meas.time = [];
    clear date_full date time
else
end

% Convert type to nominal and rename categories
meas.type = nominal(meas.descr);
meas = sortrows(meas,'type','ascend');
meas.type = nominal(meas.descr, {'adp', 'asp', 'length', 'weight', 'puls', 'temp'});

% Delete missing observations
ix = ismissing(meas);
meas_cl = meas(~any(ix,2), {'pid', 'val', 'type', 'date'});
clear ix

% Prepare dataset for unstacking
meas_cl.type = cellstr(meas_cl.type);
meas_cl.day = day(meas_cl.date);

% Delete date and day variable
meas_wide = meas_cl(:, {'pid', 'val', 'type'});
meas_wide = unstack(meas_wide, 'val', 'type', 'AggregationFun', @nanmean);

% Number of unique patients in meas_wide dataset
pat_meas = unique(meas_wide.pid);

% Calculate mean measurements per patient in meas_wide
meas_avr = zeros(max(size(pat_meas)), 6);
for n=1:max(size(pat_meas))
    patX = pat_meas(n);
    meas_mat = meas_wide(meas_wide.pid==patX, {'adp', 'asp',
               'length', 'weight', 'puls', 'temp'});
    meas_avr(n,1) = nanmean(meas_mat.adp);
    meas_avr(n,2) = nanmean(meas_mat.asp);
    meas_avr(n,3) = nanmean(meas_mat.length);
    meas_avr(n,4) = nanmean(meas_mat.weight);
    meas_avr(n,5) = nanmean(meas_mat.puls);
    meas_avr(n,6) = nanmean(meas_mat.temp);
end
clear patX meas_mat ix n

meas_wide = mat2dataset(meas_avr, 'VarNames', {'adp', 'asp', 'length',
               'weight', 'puls', 'temp'});
meas_wide.pid = pat_meas;
meas_wide = meas_wide(:, [7 1:6]);
clear meas_avr conc_date
9.6.5 Data Integration

% DATA PREPARATION - Data Integration
% This script integrates all datasets
% INPUT: patd_cl, wu_cl, labEND_wide, lab20_wide, meas_wide
% OUTPUT: ds_int, dsPAT, dsEND, dsLAB, dsMEAS
% (c) J.B.R. Visser, September 2015.

%% Import data
dsWU = wu_cl(:,{'pid','find_class','find_refdept','find_date','find_type','advise','find_concl'
  'size','loc','prev_rep','fu_unrel','gr','gr_sign','rep_discr',...
  'hypertension','diag','adrl','hist_malign','fu1_date','fu2_date',...
  'fu3_date','fu4_date','fu5_date','fu6_date','fu7_date'});
dsPAT = patd_cl;
dsEND = labEND_wide;
dsLAB = lab20_wide;
dsMEAS = meas_wide(:,{'pid','adp','asp','temp'});

% Detect difference between datasets
[DIFF_w ] = setdiff(dsWU.pid,dsPAT.pid); % [6405000] --> Delete
[DIFF_p ] = setdiff(dsPAT.pid,dsWU.pid); % none --> Missing
% Remove patient 6405000 from dataset.
% dsWU(dsWU.pid==6405000,:) = [];

%% Join work-up and patient datasets
ds0 = join(dsPAT, dsWU, 'key','pid','Type','outer','MergeKeys',true);
%% Join ds0 and biochemical screening (endo) datasets
ds1 = join(ds0, dsEND, 'key','pid','Type','outer','MergeKeys',true);
%% Join ds0 and routine lab test datasets
ds2 = join(ds1, dsLAB, 'key','pid','Type','outer','MergeKeys',true);
%% Join ds0 and measurements datasets
ds3 = join(ds2, dsMEAS, 'key','pid','Type','outer','MergeKeys',true);

%% Rename to integrated dataset ds_int
9.6.6  Data Transformation

Data Transformation: Lab Test Data

% DATA PREPARATION - Data Transformation Lab Test Data
%
% This script generates descriptive statistics to explore the lab test
% data. After that data was unstacked, transformed and new features were
% constructed.
%
% INPUT: pat_test, test_SEL, test_END, lab_cl
% OUTPUT: lab_stats, lab20_stats, labEND_stats, lab20_wide, labEND_wide,
% labEND_preFind, labEND_postFind, labEND_noFind
%
% (c) J.B.R. Visser, September 2015.

%% Descriptive Statistics per test: no_NaN, %_NaN, mean, std, min, max
% Retrieve test_names
test_names = pat_test.Properties.VarNames;
test_names = test_names(:, [2:end]);
test_stats = zeros(max(size(test_names)), 6);
i=1;
for ii=2:max(size(test_names)+1)
    var_array = double(pat_test(:,ii));
    % Sum the number of NaN values per variable
    test_stats(i,1) = sum(isnan(var_array));
    % Calculate the percentage of NaN values
    test_stats(i,2) = test_stats(i,1)/max(size(pat_test));
    % Calculate the mean (ignoring missing) per variable
    test_stats(i,3) = nanmean(var_array);
    % Calculate the std (ignoring missing) per variable
    test_stats(i,4) = nanstd(var_array);
    % Calculate the min (ignoring missing) per variable

test_stats(i,5) = nanmin(var_array);

% Calculate the max (ignoring missing) per variable
test_stats(i,6) = nanmax(var_array);
i = i+1;
end

clear vars i ii var_array

% Put descriptive statistics in dataset: lab_stats
lab_stats = mat2dataset(test_stats, 'VarNames', {'missing', 'perc_miss',
   'mean', 'std', 'min', 'max'}, 'ObsNames', test_names);

clear test_stats test_names

%% Missing Value Analysis
% Construct inclusion criterion for routine lab tests
% IF missing values < threshold THEN include in dataset.
threshold = 0.05;
for i=1:max(size(lab_stats))
   if lab_stats.perc_miss(i) < threshold
      lab_stats.incl(i) = 1;
   else
      lab_stats.incl(i) = 0;
   end
end

clear i

% Construct nominal variable with all selected tests
test_SEL = lab_stats(lab_stats.incl==1,:);
test_SEL = test_SEL.Properties.ObsNames;
test_SEL = cellstr(matlab.lang.makeValidName(cellstr(test_SEL)));
test_SEL = nominal(test_SEL);

%% Select relevant variables: pid, date, val, test
lab_red = lab_cl(:,{'pid', 'date', 'val', 'test'});

% Convert test names to MATLAB format
conv_names = 0;
if conv_names == 1
   lab_red.test = cellstr(lab_red.test);
   names = cellstr(matlab.lang.makeValidName(lab_red.test));
   lab_red.test = names;
else
end

clear var conv_names

%%% Construct lab20 containing all tests selected in MVA: routine lab tests
%%% Fill lab20 dataset with only records from tests in test_SEL
lab_red.test = nominal(lab_red.test);
lab_i = lab_red;
lab_i(:, :) = [];
lab20 = lab_i;
n = 1;
for i = 1:max(size(test_SEL))
    lab_i = lab_red(lab_red.test == test_SEL(i, 1), :);
    lab20 = [lab_i; lab20];
end
clear var n lab_i

%% Descriptive statistics for all individual tests
lab20_stats = grpstats(lab20, 'test', {'mean', 'std', 'min', 'max'}, 'DataVars', ...
                      {'val'});

%%% Create LaTeX table
lab20_stats_t = dataset2table(lab20_stats);
lab20_stats_t.test = [];
lab20_stats_t.Properties.VariableNames = {'TestCount', 'Mean', 'Std', 'Min', 'Max'};
lab20_stats_t.Properties.RowNames = strrep(lab20_stats_t.Properties.RowNames, ...
                                       '_', '_');
input.data = lab20_stats_t;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Descriptive statistics of routine lab tests';
input.tableLabel = 'descr_lab';
input.dataFormat = {'%.1f'}; % one digit precision
latex(input, 'descr_lab')

clear input lab20_stats_t

%%% Construct lab_END containing all tests indicating ENDO visit
%%% Export test names to identify relevant tests indicating ENDO
% Import biochemical screening lab tests
test_END = table2cell(readtable('test_rel.csv'));

% Fill labEND dataset with only records from tests in test_END
lab_i = lab_red;
lab_i(:, :) = [];
labEND = lab_i;
for i = 1:max(size(test_END))
    lab_i = lab_red(lab_red.test == test_END(i, 1), :);
    labEND = [lab_i; labEND];
end

clear n lab_i i lab_red

%% Calculate descriptive statistics statistics
labEND_stats = grpstats(labEND,
    'test', {'mean', 'std', 'min', 'max'},
    'DataVars', ['val']);

%% Create LaTeX table
labEND_stats_t = dataset2table(labEND_stats);
labEND_stats_t.test = [];
labEND_stats_t.Properties.VariableNames =
    {'TestCount', 'Mean', 'Std', 'Min', 'Max'};
labEND_stats_t.Properties.RowNames = strrep(labEND_stats_t.Properties.RowNames,
    '_', '\_');
input.data = labEND_stats_t;

input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Descriptive statistics of biochemical screening tests';
input.tableLabel = 'descr_endo';
input.dataFormat = {'%.1f'}; % one digit precision
latex(input, 'descr_endo')
clear input labEND_stats_t

%%% Construct lab20_cell (cell with vals per test) and lab20_expl (dataset)
%%% Create cell array with all values per selected test per cell
lab20_cell = cell(1,max(size(test_SEL))); for i=1:max(size(test_SEL))
    lab_i = double(lab20(lab20.test==test_SEL(i,1), 'val'));
    lab20_cell{i} = lab_i;
end

clear var n lab_i

%% Transpose cell matrix then unstack to dataset: patient per row
lab20_cell = lab20_cell';
lab20_cell = cell2dataset(lab20_cell, 'VarNames', {'vals'}, 'ObsNames', cellstr(test_SEL));
lab20.test = cellstr(lab20.test);
lab20_temp = unstack(lab20, 'val','test','AggregationFun',@nanmean);

%% Calculate means per test per patient in lab20_wide
pat_sel = unique(lab20_temp.pid);

%% Create a matrix for every feature that needs to be extracted
avr_sel = zeros(max(size(pat_sel)),max(size(test_SEL))); % mean
min_sel = zeros(max(size(pat_sel)),max(size(test_SEL))); % min
max_sel = zeros(max(size(pat_sel)),max(size(test_SEL))); % max
std_sel = zeros(max(size(pat_sel)),max(size(test_SEL))); % stdev
for n=1:max(size(pat_sel))
    patX = pat_sel(n);
    test_array = lab20_temp(lab20_temp.pid==patX, [3:end]);
    for i=1:max(size(test_SEL))
        avr_sel(n,i) = nanmean(eval(sprintf('meas_mat.%s', char(test_SEL(i,1)))));
        min_sel(n,i) = nanmin(eval(sprintf('meas_mat.%s', char(test_SEL(i,1)))));
        max_sel(n,i) = nanmax(eval(sprintf('meas_mat.%s', char(test_SEL(i,1)))));
        std_sel(n,i) = nanstd(eval(sprintf('meas_mat.%s', char(test_SEL(i,1)))));
    end
end
% Concatenate all test names in on big vertical list

```matlab
test_SEL_compl = vertcat(cellstr(test_SEL), strcat('STD_', cellstr(test_SEL)),...
strcat('MIN_', cellstr(test_SEL)), strcat('MAX_', cellstr(test_SEL)));
all_sel = horzcat(avr_sel, std_sel, min_sel, max_sel); % Concatenate all features
```

% Construct final unstacked 'wide' dataset

```matlab
lab20_wide = mat2dataset(all_sel, 'VarNames', test_SEL_compl);
lab20_wide.pid = pat_sel;
clear patX meas_mat ix n lab20_temp avr_sel min_sel max_sel std_sel all_sel
```

% Construct labEND_cell (cell with vals per test) and labEND_expl (dataset)

```matlab
% Create cell array with all values per selected test per cell
labEND_cell = cell(1, max(size(test_END)));
for i=1:max(size(test_END))
    lab_i = double(labEND(labEND.test==test_END(i,1), {'pid', 'val'}));
    labEND_cell{i} = lab_i;
end
clear var n lab_i i
```

% Transpose cell matrix then unstack to dataset

```matlab
labEND_cell = labEND_cell';
labEND_cell = dataset(labEND_cell, 'VarNames', {'vals'}, 'ObsNames', cellstr(test_END));
labEND.test = cellstr(labEND.test);
```

% Create temporary unstacked dataset with unique patient/date combination

```matlab
labEND_temp = unstack(labEND, 'val', 'test', 'AggregationFun', @nanmean);
```

% Create labEND_postFind where biochemical screening was done after the
adrenal mass was found

```matlab
labEND_preFind = labEND_temp;
labEND_preFind(:, :) = [];
labEND_postFind = labEND_preFind;
labEND_noFind = labEND_preFind;
lab_c = labEND_preFind;
lab_o = labEND_postFind;
pat_end_temp = unique(labEND_temp.pid);
```
find_date = dsWU(:,{'pid','find_date'});
for ii=1:size(pat_end_temp,1)
    patX = pat_end_temp(ii);
    labX = labEND_temp(labEND_temp.pid==patX, :);
    dateX = find_date(find_date.pid==patX, 'find_date');
    if isnat(dateX.find_date) == 0
        % If test has been done before finding put in preFind
        lab_c = labX(labX.date < dateX.find_date, :);
        labEND_preFind = [lab_c ; labEND_preFind];
        % If test has been done after finding put in postFind
        lab_o = labX(labX.date > dateX.find_date, :);
        labEND_postFind = [lab_o ; labEND_postFind];
    else
        % If patient has no find_date (find_class: NEG, EXT)
        labEND_noFind = [labX ; labEND_noFind];
    end
    lab_c = [];
    lab_o = [];
    labX = [];
end
clear ii i lab_c lab_o
labEND_postFind = [labEND_postFind ; labEND_noFind];
pat_end = unique(labEND_postFind.pid);

% Feature Construction for endo lab tests for patients in labEND_postFind
% Create a matrix for every feature that needs to be extracted
avr_end = zeros(max(size(pat_end)),max(size(test_END))); % mean
min_end = zeros(max(size(pat_end)),max(size(test_END))); % min
max_end = zeros(max(size(pat_end)),max(size(test_END))); % max
std_end = zeros(max(size(pat_end)),max(size(test_END))); % stdev
for n=1:max(size(pat_end)) % For every patient that received screening
    patX = pat_end(n);
    test_array = labEND_postFind(labEND_postFind.pid==patX, [3:end]);
    for i=1:max(size(test_END))
        avr_end(n,i) = nanmean(eval(sprintf('test_array.%s', char(test_END(i,1)))));
        min_end(n,i) = nanmin(eval(sprintf('test_array.%s', char(test_END(i,1)))));
    end
end
max_end(n,i) = nanmax(eval(sprintf('test_array.%s', char(test_END(i,1)))));

std_end(n,i) = nanstd(eval(sprintf('test_array.%s', char(test_END(i,1)))));

end

% Concatenate all test names in on big vertical list

test_END_compl = vertcat(cellstr(test_END), strcat('STD_',cellstr(test_END)),
...
strcat('MIN_',cellstr(test_END)), strcat('MAX_',cellstr(test_END)));

all_end = horzcat(avr_end, std_end, min_end, max_end); % Concatenate features

labEND_wide = mat2dataset(all_end,'VarNames',cellstr(test_END_compl));

labEND_wide.pid = pat_end; % Insert pid into dataset

labEND_wide = labEND_wide(:,[45 1:44]); % Set pid as first column

clear patX test_array ix n labEND_temp i avr_end std_end min_end max_end all_end

---

Data Transformation: Medication Data

% DATA PREPARATION - Data Transformation: Medication Data

% This script transforms the medication data and extracts anti-hypertensive drugs used by patients.

% INPUT: med, meds_hyp.csv

% OUTPUT: med_cl, meds_HYP, medHYP_stats, med_hyp_preFind, med_hyp_postFind, med_hyp_Find

% (c) J.B.R. Visser, September 2015.

% Prepare data: select relevant variables

med_cl = med(:,[3 7 8 9 12 13 15 16 17]);

med_cl.Properties.VarNames = {'pid','start_date','stop_date','type','descr',
'dose','freq','per_dose','unit'};
% Derive time of treatment
med_cl.med_time = day(med_cl.stop_date - med_cl.start_date);

% Transform numerical data from str2double
med_cl.dose = str2double(med_cl.dose);

%%% Construct med_hyp containing all meds indicating HYPERTENSION
% Export test names to identify relevant meds indicating hypertension
med_names = dataset(unique(med_cl.descr));
med_names = cellstr(matlab.lang.makeValidName(cellstr(med_names)));
export(dataset(med_names), 'file', 'meds.csv');

% Import hypertension meds: Selected using medical dictionary
meds_HYP = table2cell(readtable('meds_hyp.csv'));
meds_HYP = nominal(meds_HYP);

% Transform med descriptions to matlab name format
med_cl.descr = cellstr(matlab.lang.makeValidName(cellstr(med_cl.descr)));
med_cl.descr = nominal(med_cl.descr);

%%% Fill med_hyp dataset with only records from tests in med_HYP to indicate
% hypertension
med_i = med_cl;
med_i(:, :) = [];
med_hyp = med_i;
for i = 1:max(size(meds_HYP))
    med_i = med_cl(med_cl.descr==meds_HYP(i,1), :);
    med_hyp = [med_i; med_hyp];
end
clear n med_i i

% Generate descriptive statistics for anti-hypertensive drugs
medHYP_stats = grpstats(med_hyp,'descr', {'mean','std','min','max'},
                {'DataVars'},
                {'dose'});

%%% Construct hyp_med variable for patients that used anti-hypertensive
% medications at the moment of finding
med_hyp.preFind = med_hyp;
med_hyp.preFind(:, :) = [];
med_hyp_Find = med_hyp_preFind;
med_p = med_hyp_preFind;
med_f = med_hyp_Find;
pat_hyp = unique(med_hyp.pid);
find_date = dsWU(:,{'pid','find_date'});
for ii=1:size(pat_hyp,1)
    patX = pat_hyp(ii);
    medX = med_hyp(med_hyp.pid==patX,:);
    dateX = find_date(find_date.pid==patX,'find_date');
    if isnat(dateX.find_date) == 0
        % If hypertension meds were used during finding put in med_hyp_Find
        med_f = medX(medX.start_date < dateX.find_date & ...
                medX.stop_date > dateX.find_date,:);
        med_hyp_Find = [med_f; med_hyp_Find];
    % If hypertension meds were used before finding put in med_hyp_preFind
    med_p = medX(medX.start_date < dateX.find_date & ...
                medX.stop_date < dateX.find_date,:);
        med_hyp_preFind = [med_p; med_hyp_preFind];
    else
        % if patient has no find_date (find_class: NEG, EXT)
    end
    med_f = [];
    med_p = [];
end
clear patX medX dateX

% Patients having hypertension
pat_hyp_med_Find = unique(med_hyp_Find.pid);    % To evaluate meds used
pat_hyp_med_preFind = unique(med_hyp_preFind.pid); % To evaluate meds used
meds_HYP_f = unique(med_hyp_Find.descr);
meds_HYP_pre = unique(med_hyp_preFind.descr);
hyp_med = mat2dataset(pat_hyp_med_Find,'VarNames','pid');
hyp_med.hyp_med(:,1) = 1;
hyp_med_preFind = mat2dataset(pat_hyp_med_preFind,'VarNames','pid');
hyp_med_preFind.hyp_med_pre(:,1) = 1;
clear pat_hyp_med
Data Transformation: Patient Data

% DATA PREPARATION - Data Transformation: Patient Data
% This script cleans patient data
% INPUT: patd
% OUTPUT: patd_cl
%
% (c) J.B.R. Visser, September 2015.

%% Drop location variables
patd_cl = patd(:,{'pid', 'gender', 'dob', 'mar_stat', 'mult_births',...
    'dead', 'dod'});

% Dead flag variable
patd_cl.dead = nominal(patd_cl.dead);
patd_cl.mult_births = nominal(patd_cl.mult_births);
patd_cl.gender = nominal(patd_cl.gender);

% Convert marital status into nominal variable
patd_cl.marital = nominal(patd_cl.mar_stat);
patd_cl.marital = droplevels(patd_cl.marital, {'-', 'ONB - Onbekend'});
patd_cl = sortrows(patd_cl,'marital','ascend');
patd_cl.marital = nominal(patd_cl.marital,
    {'married','divorced','single','widow'});
patd_cl.mar_stat = [];

clear varclass t ans patd
9.6.7 Feature Construction

% DATA TRANSFORMATION - Feature Construction

% This script performs feature construction of the following features:
% - age
% - find_dead
% - death_6mon
% - meta_normeta_24
% - aldos
% - endo
% - date_endo
% - endo_int
% - fu1_int
% - fu2_int
% - fu3_int
% - clin_rel

% INPUT: ds_int, hyp_med, pat_end, dsEND, labEND_postFind
% OUTPUT: ds_tr

% (c) J.B.R. Visser, October 2015.

%% Prepare data: integrated dataset ds_int

ds = ds_int;

%% Constrict age variable at the time of the finding

ds.age = year(ds.find_date-ds.dob);

% If there's no finding date, take age at date: 01-01-2011 (halfway ds time horizon)

time = datetime('01-01-2011','InputFormat','dd-MM-yyyy');
i = isnan(double(ds.age));
for i=1:length(ds)
    if ia(i)==1
        ds.age(i) = year(time - ds.dob(i));
    else
        end
end
clear i time ia

%% Construct the find_dead variable: number of days after the finding that
% the patient died
ds.find_dead = day(ds.dod - ds.find_date);

%% Construct death_6mon variable if patient died 6 mo (182 days) after
% find_date
for i=1:max(size(ds))
    if ds.find_dead(i) < 182
        ds.death_6mon(i) = 1;
    else
        ds.death_6mon(i) = 0;
    end
end
clear i

%% Construct hyp_med (hypertension medication) variable
ds = join(ds, hyp_med,'key','pid','Type','outer','MergeKeys',true);
ds = join(ds, hyp_med_preFind,'key','pid','Type','outer','MergeKeys',true);

%% Construct ENDO_visit variable: meta - normeta - meta/kr - normeta/kr
endo_visit = mat2dataset(pat_end,'VarNames', 'pid');
for i=1:max(size(pat_end))
    present = isnan(double(dsEND(i,1:end)));
    % check if metanefrine or metanefrine_old is present
    % so at least one of the two should have isnan=0
    if sum(present(1,[8 10])) <= 1
        meta = 1;
    else
        meta = 0;
    end
    % check if normetanefrine or normetanefrine_old is present
    if sum(present(1,[3 5])) <= 1
        normeta = 1;
    else
        normeta=0;
    end
    % check if meta/kr or meta/kr_old is present

if sum(present(1,[7 11])) <= 1
    meta.kr = 1;
else
    meta.kr = 0;
end

% check if normeta/kr or normeta/kr_old is present
if sum(present(1,[2 6])) <= 1
    normeta.kr = 1;
else
    normeta.kr = 0;
end

% check if normeta/24 or meta/24 is present
if sum(present(1,[4 9])) <= 1
    endo_visit.meta.normeta.24(i) = 1;
else
    endo_visit.meta.normeta.24(i) = 0;
end

% check if patient had aldosterone tested
if sum(present(1,12)) == 1
    endo_visit.aldos(i) = 1;
else
    endo_visit.aldos(i) = 0;
end

% check if patient had full endo lab test
if meta==1 && normeta==1 && meta.kr==1 && normeta.kr ==1
    endo_visit.endo(i) = 1; % Indicates full visit
else
    endo_visit.endo(i) = 0;
end

% Join datasets ds and date_endo
ds = join(ds, endo_visit, 'key','pid','Type','outer','MergeKeys',true);

clear avr_end present meta normeta meta.kr normeta.kr i

% Contruct date_endo based on the metanefrine date closest to the find_date
% Use the filtered dataset labEND_postFind
pats_meta_old = unique(labEND_postFind(~isnan(labEND_postFind.MNMetanef_oud_),'pid'));
pats_meta_new = unique(labEND_postFind(~isnan(labEND_postFind.MNEMetanefrine),'pid'));
pats_meta = double([pats_meta_old; pats_meta_new]);
pats_endo_visit = double(endo_visit(endo_visit.endo==1,'pid'));

% Dataset with all metanefrine lab tests

% retrieve date from first visit endo in time (first time metanefrine)
date_endo = labEND_postFind(:,{'pid','date'});
date_endo(:, :) = [];

for n=1:max(size(pats_endo_visit))
    patX = pats_endo_visit(n);
    patX_date = labEND_postFind(labEND_postFind.pid==patX,{'pid','date'});
    patX_date = sortrows(patX_date,'date','ascend'); %Sort dates ascending
    if isempty(patX_date) == 0
        patX_date = patX_date(1,{'pid','date'}); %Pick the first date in vector
        date_endo = [date_endo; patX_date]; %Concatenate datasets
    else
    end
end

% Join datasets ds and date_endo
ds = join(ds, date_endo,'key','pid','Type','outer','MergeKeys',true);
ds.date_endo = ds.date;
ds.date = [];

clear n patX patX_date pats_meta_old pats_meta_new pats_meta lab_meta date_endo

% Constructs the follow-up interval variables
ds.endo_int = datenum(ds.date_endo-ds.find_date); % days between finding and endo

ds.fu1_int = datenum(ds.fu1_date - ds.find_date); % days between endo and fu1
ds.fu2_int = datenum(ds.fu2_date - ds.fu1_date); % days between endo and fu2
ds.fu3_int = datenum(ds.fu3_date - ds.fu2_date); % days between endo and fu3

% Construct the clin_rel variable: relevant diagnosis, adrenalectomy, gr_sign

n=max(size(ds));
for ii=1:n
    if strcmp(ds(ii,'diag').diag,'ACC')||strcmp(ds(ii,'diag').diag,'CS')||... 
        strcmp(ds(ii,'diag').diag,'SCS')||strcmp(ds(ii,'diag').diag,'PA')||... 
        strcmp(ds(ii,'diag').diag,'FEO')||double(ds(ii,'gr_sign'))==1||... 
        double(ds(ii,'adr1'))==1
        ds.clin_rel(ii) = 1;
    else
        ds.clin_rel(ii) = 0;
    end
end

%% Convert to nominal and recode <undefined> to relevant label
ds.find_class = nominal(ds.find_class);
ds.find_type = nominal(ds.find_type);
ds.find_refdept = nominal(ds.find_refdept);
ds.loc = nominal(ds.loc);
    ds.loc(isundefined(ds.loc)) = 'Unknown';  % Unknown = location unknown
    ds.diag = nominal(ds.diag);
    ds.diag(isundefined(ds.diag)) = 'Unknown';  % Unknown = not relevant
    ds.rep_discr = nominal(ds.rep_discr);
    ds.rep_discr(isundefined(ds.rep_discr)) = '0';  % 0 = No discrepancy
    ds.fu_unrel = nominal(ds.fu_unrel);
    ds.adrl = nominal(ds.adrl);
    ds.gr_sign = nominal(ds.gr_sign);
    ds.find_concl = nominal(ds.find_concl);
    ds.fu_unrel = nominal(ds.fu_unrel);
    ds.prev_rep = nominal(ds.prev_rep);
    ds.hypertension = nominal(ds.hypertension);
    ds.hyp_med = nominal(ds.hyp_med);
    ds.hyp_med_pre = nominal(ds.hyp_med_pre);
    ds.hist_malign = nominal(ds.hist_malign);
    ds.advise = nominal(ds.advise);
    ds.death_6mon = nominal(ds.death_6mon);
    ds.aldos = nominal(ds.aldos);
    ds.meta_normeta_24 = nominal(ds.meta_normeta_24);
    ds.endo = nominal(ds.endo);
    ds.clin_rel = nominal(ds.clin_rel);
%% Replace missing values in nominal variables with 0
n=max(size(ds));
for ii=1:n
    if isundefined(ds.adrl(ii)) == 1
        ds.adrl(ii) = '0';
    else end
    if isundefined(ds.advise(ii)) == 1
        ds.advise(ii) = '0';
    else end
    if isundefined(ds.find_concl(ii)) == 1
        ds.find_concl(ii) = '0';
    else end
    if isundefined(ds.fu_unrel(ii)) == 1
        ds.fu_unrel(ii) = '0';
    else end
    if isnat(ds.fu1_date(ii)) == 1
        ds.fu1(ii) = '0';
    else
        ds.fu1(ii) = '1';
    end
    if isundefined(ds.hypertension(ii)) == 1
        ds.hypertension(ii) = '0';
    else end
    if isundefined(ds.hyp_med(ii)) == 1
        ds.hyp_med(ii) = '0';
    else end
    if isundefined(ds.hyp_med_pre(ii)) == 1
        ds.hyp_med_pre(ii) = '0';
    else end
    if isundefined(ds.hist_malign(ii)) == 1
        ds.hist_malign(ii) = '0';
    else end
    if isundefined(ds.prev_rep(ii)) == 1
        ds.prev_rep(ii) = '0';
    else end
    if isundefined(ds.gr_sign(ii)) == 1
        ds.gr_sign(ii) = '0';
    else end
    if isundefined(ds.endo(ii)) == 1
        ds.endo(ii) = '0';
    else end
end
ds.endo(ii) = '0';
else end
if isundefined(ds.aldos(ii)) == 1
    ds.aldos(ii) = '0';
else end
if isundefined(ds.meta_normeta_24(ii)) == 1
    ds.meta_normeta_24(ii) = '0';
else end
end
clear ii n
ds_tr = ds;
clear ds

%% Feature Transformation
% Log transform the size variable
ds_tr.log_size = log(ds_tr.size);

% Transform the 'blanco' value from hist_malign to '0'
getlevels(ds_tr.hist_malign)
hist_malign = droplevels(ds_tr.hist_malign,'blanco');
hist_malign(isundefined(hist_malign)) = '0';
ds_tr.hist_malign = hist_malign;
clear hist_malign

% Transform the 'Bi' value from location to 'BI'
getlevels(ds_tr.loc)
ds_tr.loc = droplevels(ds_tr.loc,'Bi');
loc(isundefined(loc)) = '0';
ds_tr.hist_malign = hist_malign;
clear hist_malign
%% PROJECT STARTUP - Dataset Construction

% This script coordinates the data mining project from the point that a
% transformed dataset (ds_tr) has been created.

% (c) J.B.R. Visser, September 2015.

%% Set project parameters

cd(matlabroot)
cd(’/Users/jobvisser/Dropbox/TUe/Graduation/3 Thesis/MATLAB’)
set(0,’DefaultFigureWindowStyle’,’docked’) % Plot style: ’default’,’docked’

%% Prepare data for Model 1 (M1) at 0 months: n=643

% Dataset DSm1
DSm1 = ds_tr(ds_tr.find_class==’POS’,{’gender’,’hyp_med’,’hist_malign’,...
’loc’,’age’,’log_size’,’clin_rel’});
% Matrices Xm1 and Ym1
Xm1 = double(DSm1(:,1:end-1));
Ym1 = DSm1.clin_rel;
% Variable names in NamesM1
Names = DSm1.Properties.VarNames(1:end-1);
NamesM1 = strrep(Names, ’_’, ’\_’); % Replace _ for \_ to avoid LaTeX errors
CatVarsM1 = [1 2 3 4];
clear Names

%% Prepare data for Model 2 (M2) at t=6 months: n=111

% Dataset DSm2
DSm2 = ds_tr(ds_tr.find_class==’POS’&&ds_tr.endo==’1’,{’gender’,’hyp_med’,...
’aldos’,’hist_malign’,’loc’,’age’,’log_size’,’NMNormeta_oud_’,...
’NMKRNorm_KR_oud_’,’MNMetanef_oud_’,’MNKRMeta_KR_oud_’,’STD_NMNormeta_oud_’,...
’STD_NMKRNorm_KR_oud_’,’STD_MNMetanef_oud_’,’STD_MNKRMeta_KR_oud_’,...
’MIN_NMNormeta_oud_’,’MIN_NMKRNorm_KR_oud_’,’MIN_MNMetanef_oud_’,...
’MIN_MNKRMeta_KR_oud_’,’MAX_NMNormeta_oud_’,’MAX_NMKRNorm_KR_oud_’,...
’MAX_MNMetanef_oud_’,’MAX_MNKRMeta_KR_oud_’,’clin_rel’});
% Remove 4 patients with new endo lab tests
DSm2 = DSm2(~ismissing(DSm2(:,8)),:);
% Matrices Xm2 and Ym2
9.6.9 Feature Selection

Select features M1 Lasso Regularization

%% DATA PREPARATION - Feature Selection: Lasso Regularization M1
%
% Lasso Logistic regression for feature selection.
%
% (c) J.B.R. Visser, October 2015.
%
%% Create output structure for model results
fsLASSOm1 = struct('model',0,'fit',1,'model_fs',0,'fit_fs',1,'Names',0,...
'NamesFS',0,'X_imp',0,'Ypred',0);

%% Step 1: prepare data
% Impute missing values dataset M1
Xm1_imp = knnimpute(Xm1,6,'Distance','euclidean');

% Inspect histograms for simularity
figure(3)
subplot(1,2,1)
histogram(Xm1(:,end))
title('Original values of size')
subplot(1,2,2)
histogram(Xm1_imp(:,end))
title('Imputed values of size (euclidean K=6)')
% saveEPS(figure(3),'imp_size');

% Extract continuous predictors (lasso does not handle categorical predictors)
Xm1_imp = Xm1_imp(:,5:6);
% Normalize data
Xm1_imp = normalize(Xm1_imp);
fsLASSO1.X_imp = Xm1_imp;
fsLASSO1.Names = NamesM1(5:6);

%% Step 2: create a cross-validated fit
% Construct a cross-validated lasso regularization of a binomial classification model
% The function returns penalized maximum-likelihood fitted coefficients
[Bm1, FitInfoM1] = 
    lassoglm(Xm1_imp,Ym1, 'binomial', 'Link', 'logit', 'CV', 10, 'Alpha', 1, ...
        'PredictorNames', NamesM1(4:5), 'Standardize', true, 'NumLambda', 50)
fsLASSO1.model = Bm1;
fsLASSO1.fit = FitInfoM1;

%% Step 3: Examine the plots to find appropriate regularization
% Examine the CV-plot to see the effect of the Lambda regularization parameter
lassoPlot(Bm1,FitInfoM1, 'plottype', 'CV');
saveEPS( figure(1), 'lassoplot_ds1.eps ')

% The plot identifies the minimum-deviance point with a green circle and
% dashed line as a function of the regularization parameter Lambda.
% The blue circled point has minimum deviance plus no more than one standard deviation.

% The trace plot shows nonzero model coefficients as a function of the regularization
% As Lambda increases to the left, lassoglm sets various coefficients to zero,
% removing them from the model.
lassoPlot(Bm1,FitInfoM1, 'PlotType', 'Lambda', 'XScale', 'log', ...
    'PredictorNames', fsLASSO1.Names);
grid on
hlplot = get(gca, 'Children');
set(hlplot, 'LineWidth', 3)
legend( 'Location', 'Best')
saveEPS( figure(1), 'traceplot_ds1.eps' )
%% Find the number of nonzero model coefficients at the Lambda value with
%% minimum deviance plus one standard deviation point.

indx = FitInfoM1.IndexMinDeviance; % Column where the coefficients are located
B0 = Bm1(:,indx); % Retrieve coefficient of size variable

%% Step 4: Create a regularized model
%% The constant term B0 is in the FitInfo.Index1SE entry of the
%% FinInfo.Intercept vector.

cnst = FitInfoM1.Intercept(indx);
B1 = [cnst ; B0];
fsLASSO1.model = B1;

---

**Select features M1 Random Forest**

%% DATA PREPARATION - Feature Selection: Random Forest M1
%%
%% Determine Random Forest feature importance for selecting features.
%%
%% (c) J.B.R. Visser, October 2015.

%% Create a Treebagger ensemble: Estimate feature importance
%% Output structure for model
fsTBm1 = struct('model',0,'model_fs',0,'Names',0,'NamesFS',0,'Ximp',0,...
                'Xfs',0,'Y',0);

%% Step 1: prepare data
%% Impute missing values model 1: size
Xm1_imp = knnimpute(Xm1,6,'Distance','euclidean');
fsTBm1.Ximp = Xm1_imp;
fsTBm1.Names = NamesM1;
fsTBm1.Y = double(Ym1)-1;

%% Inspect histograms for similarity
figure(3)
subplot(1,2,1)
histogram(Xm1(:,end))
title('Original values of size')

subplot(1,2,1)
histogram(Xm1_imp(:,end))
title('Imputed values of size (corr dist)')

%% Step 2: Build forest
tic
rf_m1 = TreeBagger(300,Xm1_imp,Ym1,'Method','classification','OOBVarImp',...
    'On','MinLeafSize',1,'surrogate','off','CategoricalPredictors',...
    CatVarsM1); %'Cost',[0 50; 1 0]
toc
fsTBm1.model = rf_m1;

%% Inspect the OOB classification error
figure(1);
tic
plot(oobError(fsTBm1.model));
hlplot = get(gca, 'Children');
set(hlplot, 'LineWidth', 2)
grid on;
xlabel('Number of trees');
ylabel('Out-of-Bag Classification Error');
saveEPS( figure(1), 'oob_error_ds1.eps' )
clear hlplot

%% Inspect the OOB classification error excluding in-bag observations
% Give no performance boost, so is not used
% outTBm1.model.DefaultYfit = 'MostPopular';
% figure(2)
% plot(oobError(outTBm1.model));
% xlabel('Number of Grown Trees');
% ylabel('Out-of-Bag Error Excluding In-Bag Observations');
% saveEPS( figure(2), 'oob_error_excl_ds1.eps' )

%% Predictor importance
figure(3)
bar(fsTBm1.model.OOBPermutedVarCountRaiseMargin);
ylabel 'Out-of-Bag Feature Importance';
set(gca(), 'XTick', 1:6)
set(gca(), 'XTickLabel', NamesM1)
saveEPS( figure(3), 'feat_imp_ds1.eps' )

% Select most important features at an arbitrary cut-off of 0.2
idxvar = find(fsTBm1.model.OOBPermutedVarCountRaiseMargin>=0.2); % most important features

% Create reduced dataset with most important features
Xfsm1 = Xm1_imp(:, idxvar);
fsTBm1.Xfs = Xfsm1;
NamesFSm1 = NamesM1(idxvar);
fsTBm1.NamesFS = NamesFSm1;
% Extract continuous predictors (lasso does not handle categorical predictors)
Xm2_imp = Xm2_imp(:,6:end);
% Normalize data
Xm2_imp = normalize(Xm2_imp);
fsLASSOm2.Ximp = Xm2_imp;
fsLASSOm2.Names = NamesM2(6:23);

%% Step 2: create a cross-validated fit
% Construct a cross-validated lasso regularization of a binomial classification model
% The function returns penalized maximum-likelihood fitted coefficients
tic
[Bm2, FitInfoM2] = lassoglm(Xm2_imp,Ym2,'binomial','Link','logit',...'
  'CV',10,'Alpha',1,'PredictorNames',NamesM2(6:23),...'
  'Standardize',true,'NumLambda',50);
toc
fsLASSOm2.model = Bm2;
fsLASSOm2.fit = FitInfoM2;

%% Using rf reduced feature set
[Bm2_rf, FitInfoM2_rf] = lassoglm(fsTBm2.Xfs,Ym2,'binomial','Link','logit',...'
  'CV',10,'Alpha',1, ...'
  'PredictorNames',fsTBm2.NamesFS,'Standardize',true,'NumLambda',50);

%% Step 3: Examine the plots to find appropriate regularization
% Examine the CV-plot to see the effect of the Lambda regularization parameter
lassoPlot(Bm2,FitInfoM2,'plottype','CV');
saveEPS(figure(1), 'lassoplot_ds2.eps')

%% The trace plot - dataset 2
% As Lambda increases to the left, lassoglm sets various coefficients to zero, % removing them from the model.
lassoPlot(Bm2,FitInfoM2,'PlotType','Lambda','XScale','log',...'
  'PredictorNames',NamesM2([6:23]));
grid on
hlplot = get(gca, 'Children');
set(hlplot, 'LineWidth', 3)
legend('Location','Best')
saveEPS( figure(1), 'traceplot_ds2.epso')

%%% The trace plot - on reduced TB features
% As Lambda increases to the left, lassoglm sets various coefficients to zero,
% removing them from the model.
% lassoPlot(Bm2_rf,FitInfoM2_rf,'PlotType','Lambda','XScale','log',...
%  'PredictorNames',fsTBm2.NamesFS);
% grid on
% hlplot = get(gca, 'Children');
% set(hlplot, 'LineWidth', 3)
% % set(gca, 'Units', 'Normalized', 'Position', [.2 .5 .5 .35])
% legend('Location','Best')
% %
% saveEPS( figure(1), 'traceplot_ds2_rf.epso')

%%% Find the number of nonzero model coefficients at the Lambda value with
%%% minimum deviance plus one standard deviation point.

indx = FitInfoM2.IndexMinDeviance; % Column where the coefficients are located
B0 = Bm2(:,indx);  % Retrieve coefficient of size variable

%%% Step 4: Create a regularized model
% The constant term B0 is in the FitInfo.Index1SE entry of the
% FinInfo.Intercept vector.

cnst = FitInfoM2.Intercept(indx);
B1 = [cnst ; B0];
fsLASSOm2.model = B1;
Select features M2 Random Forest

```matlab
%% DATA PREPARATION - Feature Selection: Random Forest M2
%
%% Determine Random Forest feature importance for selecting features.

% (c) J.B.R. Visser, October 2015.

%% Create a Treebagger ensemble: Estimate feature importance
% Output structure for model
fsTBm2 = struct('model',0,'model_fs',0,'Names',0,'NamesFS',0,'Ximp',0,...
    'Xfs',0,'Y',0);

%% Step 1: prepare data
% Impute missing values model 1
Xm2_imp = knnimpute(Xm2,6,'Distance', 'euclidean');
fsTBm2.Ximp = Xm2_imp;
fsTBm2.Y = double(Ym2)-1;
fsTBm2.Names = NamesM2;
% Inspect histograms for similarity
figure(21)
subplot(1,2,1)
histogram(Xm2(:,7))
title('Original values of size')
subplot(1,2,2)
histogram(Xm2_imp(:,7))
title('Imputed values of size (corr dist)')
% saveEPS(figure(21),'imp_size');

%% Step 2: Build forest
tic
rf_m2 = TreeBagger(200,Xm2_imp,Ym2,'Method','classification','OOBVarImp',...'
    'On','MinLeafSize',1,'surrogate','off','CategoricalPredictors',...'
    CatVarsM2); % ', 'Cost',[0 5; 1 0]
toc
fsTBm2.model = rf_m2;

%% Inspect the OOB classification error
figure(1);
```

182
tic
plot(oobError(fsTBm2.model));
hlplot = get(gca, 'Children');
set(hlplot, 'LineWidth', 2)
grid on;
xlabel('Number of trees');
ylabel('Out-of-Bag Classification Error');
saveEPS( figure(1), 'oob_error_ds2.eps' )
clear hlplot

%%% Predictor importance
figure(3)
bar(fsTBm2.model.OOBPermutedVarCountRaiseMargin);
ylabel 'Out-of-Bag Feature Importance';
set(gca(),'XTick',1:22)
set(gca(),'XTickLabel',NamesM2(1:22))
ax = gca;
ax.XTickLabelRotation = 35;
saveEPS( figure(3), 'feat_imp_ds2.eps' )
clear ax

%%% Select most important features at an arbitrary cut-off of 0.2
idxvar = find(fsTBm2.model.OOBPermutedVarCountRaiseMargin>=0.2); % most important features

%%% Create reduced dataset with most important features
Xfsm2 = Xm2_imp(:,idxvar);
fsTBm2.Xfs = Xfsm2;
NamesFSm2 = NamesM2(idxvar);
fsTBm2.NamesFS = NamesFSm2;
9.6.10 M1 Model Building

```matlab
%% MODELING - Model Building M1

%% Build 3 classification models:
% - Logistic Regression
% - Random Forest
% - Fuzzy Inference System (Adaptive Neuro-Fuzzy Inference System)

% (c) J.B.R. Visser, October 2015.

%% Prepare data
Xo = fsTBm1.Xfs; % Original data Xo
Yo = fsTBm1.Y; % Original target Yo
Names = fsTBm1.NamesFS;
X_ori = [Xo,Yo];

% Create holdout sample of original data for validation
rng(1957); % For reproducibility
CVm1_ori = cvpartition(X_ori(:,end),'holdout',0.15);
Xval_ori = X_ori(CVm1_ori.test,:);
Yval = Xval_ori(:,end); % Original holdout target for validation
Xval = Xval_ori(:,1:end-1); % Original holdout data for validation
X_ori = X_ori(CVm1_ori.training,);

% Perform SMOTE
att = [1 0 1]; %
% Add 500 synthetic cases for approx. 50-50 class distribution
X_smote = SMOTE(X_ori,X_ori(:,end),500,0.3,att);
Ys = X_smote(:,end); % Oversampled data Ys
Xs = X_smote(:,1:end-1); % Oversampled data Xs

% Create cross-validation partition
CVm1 = cvpartition(Ys,'kfold',10);
%inM1 = struct('cv_val',0,'cv',0,)
% Create matrix for metrics on SMOTE and VAL
metricsM1 = zeros(6,7);
metricsM1_V = zeros(6,7);
grouporder = [1 0]; % For confusion matrix
```
save_figs = 0; % Save flag for figures

%% Clear vars
clear mdl ii Xo Yo X_ori Names Xval_ori CVm1_ori CVm1 Xval Yval att X_smote...
Xs Ys Xtest Xtrain Ytest Ytrain

%% Logistic Regression

%% Output structure for model
outLOGm1 = struct('ii',0, 'out',0, 'kappa',0, 'fpr',0, 'tpr',0, 'metrics',0, ...
 'confmat',0, 'outV',0, 'kappaV',0, 'fprV',0, 'tprV',0, 'metricsV',0, ...
 'confmatV',0, 'model',0);
outLOGm1 = repmat(outLOGm1,10,1); % 10 folds
thresholdM1 = 0.5;

%% Build Logistic Regression Model
for ii=1:CVm1.NumTestSets
  Xtrain = Xs(CVm1.training(ii),:);
  Ytrain = Ys(CVm1.training(ii),:);
  Xtest = Xs(CVm1.test(ii),:);
  Ytest = Ys(CVm1.test(ii),:);

  % Train Logistic Regression model
  tic
  mdl = fitglm(Xtrain, Ytrain, 'link', 'logit', 'distr','binomial',...
               'ResponseVar', 'clin_rel','Categorical',[1],'PredictorVars',Names);
  toc
  outLOGm1(ii).model = mdl;
  outLOGm1(ii).ii = ii;

  % Test and Validate Logistic Regression model
  outLOGm1(ii).out = predict(mdl, Xtest);
  outLOGm1(ii).outV = predict(mdl, Xval);

  % Evaluate predicted Y values based in thresholdM1
  YpredT = double(outLOGm1(ii).out >= thresholdM1);
  YpredV = double(outLOGm1(ii).outV >= thresholdM1);
  conf_logT = confusionmat(Ytest, YpredT,'order',grouporder);
  conf_logV = confusionmat(Yval, YpredV,'order',grouporder);
% Save metrics for test group (SMOTE)
[outLOGm1(ii).metrics, outLOGm1(ii).kappa, outLOGm1(ii).fpr,
  outLOGm1(ii).tpr,...
  ] = calc_metrics(conf_logT', Ytest, outLOGm1(ii).out);
outLOGm1(ii).confmat = conf_logT';
clear conf_logT YpredT
% Save metrics for validation group (ORI holdout)
[outLOGm1(ii).metricsV, outLOGm1(ii).kappaV, outLOGm1(ii).fprV,
  outLOGm1(ii).tprV,...
  ] = calc_metrics(conf_logV', Yval, outLOGm1(ii).outV);
outLOGm1(ii).confmatV = conf_logV';
clear conf_logV YpredV
end

%% Evaluate model on oversampled dataset
close all
[metricsM1(1,:), metricsM1(2,:)] = eval_mdl(outLOGm1, 10, 1, 'test','LR M1 SMOTE');
if save_figs==1
  saveEPS(figure(1), 'roc_lg_m1_test');
saveEPS(figure(2), 'auk_lg_m1_test');
else end

%% Evaluate model on original validation dataset
close all
[metricsM1_V(1,:), metricsM1_V(2,:)] = eval_mdl(outLOGm1, 10, 1, 'val','LR M1 ORI');
if save_figs==1
  saveEPS(figure(1), 'roc_lg_m1_val');
saveEPS(figure(2), 'auk_lg_m1_val');
else end

%% Random Forest

% Output structure for model
outTBm1 = struct('ii',0,'out',0,'kappa',0,'fpr',0,'tpr',0,'metrics',0,...
  'confmat',0,'outV',0, 'kappaV',0,'fprV',0,'tprV',0,'metricsV',0,...
  'confmatV',0,'model',0);
outTBm1 = repmat(outTBm1,10,1); % 10 folds
thresholdM1 = 0.5;

%% Build Random Forest
for ii=1:CVm1.NumTestSets
    Xtrain = Xs(CVm1.training(ii),:);
    Ytrain = Ys(CVm1.training(ii),:);
    Xtest = Xs(CVm1.test(ii),:);
    Ytest = Ys(CVm1.test(ii),:);

    % Train Random Forest
    tic
    mdl = TreeBagger(200,Xtrain,Ytrain,'Method','classification','OOBVarImp',...
    'Off','MinLeafSize',1,'surrogate','off','CategoricalPredictors',[],...
    'oobpred', 'on'); % , 'Cost',[0 5; 1 0]
    toc
    outTBm1(ii).model = mdl;
    outTBm1(ii).ii = ii;
    % Test and Validate model
    % Score generated by each tree is the probability of this observation
    % originating from this class computed as the fraction of observations
    % of this class in a tree leaf. The scores are averaged over the forest
    [~, scoreT] = predict(mdl, Xtest);
    outTBm1(ii).out = scoreT(:,2);
    [~, scoreV] = predict(mdl, Xval);
    outTBm1(ii).outV = scoreV(:,2);
    clear scoreT scoreV
    % Evaluate predicted Y values based in thresholdM2
    YpredT = double(outTBm1(ii).out >= thresholdM1);
    YpredV = double(outTBm1(ii).outV >= thresholdM1);
    conf_logT = confusionmat(Ytest, YpredT,'order',grouporder);
    conf_logV = confusionmat(Yval, YpredV,'order',grouporder);
    % Save metrics for test group (SMOTE)
    [outTBm1(ii).metrics, outTBm1(ii).kappa, outTBm1(ii).fpr, ...
     outTBm1(ii).tpr,...
     [~,"] = calc_metrics(conf_logT', Ytest,outTBm1(ii).out);
    outTBm1(ii).confmat = conf_logT';
    clear conf_logT YpredT
    % Save metrics for validation group (ORI holdout)
[\text{outTBm1(ii).metricsV}, \text{outTBm1(ii).kappaV}, \text{outTBm1(ii).fprV},
\text{outTBm1(ii).tprV},...]
\text{= calc_metrics(\text{conf_logV'}, Yval, \text{outTBm1(ii).outV});}
\text{outTBm1(ii).confmatV = conf_logV';}
\text{clear conf_logV YpredV}
end

\text{%% Evaluate model on oversampled dataset}
\text{close all}
\text{[metricsM1(3,:), metricsM1(4,:)] = eval_mdl(outTBm1, 10, 1, 'test','RF M1 SMOTE');}
\text{if save_figs==1}
\text{saveEPS(figure(1),'roc_rf_m1_test');}
\text{saveEPS(figure(2),'auk_rf_m1_test');}
\text{else end}
\text{%% Evaluate model on original validation dataset}
\text{close all}
\text{[metricsM1_V(3,:), metricsM1_V(4,:)] = eval_mdl(outTBm1, 10, 1, 'val','RF M1 ORI');}
\text{if save_figs==1}
\text{saveEPS(figure(1),'roc_rf_m1_val');}
\text{saveEPS(figure(2),'auk_rf_m1_val');}
\text{else end}

\text{%% Fuzzy Inference System (ANFIS)}
\text{-----------------------------------------------}
\text{%% Output structure for model}
\text{outFISm1 = struct('\text{ii}',0,'\text{out}',0,'\text{kappa}',0,'\text{fpr}',0,'\text{tpr}',0,'\text{metrics}',0,...
'\text{confmat}',0,'\text{outV}',0, '\text{kappaV}',0,'\text{fprV}',0,'\text{tprV}',0,'\text{metricsV}',0,...
'\text{confmatV}',0,'\text{model}',0);}
\text{outFISm1 = repmat(outFISm1,10,10); % 10 folds}
\text{thresholdM1 = 0.5;}

\text{%% Build FIS}
\text{for c=2:10 % For all different numbers of cluters}
\text{for ii=1:CVm1.NumTestSets}
\text{Xtrain = Xs(CVm1.training(ii),:);}
\text{Ytrain = Ys(CVm1.training(ii),:);}
\text{Xtest = Xs(CVm1.test(ii),:);}
Ytest = Ys(CVm1.test(ii),:);

% Train FIS
tic
fis = genfis3(Xtrain,Ytrain,'sugeno',c);
[fis,~,~] = anfis([Xtrain, Ytrain], fis);
toc
outFISm1(ii,c).ii = ii;

% Test and Validate Logistic Regression model
outFISm1(ii,c).out = evalfis(Xtest,fis);
outFISm1(ii,c).outV = evalfis(Xval,fis);
outFISm1(ii,c).model = fis;

% Evaluate predicted Y values based in thresholdM2
YpredT = double(outFISm1(ii,c).out >= thresholdM1);
YpredV = double(outFISm1(ii,c).outV >= thresholdM1);
conf_logT = confusionmat(Ytest, YpredT,'order',grouporder);
conf_logV = confusionmat(Yval, YpredV,'order',grouporder);

% Save metrics for test group (SMOTE)
[metricsm1(ii,c).metrics, metricsm1(ii,c).kappa, metricsm1(ii,c).fpr,
 outFISm1(ii,c).tpr,...
 ,~] = calc_metrics(conf_logT', Ytest,outFISm1(ii,c).out);
outFISm1(ii,c).confmat = conf_logT';
clear conf_logT YpredT

% Save metrics for validation group (ORI holdout)
[metricsm1(ii,c).metricsV, metricsm1(ii,c).kappaV, metricsm1(ii,c).fprV,
 outFISm1(ii,c).tprV,...
 ,~] = calc_metrics(conf_logV', Yval, outFISm1(ii,c).outV);
outFISm1(ii,c).confmatV = conf_logV';
clear conf_logV YpredV
end
end

% Evaluate model on oversampled dataset
%close all
[metricsM1(5,:), metricsM1(6,:)] = eval_md1(outFISm1, 10, 2, 'test','FIS M1 SMOTE')
if save_figs==1
saveEPS(figure(1),'roc_fis_m1_test');
saveEPS(figure(2), 'auk_fis_m1_test');
else end

%% Evaluate model on original validation dataset
close all
[metricsM1_V(5,:), metricsM1_V(6,:)] = eval_mdl(outFISm1, 10, 2, 'val', 'FIS M1 ORI')
if save_figs==1
saveEPS(figure(1), 'roc_fis_m1_val');
saveEPS(figure(2), 'auk_fis_m1_val');
else end

%% Save scores on unbalanced validation set M1:
% Will be used for decision curve analysis
scoresM1_val = zeros(96,4);
scoresLOG = zeros(96,10);
scoresRF = zeros(96,10);
scoresFIS = zeros(96,10);
for ii=1:10
    scoresLOG(:,ii) = outLOGm1(ii).outV;
scoresRF(:,ii) = outTBm1(ii).outV;
scoresFIS(:,ii) = outFISm1(ii,2).outV;
end
scoresM1_val(:,1) = mean(scoresLOG,2);
scoresM1_val(:,2) = mean(scoresRF,2);
scoresM1_val(:,3) = mean(scoresFIS,2);
scoresM1_val(:,3) = normalize(scoresM1_val(:,3)); % ANFIS scores are normalized
scoresM1_val(:,4) = Yval;
scoresM1_val = mat2dataset(scoresM1_val, 'VarNames', {'LR', 'RF', 'FIS', 'clin_rel'});
saveCSV(scoresM1_val, 'scoresM1_val.csv')
clear scoresLOG scoresRF scoresFIS scoresM1_val
%% MODELING - Model Building M2

% Build 3 classification models:
%   - Logistic Regression
%   - Random Forest
%   - Fuzzy Inference System (Adaptive Neuro-Fuzzy Inference System)

% (c) J.B.R. Visser, October 2015.

%% Prepare data
Xo = fsTBm2.Xfs; % Original data Xo
Yo = fsTBm2.Y; % Original target Yo

% Log transform and normalize
{MAX_NMKNorm_KR_oud, MAX_MNKRMeta_KR_oud}
Xo(:,5:6) = log(Xo(:,5:6));
Names = fsTBm2.NamesFS;
X_ori = [Xo,Yo];

% Create holdout sample of original data for validation
rng(192); % For reproducibility
CVm2_ori = cvpartition(X_ori(:,end),'holdout',0.15);
Xval_ori = X_ori(CVm2_ori.test,:);
Yval = Xval_ori(:,end); % Original holdout target for validation
Xval = Xval_ori(:,1:end-1); % Original holdout data for validation
X_ori = X_ori(CVm2_ori.training,:);

% Perform SMOTE
att = [0 0 0 0 0 0 1]; %
% Add 80 synthetic cases for approx. 50-50 class distribution
X_smote = SMOTE(X_ori,X_ori(:,end),76,0.3,att);
Ys = X_smote(:,end); % Oversampled data Ys
Xs = X_smote(:,1:end-1); % Oversampled data Xs

% Create cross-validation patition
CVm2 = cvpartition(Ys,'kfold',10);

% Create matrix for metrics on SMOTE and VAL
metricsM2 = zeros(6,7);
metricsM2_V = zeros(6,7);
grouporder = [1 0]; % For confusion matrix
save_figs = 0; % Save flag for figures

%% Clear vars
clear mdl ii Xo Yo X_ori Names Xval_ori CVm2_ori CVm2 Xval Yval att X_smote...
Xs Ys Xtest Xtrain Ytest Ytrain

%% Output structure for model
outLOGm2 = struct('ii',0,'out',0,'kappa',0,'fpr',0,'tpr',0,'metrics',0,...
'confmat',0,'outV',0,'kappaV',0,'fprV',0,'tprV',0,'metricsV',0,...
'confmatV',0,'model',0);
outLOGm2 = repmat(outLOGm2,10,1); % 10 folds
thresholdM2 = 0.5;

%% Build Logistic Regression Model
for ii=1:CVm2.NumTestSets
    Xtrain = Xs(CVm2.training(ii),:);
    Ytrain = Ys(CVm2.training(ii),:);
    Xtest = Xs(CVm2.test(ii),:);
    Ytest = Ys(CVm2.test(ii),:);

    % Train Logistic Regression model
tic
    mdl = fitglm(Xtrain, Ytrain, 'link', 'logit', 'distr', 'binomial',...
        'ResponseVar', 'clin_rel', 'Categorical', [], 'PredictorVars', Names);
toc
    outLOGm2(ii).model = mdl;
    outLOGm2(ii).ii = ii;

    % Test and Validate Logistic Regression model
    outLOGm2(ii).out = predict(mdl, Xtest);
    outLOGm2(ii).outV = predict(mdl, Xval);
    % Evaluate predicted Y values based in thresholdM2
    YpredT = double(outLOGm2(ii).out >= thresholdM2);
YpredV = double(outLOGm2(ii).outV >= thresholdM2);
conf_logT = confusionmat(Ytest, YpredT,'order',grouporder);
conf_logV = confusionmat(Yval, YpredV,'order',grouporder);
% Save metrics for test group (SMOTE)
[outLOGm2(ii).metrics, outLOGm2(ii).kappa, outLOGm2(ii).fpr, 
outLOGm2(ii).tpr,...
~,~] = calc_metrics(conf_logT, Ytest,outLOGm2(ii).out);
clear conf_logT YpredT
% Save metrics for validation group (ORI holdout)
[outLOGm2(ii).metricsV, outLOGm2(ii).kappaV, outLOGm2(ii).fprV, 
outLOGm2(ii).tprV,...
~,~] = calc_metrics(conf_logV', Yval, outLOGm2(ii).outV);
outLOGm2(ii).confmatV = conf_logV';
clear conf_logV YpredV
end

%%% Evaluate model on oversampled dataset
close all
[metricsM2(1,:), metricsM2(2,:)] = eval_mdl(outLOGm2, 10, 1, 'test','LR M2 SMOTE');
if save_figs==1
saveEPS(figure(1),'roc_lg_m2_test');
saveEPS(figure(2),'auk_lg_m2_test');
else end
%%% Evaluate model on original validation dataset
close all
[metricsM2_V(1,:), metricsM2_V(2,:)] = eval_mdl(outLOGm2, 10, 1, 'val','LR M2 ORI');
if save_figs==1
saveEPS(figure(1),'roc_lg_m2_val');
saveEPS(figure(2),'auk_lg_m2_val');
else end

%%% Random Forest
%%% Output structure for model
outTBm2 = struct('ii',0,'out',0,'kappa',0,'fpr',0,'tpr',0,'metrics',0,...
  'confmat',0,'outV',0,'kappaV',0,'fprV',0,'tprV',0,'metricsV',0,...
  'confmatV',0,'model',0);
outTBm2 = repmat(outTBm2,10,1); % 10 folds
thresholdM2 = 0.5;

%% Build Random Forest
for ii=1:CVm2.NumTestSets
  Xtrain = Xs(CVm2.training(ii),:);
  Ytrain = Ys(CVm2.training(ii),:);
  Xtest = Xs(CVm2.test(ii),:);
  Ytest = Ys(CVm2.test(ii),:);
  
  % Train Random Forest
  tic
  mdl = TreeBagger(200,Xtrain,Ytrain,'Method','classification','OOBVarImp',...
    'Off','MinLeafSize',1,'surrogate','off','CategoricalPredictors',[],...
    'oobpred','on'); % ,'Cost',[0 5; 1 0]
  toc
  outTBm2(ii).model = mdl;
  outTBm2(ii).ii = ii;

  % Test and Validate model
  % Score generated by each tree is the provability of this obervation
  % originating from this class computed as the fraction of observations
  % of this class in a tree leaf. The scores are averaged over the forest
  [~, scoreT] = predict(mdl, Xtest);
  outTBm2(ii).out = scoreT(:,2);
  [~, scoreV] = predict(mdl, Xval);
  outTBm2(ii).outV = scoreV(:,2);
  clear scoreT scoreV

  % Evaluate predicted Y values based in thresholdM2
  YpredT = double(outTBm2(ii).out >= thresholdM2);
  YpredV = double(outTBm2(ii).outV >= thresholdM2);
  conf_logT = confusionmat(Ytest, YpredT,'order',grouporder);
  conf_logV = confusionmat(Yval, YpredV,'order',grouporder);

  % Save metrics for test group (SMOTE)
  [outTBm2(ii).metrics, outTBm2(ii).kappa, outTBm2(ii).fpr, 
  outTBm2(ii).tpr,...
    ',',] = calc_metrics(conf_logT', Ytest,outTBm2(ii).out);
outTBm2(ii).confmat = conf_logT;
clear conf_logT YpredT
% Save metrics for validation group (ORI holdout)
[outTBm2(ii).metricsV, outTBm2(ii).kappaV, outTBm2(ii).fprV,
  outTBm2(ii).tprV,...
~, ~] = calc_metrics(conf_logV', Yval, outTBm2(ii).outV);
outTBm2(ii).confmatV = conf_logV';
clear conf_logV YpredV
end

%% Evaluate model on oversampled dataset
close all
[metricsM2(3,:), metricsM2(4,:)] = eval_mdl(outTBm2, 10, 1, 'test', 'RF M2
SMOTE');
if save_figs==1
  saveEPS(figure(1), 'roc_rf_m2_test');
saveEPS(figure(2), 'auk_rf_m2_test');
else end

%% Evaluate model on original validation dataset
close all
[metricsM2_V(3,:), metricsM2_V(4,:)] = eval_mdl(outTBm2, 10, 1, 'val', 'RF M2
ORI');
if save_figs==1
  saveEPS(figure(1), 'roc_rf_m2_val');
saveEPS(figure(2), 'auk_rf_m2_val');
else end

% Fuzzy Inference System

% Output structure for model
outFISm2 = struct('ii', 0, 'out', 0, 'kappa', 0, 'fpr', 0, 'tpr', 0, 'metrics', 0,...
  'confmat', 0, 'outV', 0, 'kappaV', 0, 'fprV', 0, 'tprV', 0, 'metricsV', 0,...
  'confmatV', 0, 'model', 0);
outFISm2 = repmat(outFISm2, 10, 10); % 10 folds
thresholdM2 = 0.5;

% Build FIS
for c=2:10 % For all different numbers of clusters
for ii=1:CVm2.NumTestSets
    Xtrain = Xs(CVm2.training(ii),:);
    Ytrain = Ys(CVm2.training(ii),:);
    Xtest = Xs(CVm2.test(ii),:);
    Ytest = Ys(CVm2.test(ii),:);

    % Train FIS
    tic
    fis = genfis3(Xtrain,Ytrain,'sugeno',c);
    [fis,~,~] = anfis([Xtrain, Ytrain], fis);
    toc
    outFISm2(ii,c).ii = ii;
    % Test and Validate Logistic Regression model
    outFISm2(ii,c).out = evalfis(Xtest,fis);
    outFISm2(ii,c).outV = evalfis(Xval,fis);
    outFISm2(ii,c).model = fis;

    % Evaluate predicted Y values based in thresholdM2
    YpredT = double(outFISm2(ii,c).out >= thresholdM2);
    YpredV = double(outFISm2(ii,c).outV >= thresholdM2);
    conf_logT = confusionmat(Ytest, YpredT,'order',grouporder);
    conf_logV = confusionmat(Yval, YpredV,'order',grouporder);
    % Save metrics for test group (SMOTE)
    [outFISm2(ii,c).metrics, outFISm2(ii,c).kappa, outFISm2(ii,c).fpr,
    outFISm2(ii,c).tpr,...
    '~','~'] = calc_metrics(conf_logT', Ytest,outFISm2(ii,c).out);
    clear outFISm2(ii,c).confmat = conf_logT';
    % Save metrics for validation group (ORI holdout)
    [outFISm2(ii,c).metricsV, outFISm2(ii,c).kappaV, outFISm2(ii,c).fprV,
    outFISm2(ii,c).tprV,...
    '~','~'] = calc_metrics(conf_logV', Yval, outFISm2(ii,c).outV);
    outFISm2(ii,c).confmatV = conf_logV';
    clear conf_logV YpredV
end
end

% Evaluate model on oversampled dataset
close all
[metricsM2(5,:), metricsM2(6,:)] = eval_mdl(outFISm2, 10, 2, 'test','FIS M2 SMOTE');

if save_figs==1
    saveEPS(figure(1), 'roc_fis_m2_test');
    saveEPS(figure(2), 'auk_fis_m2_test')
else end

%% Evaluate model on original validation dataset
close all
[metricsM2_V(5,:), metricsM2_V(6,:)] = eval_mdl(outFISm2, 10, 2, 'val','FIS M2 ORI');

if save_figs==1
    saveEPS(figure(1), 'roc_fis_m2_val');
    saveEPS(figure(2), 'auk_fis_m2_val');
else end

%% Save scores on unbalanced validation set M2:
% Will be used for decision curve analysis
scoresM2_val = zeros(16,4);
scoresLOG = zeros(16,10);
scoresRF = zeros(16,10);
scoresFIS = zeros(16,10);
for ii=1:10
    scoresLOG(:,ii) = outLOGm2(ii).outV;
    scoresRF(:,ii) = outTBm2(ii).outV;
    scoresFIS(:,ii) = outFISm2(ii,2).outV;
end
scoresM2_val(:,1) = mean(scoresLOG,2);
scoresM2_val(:,2) = mean(scoresRF,2);
scoresM2_val(:,3) = mean(scoresFIS,2);
scoresM2_val(:,4) = Yval;
scoresM2_val(:,3) = normalize(scoresM2_val(:,3)); % Normalize ANFIS output

scoresM2_val = mat2dataset(scoresM2_val,'VarNames', {'LR','RF','FIS','clin_rel'});
saveCSV(scoresM2_val,'scoresM2_val.csv')
clear scoresLOG scoresRF scoresFIS scoresM2_val
9.6.12 Model Ranking

%% MODELING - MODEL RANKING

% Create tables with performance metrics per model for comparison

% (c) J.B.R. Visser, November 2015.

%% Table with metrics model 1: on over-sampled test sets
metricsM1 = metricsM1;
%% Create LaTeX table
metricsM1_ds = mat2dataset(metricsM1, 'VarNames', {'mu1', 'sigma1', ...
    'mu2', 'sigma2', 'mu3', 'sigma3'}, 'ObsNames', {'Accuracy', 'Sensitivity', ...
    'Specificity', 'Precision', 'AUC', 'Kappa', 'AUK'});

% Create LaTeX table
metricsM1_ds = dataset2table(metricsM1_ds);
input.data = metricsM1_ds;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Model 1: Results on over-sampled dataset';
input.tableLabel = 'results_m1_test';
latex(input, 'results_m1_test')
clear metricsM1_ds input

%% Table with metrics model 1: on original validation set
metricsM1_V = metricsM1_V;
%% Create LaTeX table
metricsM1_ds = mat2dataset(metricsM1_V, 'VarNames', {'mu1', 'sigma1', ...
    'mu2', 'sigma2', 'mu3', 'sigma3'}, 'ObsNames', {'Accuracy', 'Sensitivity', ...
    'Specificity', 'Precision', 'AUC', 'Kappa', 'AUK'});

% Create LaTeX table
metricsM1_ds = dataset2table(metricsM1_ds);
input.data = metricsM1 ds;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
input.tableCaption = 'Model 1: Results on original dataset';
input.tableLabel = 'results_m1_val';
dataFormat = {’%.4f’}; % one digit precision
latex(input,'results_m1_val')
clear metricsM1_ds input

%% Table with metrics model 2: on over-sampled test sets
metricsM2 = metricsM2';
%% Create LaTeX table
metricsM2_ds = mat2dataset(metricsM2, 'VarNames', {'mu1', 'sigma'....
 'mu2', 'sigma2', 'mu3', 'sigma3'},'ObsNames',{,'Accuracy','Sensitivity'... 
 'Specificity', 'Precision', 'AUC', 'Kappa', 'AUK'});

% Create LaTeX table
metricsM2_ds = dataset2table(metricsM2_ds);
input.data = metricsM2_ds;
input.tableBorders = 1;
input.transposeTable = 1;
input.tableCaption = 'Model 2: Results on over-sampled dataset';
input.tableLabel = 'results_m2_test';
dataFormat = {’%.4f’}; % one digit precision
latex(input,'results_m2_test')
clear metricsM2_ds input

%% Table with metrics model 2: on original validation set
metricsM2_V = metricsM2_V';
%% Create LaTeX table
metricsM2_ds = mat2dataset(metricsM2_V, 'VarNames', {'mu1', 'sigma'....
 'mu2', 'sigma2', 'mu3', 'sigma3'},'ObsNames',{,'Accuracy','Sensitivity'... 
 'Specificity', 'Precision', 'AUC', 'Kappa', 'AUK'});

% Create LaTeX table
metricsM2_ds = dataset2table(metricsM2_ds);
input.data = metricsM2_ds;
input.tableBorders = 1;
input.makeCompleteLatexDocument = 0;
input.transposeTable = 1;
9.6.13  Model Evaluation

% MODELING - Model Evaluation
% This script generates:
% - Random Forest: Out-Of-Bag classification error M1
% - Random Forest: Out-Of-Bag classification error M2
% - FIS: Membership functions plot M1
% - FIS: Membership functions plot M2

% (c) J.B.R. Visser, November 2015.

%% Inspect the OOB error model 1
close all
figure(1);
tic
plot(oobError(outTBm1(1).model));
hiplot = get(gca, 'Children');
set(hiplot, 'LineWidth', 2);
grid on;
xlabel('Number of trees');
ylabel('Out-of-Bag Classification Error');
saveEPS( figure(1), 'oob_error_rf_m1.eps' )
clear hiplot

%% Inspect the OOB error model 2
close all
figure(1);
tic
plot(oobError(outTBm2(1).model));
hiplot = get(gca, 'Children');
set(hiplot, 'LineWidth', 2)
grid on;
xlabel('Number of trees');
ylabel('Out-of-Bag Classification Error');
saveEPS(figure(1), 'oob_error_rf_m2.eps')
clear hlplot

%% FIS M1 Membership Plots
figure(1);
[x,mf] = plotmf(outFISm1(1,2).model,'input',1);
subplot(2,1,1), plot(x,mf);
xlabel('Membership functions for gender');
[x,mf] = plotmf(outFISm1(1,2).model,'input',2);
subplot(2,1,2), plot(x,mf);
xlabel('Membership functions for log\_size');
saveEPS(figure(1), 'mf_m1.eps')

%% FIS M2 Membership Plots
close all
figure(1);
[x,mf] = plotmf(outFISm2(1,2).model,'input',1);
subplot(3,2,1), plot(x,mf);
xlabel('Membership functions (MF) for age');
[x,mf] = plotmf(outFISm2(1,2).model,'input',2);
subplot(3,2,2), plot(x,mf);
xlabel('Membership functions (MF) for log\_size');
[x,mf] = plotmf(outFISm2(1,2).model,'input',3);
subplot(3,2,3), plot(x,mf);
xlabel('MF for MNKRMeta\_KR\_oud\_');
[x,mf] = plotmf(outFISm2(1,2).model,'input',4);
subplot(3,2,4), plot(x,mf);
xlabel('MF for STD\_MNKRMeta\_KR\_oud\_');
[x,mf] = plotmf(outFISm2(1,2).model,'input',5);
subplot(3,2,5), plot(x,mf);
xlabel('MF for MAX\_MNKRMeta\_KR\_oud\_');
[x,mf] = plotmf(outFISm2(1,2).model,'input',6);
subplot(3,2,6), plot(x,mf);
xlabel('MF for MAX\_MNKRNNorm\_KR\_oud\_');
saveEPS(figure(1), 'mf_m2.eps')
### Decision Curve Analysis (R code)

```r
# Basic Data Set-up
# Set our directory
setwd("/Users/jobvisser/Dropbox/TUe/Graduation/3 Thesis/MATLAB/export")
# Source file to use dca command
source("dca.R")
# Load Scores from models
data.val.m1 = read.csv("scoresM1_val.csv")
data.test.m1 = read.csv("scoresM1_test.csv")
data.val.m2 = read.csv("scoresM2_val.csv")
data.test.m2 = read.csv("scoresM2_test.csv")

# Decision Curve Analysis for Model 1
attach(data.val.m1)
# Save Model 1 on validation set
setEPS()
postscript(file="/Users/jobvisser/Dropbox/TUe/Graduation/3 Thesis/LaTeX/figs/dec_curve_val_m1.eps", onefile=FALSE, horizontal=FALSE)
# Run decision curve
dca(data=data.val.m1, outcome="clin_rel", predictors=c("LR","RF","FIS"),
xstop=0.11,
ymin=0)
dev.off()

attach(data.test.m1)
# Save Model 1 on test set
setEPS()
postscript(file="/Users/jobvisser/Dropbox/TUe/Graduation/3 Thesis/LaTeX/figs/dec_curve_test_m1.eps", onefile=FALSE, horizontal=FALSE)
# Run decision curve
dca(data=data.test.m1, outcome="clin_rel", predictors=c("LR","RF","FIS"),
xstop=0.15,
ymin=0)
dev.off()
```

(continued on page 203)
# Decision Curve Analysis for Model 2

# Incorporating Harms into Model Assessment
# the harm of measuring the marker is stored in a scalar
harm_ct = 0.002

attach(data.val.m2)
# Save Model 2 on validation set
setEPS()
postscript(file="/Users/jobvisser/Dropbox/TUe/Graduation/3Thesis/LaTeX/figs/dec_curve_val_m2.eps", onefile=FALSE, horizontal=FALSE)
# Run decision curve
dca(data=data.val.m2, outcome="clin_rel", predictors=c("LR","RF","FIS"),
    harm=c(harm_ct, harm_ct, harm_ct), xstop=0.17,
    ymin=0)
dev.off()

attach(data.test.m2)
# Save Model 2 on test set
setEPS()
postscript(file="/Users/jobvisser/Dropbox/TUe/Graduation/3Thesis/LaTeX/figs/dec_curve_test_m2.eps", onefile=FALSE, horizontal=FALSE)
# Run decision curve
dca(data=data.test.m2, outcome="clin_rel", predictors=c("LR","RF","FIS"),
    harm=c(harm_ct, harm_ct, harm_ct), xstop=0.4,
    ymin=0.22)
dev.off()
9.6.15 Functions

SMOTE

function NewSample = SMOTE(Sample, ClassArr, target, th, att)
%% This script performs SMOTE over-sampling
% Sample = entire dataset with entries
% ClassArr = array with classifying
% variable, to determine minority class
% target = minimum number of samples in minority class
% th = value for calculations
% att = array to indicate if variable is nominal or not.

% Store the indices of the entries in the Sample belonging to the
% classification and determine the amount of neighbours needed for the
% minimum target level
index = find(ClassArr == 1);
k = ceil(target / length(index));
% Create matrix (length(index)*k) with indices for the nearest
% neighbours with k as the number of neighbours
NN = NearestNeighbour(index, length(index), k);

% Determine the new entries
S = [];
for i = 1:length(index)
    for j = 2:k
        P = th .* Sample(index(i), :) + (1 - th) .* Sample(NN(i, j), :);
        S = [S; P];
    end
end

% Round the variables which are nominal or 1/0
if (exist('att', 'var'))
    for i = 1:length(att)
        if(att(i) == 1)
            S(:, i) = round(S(:, i));
        end
    end
end

204
% Returning the new Sample matrix shuffled
NewSample = [Sample; SI];
NewSample = NewSample(randperm(size(NewSample, 1)), :);

% Function which creates a matrix of dimensions SampleSize*k with all
% the indices of the nearest neighbours of all entries in the Sample
function NN = NearestNeighbour(index, SampleSize, k)
    % For every entry in the Sample a array of NN indices are created
    for i = 1:SampleSize
        % The first part all have the same indices due to to few
        % Samples with lower indices
        if (i <= ceil(k / 2) + 1)
            NN(i, :) = index([1:k]);
            NN(i, i) = NN(i, 1); % Set the cell where index(i) is
            % stored to the first stored value,
            % so index(i) can be set to NN(i, 1)
            % at the end for further processing
        end
        % Entries in the middle
        elseif (i > ceil(k / 2) + 1 && i <= SampleSize - ceil(k / 2))
            NN(i, :) = index([(i - ceil(k / 2)):(i + floor(k / 2) - 1)]);
            NN(i, ceil(k / 2) + 1) = NN(i, 1);
            % The same story as with the first entries of the Sample, only
            % for the last entries
        else
            NN(i, :) = index([(SampleSize - k + 1):SampleSize]);
            NN(i, k - (SampleSize - i)) = NN(i, 1);
        end
        % Swapping the index if i to the first column
        NN(i, 1) = index(i);
    end
end
function [ metrics_avr, metrics_std ] = eval_mdl( out, folds, cluster, type, name )

% EVAL_MDL generates all relevant performance metrics for model evaluation
%
% INPUT: out = Output structure (fields: ii, out, kappa, fpr, tpr,
%       metrics, confmat, outV, kappaV, fprV, tprV, metricsV,
%       confmatV, model)
% folds = Number of folds included in the output structure
% cluster = Cluster where you want the results from
%
% OUTPUT: metrics = Average of performance metrics:
% - accuracy
% - sensitivity (eqv. tpr, hit rate, recall)
% - specificity (eqv. true negative rate)
% - precision (eqv. positive predictive rate)
% - AUC = area under the curve
% - kappa = kappa coefficient
% - AUK = area under kappa
% ROC = Averaged ROC curve for model + metrics legend
% AUK = Averaged AUK curve for model + metrics legend
%
% (c) J.B.R. Visser, October 2015.

% c = cluster; % number of clusters selected
% X = 0; % fpr for x coordinates of ROC curve
metrics = zeros(folds,7);

if strcmp(type, 'test') == true
    for ii=1:folds
        X = [ X; out(ii,c).fpr ];
        metrics(ii,:) = out(ii,c).metrics;
    end
metrics_std = zeros(1,7);
metrics_avr = zeros(1,7);

    for i=1:7 % All performance metrics
        metrics_std(1,i) = std(metrics(:,i)); % std of performance metrics
metrics_avr(1,i) = mean(metrics(:,i)); \% mean of performance metrics

X = unique(X);
[n,~] = size(X);
Y = zeros(n,1); \% tpr for y coordinates of ROC curve
K = zeros(n,1); \% vector with kappa values for AUK curve

for ii=1:folds
    xx = out(ii,c).fpr; \% take fpr vector from fold
    yy = out(ii,c).tpr; \% take tpr vector from fold
    kk = out(ii,c).kappa; \% take kappa vector from fold
    for jj=1:n \% for every value in the fpr vector
        Y(jj) = Y(jj) + max(yy(xx<=X(jj)));
        K(jj) = K(jj) + kk(find(xx<=X(jj), 1, 'last'));
    end
end

elseif strcmp(type, 'val') == true
    for ii=1:folds
        X = [ X; out(ii,c).fprV ];
        metrics(ii,:) = out(ii,c).metricsV;
    end
metrics_std = zeros(1,7);
metrics_avr = zeros(1,7);
for i=1:7 \% All performance metrics
    metrics_std(1,i) = std(metrics(:,i)); \% std of performance metrics
    metrics_avr(1,i) = mean(metrics(:,i)); \% mean of performance metrics
end

X = unique(X);
[n,~] = size(X);
Y = zeros(n,1); \% tpr for y coordinates of ROC curve
K = zeros(n,1); \% vector with kappa values for AUK curve

for ii=1:folds
    xx = out(ii,c).fprV; \% take fpr vector from fold
    yy = out(ii,c).tprV; \% take tpr vector from fold
    kk = out(ii,c).kappaV; \% take kappa vector from fold
    for jj=1:n \% for every value in the fpr vector
        Y(jj) = Y(jj) + max(yy(xx<=X(jj)));
        K(jj) = K(jj) + kk(find(xx<=X(jj), 1, 'last'));
    end
end
\begin{verbatim}
Y(jj) = Y(jj) + \text{max}(yy(xx<=X(jj))); 
K(jj) = K(jj) + kk(find(xx<=X(jj), 1, 'last')); 
end 
end 
else 
end 

Y = Y / folds; \% average of tpr vector 
K = K / folds; \% average of kappa vector 

%% Construct ROC and AUK curves 
mdl = name; 
if c == 1 
title_1 = sprintf('Averaged ROC curve for model: %s', mdl); 
title_2 = sprintf('Averaged Kappa curve for model: %s', mdl); 
else 
title_1 = sprintf('Averaged ROC curve for model: %s, c = %s', mdl, num2str(c)); 
title_2 = sprintf('Averaged Kappa curve for model: %s, c = %s', mdl, num2str(c)); 
end 
% Create plot for ROC curve 
figure; 
%subplot(1,2,1); 
plot(X,Y); 
xlabel('False positive rate',...' 
'FontSize',12,'FontName','Times'); 
ylabel('True positive rate',...' 
'FontSize',12,'FontName','Times'); 
title(title_1,'FontUnits','points',... 
'interpreter','latex',...' 
'FontSize',12,'FontName','Times'); 
% Generate legend box with (average of) metrics 
legend_1 = {{'Accuracy = ', num2str(metrics_avr(1,1), 3)}, ... 
{'Sensitivity = ', num2str(metrics_avr(1,2), 3)}, ... 
{'Specificity = ', num2str(metrics_avr(1,3), 3)}, ... 
{'Precision = ', num2str(metrics_avr(1,4), 3)}, ... 
{'AUC = ', num2str(metrics_avr(1,5), 3)}; 
annotation('textbox',[0.65 0.19 0.137105555555555 0.133617021682415],... 
\end{verbatim}
'String', legend_1,'FontUnits','points',...
'Interpreter','latex',...
'FontSize',11,...
'FontName','Times',...
'FitBoxToText','on',...
'EdgeColor','none');

% Create plot for AUK curve
figure;
%xsublot(1,2,2);
plot(X,K);
xlabel('False positive rate','FontUnits','points','interpreter','latex',... 
'FontSize',12,'FontName','Times');
ylabel('Cohens kappa','FontUnits','points','interpreter','latex',... 
'FontSize',12,'FontName','Times');
title(title_2,'FontUnits','points',...' 
'interpreter','latex','FontSize',12,'FontName','Times');
% Generate legend box with (average of) metrics
legend_2 = {{'Cohens kappa = ', num2str(metrics_avr(1,6), 3)], ... 
['AUK = ', num2str(metrics_avr(1,7), 3)]};
annotation('textbox',[0.65 0.152191044686722 0.241160772543038 
0.0842589067777768],...
'String', legend_2,'FontUnits','points',...
'Interpreter','latex',...
'FontSize',11,...
'FontName','Times',...
'FitBoxToText','on',...
'EdgeColor','none');

end
function [ metrics, k, fpr, tpr, xroc, yroc ] = calc_metrics( conf, Ytest, sc )

%METRICS calculates the performance metrics based on the confusion matrix:

% ___real___
% predict | --------- |
% | TP | FP |
% | FN | TN |

% INPUT: conf(1,1) = true positive (eqv. hit)
% conf(1,2) = false positive (eqv. false alarm)
% conf(2,1) = false negative (eqv. miss)
% conf(2,1) = true negative (eqv. correct reject)

% OUTPUT: metrics = Matrix with performance metrics:
% -accuracy
% -sensitivity (eqv. tpr, hit rate, recall)
% -specificity (eqv. true negative rate)
% -precision (eqv. positive predictive rate)
% -AUC = area under the curve
% -kappa = kappa coefficient
% -AUK = area under kappa

% k = Vector of kappa index values.
% fpr = Vector of false positive rates.
% tpr = Vector of true positive rates.
% xroc = Vector of x coordinates for ROC curve
% yroc = Vector of y coordinates for ROC curve

% (c) J.B.R. Visser, October 2015.

% Transpose confusion matrix because confusionmat delivers it with actual
% labels in the vertical axis.
% conf = conf';

n = double(conf(1,1) + conf(1,2) + conf(2,1) + conf(2,2));
metrics(1,1) = (conf(1,1) + conf(2,2)) / n; % accuracy
metrics(1,2) = conf(1,1) / (conf(1,1) + conf(2,1)); % sensitivity
metrics(1,3) = conf(2,2) / (conf(1,2) + conf(2,2)); % specificity
metrics(1,4) = conf(1,1) / (conf(1,1) + conf(1,2)); \% precision
[xroc,yroc,~,metrics(1,5)] = perfcurve(Ytest,sc,1); \% xroc, yroc, AUC

\%
Kappa: an observational probability of agreement and (random acc) is a
\%
hypothetical expected probability of agreement under an appropriate set of
\% baseline constraints
agree_obs = metrics(1,1); \% total accuracy i.e. observed agreement
agree_exp = (((conf(1,1)+conf(1,2))*(conf(1,1)+conf(2,1)))/n) + ... \% predT*realT
 (((conf(2,1)+conf(2,2))*(conf(1,2)+conf(2,2)))/n)) ... \% predF*realF
 / n; \% (predT*realT+predF*realF) / n

metrics(1,6) = (agree_obs - agree_exp) / (1 - agree_exp); \% kappa
[k,fpr,tpr] = calckappa(sc, Ytest);
metrics(1,7) = trapz(fpr, k); \% AUK

end