Machine learning in healthcare invoicing systems

Using text mining and supervised learning to verify classifications of unstructured medical texts

A. E. Brons
4300807

Master's Thesis
Artificial Intelligence
Faculty of Science

March 20th 2017

Internal supervisors:
dr. A.J. Feelders
prof. dr. A.P.J.M. Siebes

External supervisor:
MSc. J. Kuijpers

Universiteit Utrecht

topicus

Abstract

In the process of healthcare invoicing, many mistakes are made when physicians assign activity codes to treatments. Health insurance companies require hospitals to check the assigned activity codes and therefore Electronic Health Records (EHR) are examined manually. In this thesis, a system is proposed that automatically checks whether assigned activity codes are correct or not, based on unstructured EHR texts. This binary prediction was made with the use of supervised machine learning algorithms. Several algorithms are compared: Logistic regression, Naive Bayes, Neural Network, and Support Vector Machines. Furthermore, the classification problem was extended to a multi-class classification in which the reason of rejecting an incorrectly assigned activity code was predicted. Accuracies of 93.3% and 87.4% were achieved for respectively the binary and the multi-class classification. It was found that feature selection had a higher impact on the results than the choice of the algorithm. Future work can investigate new activity codes that have other requirements. Moreover, the current system can be used for prevention instead of checking.

Keywords: healthcare invoicing systems, supervised machine learning, text mining, data mining

Table of contents

1 Introduction

"The techniques of Artificial Intelligence are to the mind what bureaucracy is to human social interaction."

Terry Winograd

Over the last decades, Artificial Intelligence has evolved exceptionally fast. Things that were not considered possible several years ago, now are widely used. Nowadays, it is faster to ask Siri when the train departs than to look it up yourself. Moreover, self-driving cars are emerging and it will not be long before your car brings you wherever you want to go to while you sit back and relax. Furthermore, vast innovations from Artificial Intelligence were introduced in healthcare. Take for instance the iKnife [1, 21], a self-learning device that gives direct feedback about the tumour tissue that is being cut, which helps surgeons in deciding what amount of tissue to remove. Another example is computer-aided diagnosis in medical imaging [11]. By using Artificial Intelligence techniques, computers are now able to diagnose certain diseases by evaluating medical images. Medical improvements have led to improved human health and longevity. However, they have caused health expenses to growth significantly as well [4].

There exist numerous rules and regulations in healthcare to control the high costs. Therefore, medical institutions are bound to strict requirements when invoicing healthcare provided by them [32]. However, these requirements consume time and resources that could otherwise have been devoted to medical care. To start with, hospitals need to categorise the medical care they provide in order to invoice the health insurance companies using the corresponding invoice codes. Subsequently, when hospitals have already been rewarded, they also have to prove that the registration and classification during the process was accurate, lest they have to refund the overpay.

Not only technologies in healthcare have improved quickly over the last years, information technology has improved substantially as well. Consequently, a great extent of the healthcare declaration process can now be executed automatically. Nevertheless, there still remain parts of the process that have to be carried out manually. More specifically, while applications that define invoice codes

and that send itimised invoices to health insurance companies already exist, they do not yet check whether the used activity code, on which the invoice code is based, is correct. Therefore, automating the process of checking whether every registered activity belonging to the classified treatment has indeed been performed could further optimise the procedure.

The current research proposes and builds a proof of concept that uses supervised machine learning techniques to automatically check whether or not an activity code was justified. This research project is conducted at Topicus, a company that develops, among other things, technological solutions for healthcare institutions. The particular project that this research is part of consists of several software applications that support the financial aspect of healthcare. As mentioned above, Topicus has already developed applications that categorise provided medical care into the correct invoice classes and that compose and send invoices to health insurance companies. On the contrary, an application that checks whether the requirements of the treatment classification have been met still has to be built.

The remaining part of this introduction is structured as follows. To start with, the problem at hand will be explained in more detail. Subsequently, related research will be discussed. Lastly, the research question will be posed, while simultaneously presenting the corresponding hypothesis.

1.1 Problem context

To record their provided medical care, hospitals use an Electronic Health Record (EHR), which enables medical practitioners to systematically collect information of patients in a digital format. To subsequently report their information to health insurance companies in order to get paid, hospitals use the so-called DBC system, which stands for Diagnosis Treatment Combination (Diagnose Behandel Combinatie). This system is developed by the Nederlandse Zorgautoriteit (NZa), which stands for Dutch Healthcare Authority [33]. The DBC system is based on the RSAD model, which is an abbreviation for Register, Summarise, Deduce, and Invoice (Registreren, Samenvatten, Afleiden en Declareren). The DBC system was not only introduced to register provided care with the purpose to invoice, it was also introduced to improve both the quality and the efficiency of healthcare [17].

Before explaining the DBC system in more detail, it is important to emphasise the difference between two different codes, namely activity codes and DBC codes. The NZa has provided a list with all possible activity codes and their corresponding short descriptions [31]. Based on this list

and the diagnosis, physicians report both the provided therapy and the corresponding activity code in the EHR. Additionally, there are DBC codes, which are used for invoicing. Using decision trees, DBC codes are derived from activity codes, but certainly do not equal them.

Three main characteristics of the DBC system are the product structure, the registration rules, and the grouper [33]. First, the product structure contains the arrangement and grouping of all diagnoses and medical activities in the set of DBC codes. In other words, the product structure defines how to translate the activity codes that are reported in the EHR into DBC codes. Second, the registration rules define when to initiate and close a health trajectory for a patient. Furthermore, these rules prescribe what information is needed for registration in order to select the correct DBC code at a subsequent stage. Lastly, there is the grouper, which is a system that combines the activity codes stored in a patient's health trajectory with the product structure to deduce the corresponding DBC codes. This is a complex process since there exist over 5000 DBC codes. After the grouper has returned the DBC code, the medical institution uses a specific software system to invoice the provided medical care based on this code. In order to improve efficiency, hospitals receive an agreed price for a provided combination of diagnosis and treatment. Therefore, hospitals are forced to minimise inessential activities. This can for instance be done by reducing duplicate and unnecessary tests and treatments, or by reducing the length of a hospitalisation [8].

In sum, when a patient requests medical care, a health trajectory is opened. Every activity executed to diagnose or treat a specific disease, is registered in the EHR. For instance, in case of a broken leg, the activity to diagnose could consist of an X-ray while the treatment could consist of plastering the broken leg or perform surgery. The corresponding activity code is added to the patient's trajectory in the DBC system. After 120 days, or when the treatment is finished, the trajectory is closed and the data are sent to the grouper. The grouper connects this information to a specific DBC code, which in turn can be used by the hospital to send an invoice to the health insurance company.

There exist applications that determine DBC codes and send itemised invoices corresponding to these codes. However, activity codes have to be determined manually. As experienced by Topicus, many mistakes are made by physicians in selecting activity codes. Because a large amount of money is involved, hospitals have to prove that the classification of their activity code was accurate - even when the invoices have already been paid. More specifically, they have to prove for a given number of cases, that specific actions required for the derived activity code have indeed been performed an registered. Unfortunately, this part of the process often needs to be done manually, which involves the very time consuming task of analysing digital unstructured texts. Every assigned activity code requires a specific combination of a diagnosis, physician's description and medical activities, which

is for instance saved in a description of an Emergency Room visit or a referral letter, both of which are in turn saved in the EHR. Despite the fact that the registration process has many requirements, there are no fixed rules for the notation of the information in documents such as referral letters and consult descriptions. Hence, little consistency exists among documents, which complicates the problem of extracting the right information from the unstructured text.

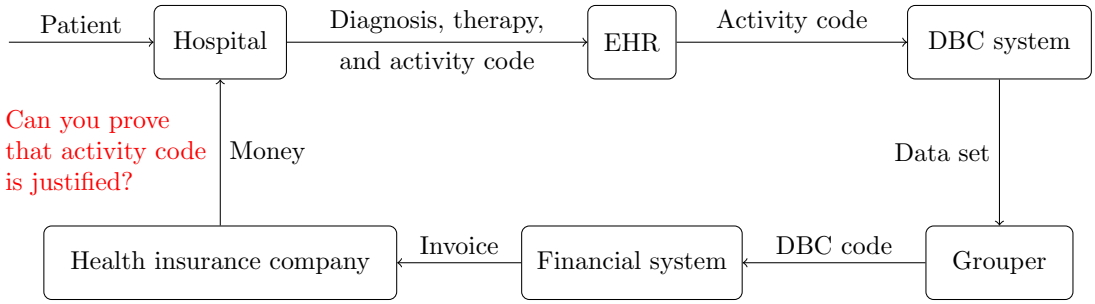

Figure 1.1: Schematic overview of the invoice process in hospitals

1.2 Related work

No previous research could be found regarding the automation of the declaration process in the Dutch healthcare system, due to the rareness and specificity of both the Dutch healthcare and health insurance systems. However, other concepts regarding data mining in healthcare have been investigated. One of the first applications in medical data mining was health-KEFIR, short for Key Findings Reporter in healthcare [27, 29]. This application automatically analysed deviations in relevant variables such as laboratory results or medication use. Since a lot of variables should be taken into account when deciding on a patient's therapy, physicians might overlook some deviations. A medical data mining system can help by directing attention to important deviations, hence facilitating the decision on the patient's therapy. Moreover, data mining could be used to provide assistance in predicting a medical diagnosis [18, 39]. Large sets of medical information such as laboratory results and descriptions of examinations combined with the correct diagnosis can be used to train a predictive algorithm. When the algorithm is sufficiently trained, its predictions can help physicians in diagnosing. Furthermore, medical data mining could be used to help predict the prognosis of a (chronic) disease [20],
or even predict the survival time of patients that are on the waiting list for an organ transplant [24].

The use of medical data mining creates several possibilities to improve healthcare, but this area

also induces a lot of difficulties and limitations that have frequently been discussed [7, 23, 29]. To start with, despite the fact that abundant data sets are available, not all data can be analysed easily because of privacy and security concerns. In North America, Europe, and Asia at least some of the medical information of as many as three-quarters of a billion persons is collected in electronic form [7]. To ensure their privacy, the data have to be anonymised and the analysis has to be carried out in a secured environment. Furthermore, the data set may contain inconsistent, insignificant or redundant information, or the data may even be incomplete due to tests that were not or imprecisely performed. Furthermore, an important characteristic of medical data is the weight of physicians' descriptions. These descriptins are difficult to standardise, and therefore difficult to mine because they are written in unstructured or semi-structured text. In describing a patient's condition and describing the relationship among medical entities, both ambiguous terms and distinct grammatical constructs are regularly used. Moreover, a lot of abbreviations are used by physicians, whether standard or not [9]. Sometimes it is barely feasible to manually decode the tangle of uncommon abbreviations written by a physician, let alone to automate this process. Because of all these possibilities in the encoding and description process, each medical condition can be described in multiple ways. However, in data mining it is preferred to have the data in canonical form, which is a notation that encloses all interchangeable forms of the same concept [2]. Since there are no prescribed registration forms of medical information, there is a profusion of distinct expressions that are all medically equivalent [7]. Even elementary concepts in medicine do not have a canonical form, which makes it fairly difficult to analyse all of these texts.

Popowich [35] has investigated the problem of analysing unstructured medical text. More specifically, the use of text mining and natural language processing in determining whether healthcare claims involve potential fraud or abuse. The aim of his study was to build an algorithm that determines whether other parties were (partially) responsible for covering the costs of medical claims. As mentioned, medical texts can be full of synonyms, abbreviations, acronyms, and jargon, which complicates the text mining process. A large number of synonyms might cause an unnecessary number of features. Moreover, individual features might be less predictive than a single feature that combines all synonyms. This problem was solved using WordNet [15], which is a lexical database for the English language, was used. In this database, synonyms are grouped into so-called synsets. Since WordNet is not specifically built for the medical lexicon, a subset of its synonym database was adapted to the medical domain. Unfortunately, using WordNet is not possible in this thesis, as the focus is on Dutch medical texts. Although a Dutch version of WordNet is available, it merely contains synonyms for a small subset of the strings available in the English version. Moreover, a large number of words that is used in the texts to be analysed is too specific to the medical field to occur in the general database. SNOMED Clinical Terms [34], a collection of medical terms, codes and synonyms, also seemed appropriate to use, but regrettably this database does not exist in Dutch

either.

Regarding the text mining algorithm, Popowich reported several preparatory steps. To start with, text fragments are used as input. Subsequently, these fragments are split into individual words and the most relevant fragments are selected. Then, using WordNet, the number of selected words is diminished by replacing synonyms. Next, the relevance of words is determined and words that are not relevant enough are removed. After these steps have been completed, the remaining indicators are used to train the algorithm.

Although a limited number of relevant publications in this area of research could be found, extant literature has addressed several important difficulties. Analysis of the substantial amount of available medical data could lead to improvements in healthcare if some aspects that are typical for medical data are taken into account. First, the privacy of all of the data should be secured. Second, the possibility that data might be incorrect or missing should be considered. Lastly, it should be contemplated that mining unstructured texts might be difficult due to myriad ways to express a medical description. The use of lists of synonyms and abbreviations might be helpful here.

1.3 Research questions

Restating, hospitals have to prove that their declared activity codes are correct, i.e. that every required activity belonging to this activity code has indeed been registered and performed. This information is often disclosed in referral letters or in descriptions of Emergency Room visits, both of which are stored in the hospital's electronic health record. Therefore, digital unstructured texts frequently have to be analysed manually. However, manually analysing digital unstructured texts can presumably be replaced an application that uses Artificial Intelligence techniques. The present study uses machine learning techniques to propose and build a proof of concept checking whether every required activity for a specific activity code is indeed performed and recorded. In other words, the following research question is posed:
Is it possible to develop an automatic classification system, using supervised machine learning algorithms, that determines whether an activity code was assigned correctly or not, based on digital unstructured text stored in the electronic health record?

As mentioned before, each DBC code requires a specific set that consists of the diagnosis, the physician's interpretation and the medical activities. Hospitals only get paid when they have both performed and registered all of this information. The analysis and inspection of the classification

procedure has two possible outcomes: It can either be correct, or incorrect. Since prior classification data are available and both classes are predefined, supervised machine learning algorithms will be used. However, several supervised machine learning algorithms can be used to solve the stated classification problem. Therefore, not only a proof of concept is proposed, it is also investigated which machine learning algorithm reaches the best performance in analysing these healthcare texts. This leads to the following subquestion:
Which supervised machine learning algorithm performs best?

In addition to the available information about the correctness of the prior activity code classification, the reasons for rejecting incorrectly classified activities are available as well. Since this information can be useful to help hospitals improve their administration and convince health insurance companies, it would be beneficial to predict the reason for rejection along with the aforementioned binary classification. This leads us to the second subquestion:
Is it possible to predict the reason of rejection for activity codes that are classified as incorrect?

The performance of the classification is not merely affected by the kind of machine learning algorithm, as the features that are used as input influence the performance as well. Therefore, generating and selecting predictive features is important. This results in the third subquestion:
Which features are the most predictive in the current classification problem?

An automatic classification system that either confirms or rejects prior manual classifications in the healthcare declaration system will result in a considerable decrease in workload, since manually analysing unstructured texts in the electronic health record is a very time consuming task. Although no previous work could be found regarding this specific research problem, text mining is already successfully used to solve several other problems in healthcare industry [20, 27, 29, 35, 39]. Hence it is expected that it will be possible to build a model that confirms prior classifications of activity codes, using supervised learning techniques. However, predicting the performance of the model using different supervised machine learning algorithms is more difficult. The aim of the proof of concept is to evaluate whether the prior assignment of an observation to a class was correct or not. Assuming that the majority of the prior observations is assigned correctly, the easiest approach is to predict that all observations are assigned correctly. In that case, the accuracy rate is equal to the percentage of cases in the majority class. Therefore, the accuracy of the envisioned proof of concept is expected to be at least the same as the percentage of cases in the majority class. However, it is nearly infeasible to predict the exact accuracy of the model.

1.4 Outline

In this first chapter, the Dutch healthcare declaration system was introduced. Although over the last years progress has been made regarding the automation of this process, no application yet exists that simplifies the process. In particular, an application that helps proving that every requirement corresponding to a specific activity code is provided and registered is still absent. The current study uses supervised machine learning techniques to propose and build an approach that supports this part of the process. In order to formulate an answer to the research questions and to evaluate the validity of the aforementioned hypotheses, the proposed proof of concept will be implemented.

In Chapter 2, relevant methods, theories, and techniques in the area of machine learning will be outlined. In Chapter 3, the data set that is used for the current research will be discussed. Subsequently, in Chapter 4, the architecture in which the aforementioned techniques are integrated into a proof of concept will be laid out. Chapter 5 will present the experimental setup of this research, filled by the experimental results in Chapter 6. Finally, in Chapter 7, the conclusion of this research will be presented and discussed.

2 Theory

"What is the difference between theory and practice? There is no difference, in theory. But in practice there is."

Ian Witten and Eibe Frank

In the first chapter, the research problem is introduced and its context is explained. In this thesis, data mining will be used to analyse unstructured text in medical documents in order to be able to automatically check prior manually assigned activity codes. The current chapter addresses the theoretical background of text mining and describes the methods and techniques that will be used in the proposed application.

2.1 Data mining

According to Witten and Frank [43], data mining is defined as the process of, automatically or semi-automatically, discovering meaningful structural patterns in large quantities of data in order to make nontrivial predictions on new data. Moreover, they define machine learning as the acquisition of structural descriptions from examples that are used for prediction, understanding, and explanation. The concepts 'data mining' and 'machine learning' are frequently used interchangeably. However, the term machine learning can be used more comprehensively. Namely, the most important part of this field of study consists of learning to make right predictions on new data. 'Learning' can be defined as changing behaviour such that future performance will be better. In other words, the data is mined in order to learn how to improve predictions on new data [43].

With the use of machine learning, three distinct learning problems can be solved: supervised, unsupervised, and reinforcement learning problems [38]. As the name indicates, supervised learning algorithms use the help of a supervisor that provides correct examples. The training data comprises observations along with their corresponding response [3]. The goal is to fit a model that relates

the associated response to the predictors in order to accurately predict the response for future observations [22]. In other words, a function is learned from examples of its inputs and outputs. In supervised learning, learning a continuous function is distinguished from learning a discrete-valued function. The first is called regression, the latter classification. In contrast to supervised learning, unsupervised learning does not learn from data sets that are provided with correct answers. Unsupervised learning is denoted as discovering groups of similar examples within the data [3]. An unsupervised learning algorithm will try to infer a function to describe a hidden structure from unlabeled data. Thus, a pattern is learned from the input while no specific output values are supplied [38]. Using unsupervised learning to define groups with corresponding features is called clustering. The most important difference between supervised and unsupervised learning lies in the labels of the classes. In supervised learning, cases are classified into predefined categories, while in unsupervised learning the classes are not known in advance. The third kind of learning is reinforcement learning, which is defined as learning how to map situations to actions, in order to maximise a numerical reward signal [40]. This type of learning distinguishes itself from supervised learning by discovering what actions yield the highest reward by trial-and-error, instead of learning what action to perform in what situation by error signals from examples.

The central question to this thesis is whether assigned activity codes were correct or not. Since the class labels are known in advance, and a data set with correct classifications is available, the classification problem in this thesis will be solved using supervised learning techniques. To get more comfortable with these kinds of techniques, supervised learning and specifically classification will be laid out in detail.

2.1.1 Classification

In classification, a distinction is made between single label and multi-label classification. In the first case, categories do not overlap and an observation may only be grouped in a single class. On the contrary, multi-label allows categories to overlap. Therefore, an observation can be assigned to several categories simultaneously. However, assigning a case to several categories is not obligatory and a case might also be assigned to one or none of the categories. In this thesis, a case will be assigned to just one class, and therefore it is a single label classification. More specifically, it is a special instance of a single label classification problem. Since a case has to be classified as either correct or incorrect, it is a binary, of Boolean, classification.

As mentioned before, classification is a supervised learning technique that is used to learn a discrete-valued function from examples of its inputs and outputs. Thus, observations have to be

assigned to predefined classes. In other words, the discrete-valued function will map the observations to the predefined categories. The goal is to approximate an unknown target function. For the binary classification problem, the function can be described as:

$$f : x \rightarrow \{\mathcal{T}, \mathcal{F}\} \tag{2.1}$$

In case the reasons for rejection are predicted as well, there are some more categories available, resulting in the following function:

$$f : x \rightarrow \{\mathcal{T}, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\} \tag{2.2}$$

Each $\mathcal{F}$-class represents a category in which similar reasons of rejections are grouped. These different groups will be discussed in more detail in Chapter 3.

2.1.2 Training and testing

The data set has to be split into a training set, used to learn the function, a validation set, used to optimise the parameters, and a test set, used to assess the function's accuracy. To obtain adequate optimization and predictive performance, each set should contain unique data. Otherwise the function might perform well on the training data but poor on on unseen data, such that the predictive performance might be too optimistic. Overfitting is an example of a problem that might occur when the training data is also used to validate and test the trained function. Although an overfitted function is correct, it is too narrow since it is trained too explicitly on a specific training set, as is illustrated in Figure 2.1. In other words, it is trained on the noise in the data rather than of the underlying relation. Therefore, such a function is only useful in predicting items whose class is already known, while the goal is to achieve the best predictive performance on new data [13]. The performance on a training set hence does not necessarily predict the performance on a test set with new data.

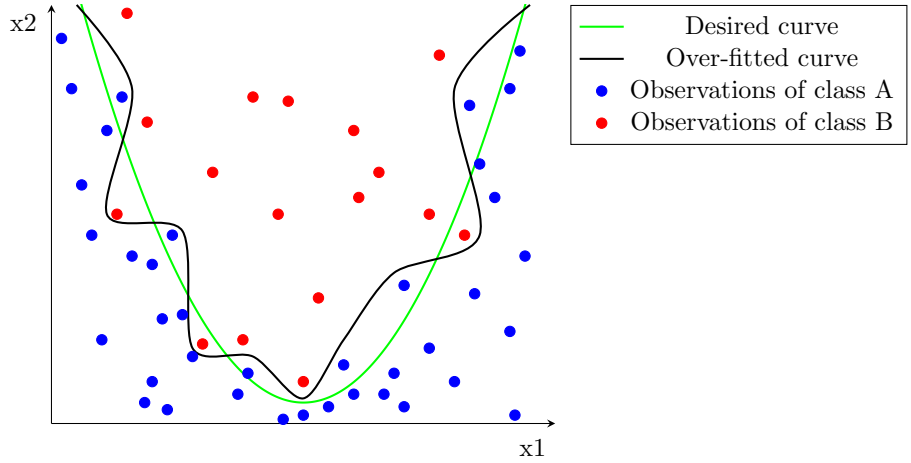

Figure 2.1: Overfitting

Not only do three unique sets of data have to be available, all sets have to contain enough unique data as well. The training set has to be as large as possible to build a proficient model. Moreover, the validation set has to contain enough data in order to reach good performance in optimisation. Furthermore, to be able to obtain a reliable estimate of the predictive performance, the test set has to be sufficiently large as well. However, the supply of data might be limited and therefore it might be demanding to retrieve enough data to form three unique and sufficient data sets. Fortunately, cross-validation can be used to tackle this problem [3]. When applying cross-validation, the proportion of the available data that can be used for both training and testing will increase. The data set will be divided into S subsets, which in the simplest case will all be of equal size. Therefore, this process is called S-fold cross-validation.

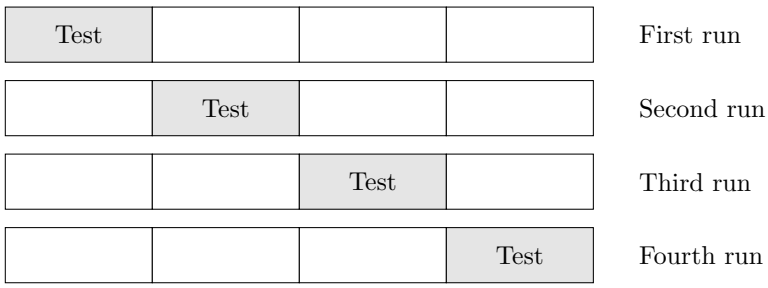

Figure 2.2: S-fold cross validation with S = 4

In Figure 2.2, S-fold cross-validation is illustrated. In each run, $S - 1$ groups (white) will be used to train a set of models, while the remaining group (grey) will be used to evaluate the trained models. In total, S runs will be executed and the performance scores from these runs will be

averaged. However, cross-validation increases not only the amount of data that can be used for training and testing, but also the total number of executed runs. The inevitable extension of runs that have to be performed, with a factor S to be specific, may cause problems when the training of a model is computationally expensive.

Another aspect that might influence the performance of the model is the disproportion in the number of examples assigned to each class, which is called the problem of skewed class distribution [30, 43]. In a binary classification problem this implies that there is a substantial discrepancy between the number of observations that are labelled as true and the number of observations that are labelled as false. If the class distribution of the example data set is skewed, this might result in a classifier that is able to accurately predict the majority class, but a predictive accuracy for the minority class that is not satisfactory. This is for instance the case if each class is assigned to the majority class. This problem will arise in case the misclassification cost for the majority class is much higher than the misclassification cost for the minority class. For this problem, a cost-sensitive learning system can be used in order to reduce the cost of misclassified observations instead of reducing the classification error. The class distribution in the example data set will be artificially imbalanced to make the minority class more costly in order to reach a better classification accuracy of the minority class. Therefore, the examples will be weighted based on the class to which they are assigned to. Then, the classifier will be skewed toward the avoidance of errors in the minority class in order to reduce the overall cost. Furthermore, skewed class distribution might be solved by applying undersampling or oversampling. Both involve artificial manipulation of the example data set. In undersampling a more balanced class distribution will be achieved by eliminating examples of the majority class. In contrast, oversampling balances the data set by replicating examples of the minority class.

2.1.3 Performance measure

Predictive performance of the classifier is often measured by means of the error rate, which is the relative number of misclassifications [30, 43]. Because the classifier was based on the training set it is important not to confound the error rate on the training data, also called the resubstitution error, with the error rate on the test data. The opposite of the error rate, called the success rate or accuracy, is also used to indicate predictive performance. Here, the relative number of correct classifications is measured instead of the relative number of misclassifications.

Using the total number of errors to define the performance of a classifier might be misleading. As illustrated in Table 2.1, there exist four different possible outcomes in a binary classification

problem. Correct classifications can be divided into true positives and true negatives. On the contrary, misclassifications can be divided into false positives and false negatives. False positives occur when an observation is classified as true, while it should have been classified as false. False negatives occur when an observation is classified as false, while it belongs in the positive class. In the standard error rate, the costs of all errors are the same. However, a false negative error might have more impact than a false positive. Consider for instance the question whether a patient is ill or not. If the classification is false positive, someone will undergo unnecessary further examinations. On the other hand, treating a patient because of a false negative error may have disastrous consequences. Similar to the solution of a skewed distributed data set, cost-sensitive classification can be used to solve the problem of different types of classifications.

		Predicted class	
		True	*False*
Correct class	*True*	True positive	False negative
	False	False positive	True negative

Table 2.1: Confusion matrix

2.1.4 Feature representation

In addition to reflecting on the amount and distribution of training and test data, the content of the data should also be considered. Since the model function is based on the used data, the data have to be representative for the real world problem that the classifier is modelled for. If the training set does not contain representative data, the function might perform perfectly on the training and test data but lousy on real-world data. However, even if the training data are representative, the data in its original form is rarely useful as input for the machine learning algorithm. Relevant features, which are distinct properties of the data, should be used as input rather than just the raw texts in order to improve the classifier's performance [43]. The first step in this process is feature generation, which involves transforming the data into features. However, this may lead to copious features and an increased running time of the learning algorithm. Since numerous of these features are irrelevant or redundant to the target concept, it will be beneficial to prune these excessive features [10]. This leads us to the next step in the process: feature selection. In this, the number of features is reduced by taking a subset of all features in which only features that are relevant and not redundant are selected. Both feature generation and selection are of great importance in text mining, since learning algorithms in this area can neither cope with flat texts nor with an over-abundant number of features as input.

2.2 Text mining

Text mining is a subfield of data mining and can be described as the process of extracting patterns from unstructured text documents [41]. The goal is to assign texts to the correct categories, based on their context [12]. Whereas in data mining the data are already stored in a structured format, in text mining the data should be preprocessed first to gain a structured set of data from unstructured natural language [14]. The process of discovering structured information from unstructured text has also been referred to as information extraction.

2.2.1 Pre-processing

Pre-processing is done to convert unstructured natural language into a structured form that can be handled by text mining algorithms. Many sophisticated pre-processing techniques exist, but they can be divided into two groups: natural language processing (NLP) and information extraction (IE) techniques. The first involves domain-independent linguistic features, while the latter deals with domain-specific features. A frequently used NLP technique is tokenisation, in which the continuous stream of characters is divided into meaningful constituents. Thus, the text document is split up in tokens, such as chapters, sentences, or words, by searching for given token delimiters [14]. Some characters are ambiguous as token delimiters. For instance, a full stop might either denote the end of a sentence or it is part of an abbreviation that should not be split up. The same holds for the whitespace character, which is not as sufficient for separating words as would be expected. Although words are separated by white space, there also exist separate words that should be considered as one token together, such as 'New York'. Since a lot of abbreviations are used in the medical research area, using the correct token delimiters is particularly important for the current research.

Another important pre-processing technique is stemming. Here, the number of unique words is diminished by reducing all words with the same root or stem to a common form. Usually, this is done by removing the derivational and inflectional suffixes from each word [26]. Examples of stemming are reducing plural forms to singular forms, such as 'cars' to 'car', or reducing past simple tenses to simple present tenses, such as 'drove' to 'drive'. After tokenisation, words with an equal word stem can be treated as synonyms. This is referred to as conflation.

In order to diminish the number of features that will be generated at a subsequent stage, replacing synonyms can be useful. Here words will be replaced by a word that is semantically equal. The word 'automobile' will for instance be replaced by 'car', but replacing synonyms also involves

removing capitals by substituting the word 'Car' by 'car'. When replacing synonyms, especially stemmed words, the semantics of the text will slightly change and the syntax of the text will change substantially. Thus, by substituting synonyms specific information will be lost. However, the number of features that the learning algorithm has to process is also diminished. Since most learning algorithms are not capable of processing excessive numbers of features and the remaining information is often sufficient, substituting synonyms is deemed to be a proper solution.

The final pre-processing step that will be discussed here is filtering. Filtering involves the removal of for instance stop words and punctuation. Stop words can be described as words that are not indicative for the subject and therefore not predictive in the classification process, for instance 'the', 'and', or 'of'. By removing these words, the number of generated features will be significantly reduced.

2.2.2 Feature generation

When the texts have been transformed into tokens, in this case words, they have to be transformed into features, since learning algorithms can not cope with unstructured text as input. The aim of feature generation is to translate entities and relationships that are likely to be relevant into features that learning algorithms can handle [14]. The simplest option is using the binary word frequency, in which a term is weighted 1 if the word occurs in the document and 0 if it does not occur. Furthermore, there is a method called "bag of words," in which the text is represented as a set of words combined with their corresponding frequency [2]. Thus, the term weights are calculated based on the assumption that the more frequent a term exists in a text, the more important it is for the classification. However, both methods do not take the direct relationship between terms into account since the weights are not based on the frequencies of word combinations. To be able to use these interrelationships in calculating the weights, n-grams are widely used [5]. Herein, n denotes the length of a sequence of words. If for example the frequency of a word in combination with both neighbours is examined, n is 3. In contrast, the bag of words method uses 1-grams, since only the frequency of single terms is determined. Although using n-grams might produce features that are more predictive, the number of features also increases. Therefore, selecting the most relevant features is even more important when using n-grams.

2.2.3 Feature selection

In order to avoid that an algorithm has to process a large number of features that are irrelevant, the most predictive features should be selected. Several methods can be used to define whether a feature is relevant or not [2]. To start with, the χ^2 test of independence, in which the independency between term t and a particular class is computed, is widely used. The χ^2-statistic for a term t, using binary word frequency, is given by:

$$\chi^2(t) = \frac{n \cdot F(t)^2 \cdot (p_i(t) – P_i)^2}{F(t) \cdot (1 – F(t)) \cdot P_i \cdot (1 – P_i))} \tag{2.3}$$

In this, n is the total number of documents and $p_i(t)$ denotes the conditional probability of class i for documents that contain term t. Moreover, P_i denotes the global fraction of documents that belong to class i, while $F(t)$ is the global fraction of documents that contain term t. A major advantage of this method is that the value is normalised, which ensures that these values are comparable across terms in the same class.

Another method that can be used to quantify the discrimination level of a feature is the Gini-index, which is defined as:

$$G(t) = 1 - \sum_{i=1}^{k} p_i(t)^2 \tag{2.4}$$

Herein, $p_i(t)$ denotes the conditional probability that a document belongs to class i, given that the document contains term t. In case the classification is binary, the formula of the Gini-index can be written in a simplified form:

$$G(t) = p(0|t)p(1|t) = p(0|t)(1 - p(0|t)) \tag{2.5}$$

The Gini index is at maximum if the documents containing term t are evenly distributed among k classes, while a minimal Gini value implies that all documents that contain term t belong to a particular class. Thus, the closer the Gini value is to the minimum, the more predictive a term is for a specific class.

The last method that will be discussed here is information gain, also referred to as entropy. The information gain for a given term t is written as:

$$I(t) = -\sum_{i=1}^{k} P_i \cdot \log(P_i) + F(t) \cdot \sum_{i=1}^{k} p_i(t) \cdot \log(p_i(t)) + (1 – F(t)) \cdot \sum_{i=1}^{k} (1 – p_i(t)) \cdot \log(1 – p_i(t)) \tag{2.6}$$

In this, P_i is the global probability of class i, and $p_i(t)$ is the probability of class i given that the

document contains term t. The fraction of the documents that contain term t is denoted by $F(t)$. The smaller the information gain, the smaller the discriminatory power of term t.

All these values can be used to select the most predictive features in order to avoid that the algorithm has to process multiple irrelevant features. Based on either the χ^2 value, the Gini-index or the entropy, the features are ranked. A cut-off value is determined and only features that have a value higher than the cut-off value will be used as input for the algorithm.

2.2.4 Learning algorithms

After generating and selecting features out of the unstructured texts, input that learning algorithms can cope with is available. The next decision that has to be made is what learning algorithm will be used to solve the classification problem. Learning algorithms are widely used in many domains, but not every algorithm is suitable for each domain. It is possible for a learning algorithm to perform well in one domain, but perform suboptimal in others [6]. Therefore, it is important to select appropriate learning methods in order to attain satisfactory results.

One of the simplest approaches of building an automatic classification problem solver is the rule-based approach. A so-called expert system is built, in which simple 'if-then-else' rules are manually constructed. Although such an approach can achieve good results, it takes substantial time, effort, and domain knowledge to build a high-performance expert system. Moreover, the binary decisions about category membership are rigid and difficult to modify [12]. A more sophisticated strategy is to use inductive learning techniques. Whereas deductive learning is truth-preserving and the classes are predicted based on rules, inductive learning adds information by discovering the rules that best fit the training observations and their corresponding classes.

Logistic regression

In inductive learning, which is an important technique in the field of text mining, a difference is made between linear and non-linear models. Both artificial neural networks and support vector machines are examples of non-linear models. Whereas the dependent variable in a linear regression model is continuous, the dependent variable in a logistic regression model is categorical. A binary logistic regression model is used if there are two categories, whereas a multinomial logistic regression model is used if there are more than two categories. The binary logistic regression model is written

as:

$$P(C = 1|x) = \frac{e^{w_0 + \sum w_i x_i}}{1 + e^{w_0 + \sum w_i x_i}} \tag{2.7}$$

In this, C is the binary class label with value 0 or 1 and $x = (x_1, \ldots, x_p)$ are the features. The coefficients $(w_0, \ldots, w_p)$, can be estimated from the data using the maximum likelihood estimation.

Naive Bayes

The Naive Bayes algorithm is a relatively simple model that constructs probabilistic classifiers based on Bayes' Theorem. This theorem uses prior knowledge of conditions that might be related to an attribute value to describe its probability. In Naive Bayes, the Bayes' Theorem is combined with the independence assumption between features. In other words, it is assumed that the effect of an attribute value on a particular class is independent of the other attribute values. Thus, the presence of a particular feature in a particular class does not depend on the presence of any other feature. According to the Bayesian principle, a case will be assigned to the class that has the largest posterior probability. The posterior probability is written as:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{2.8}$$

In this, $P(c)$ is the prior probability of class c, $P(x|c)$ is the probability of attribute value x given class c, and $P(x)$ is the prior probability of attribute value x. The Naive Bayes algorithm is known for its simplicity and stability. Especially when using a large data set, a naive Bayes classifier is quite accurate. Although the assumption that all attribute values are independent is almost impossible in real life, the accuracy of this algorithm is comparable to the performance of decision trees and neural networks [25].

Neural network

The idea of an artificial neural network is based on biological neural networks. In biology, a neuron 'fires' when a linear combination of inputs exceeds some threshold. Artificial neural networks are used to approximate functions that depend on numerous inputs. The network consists of several layers: the input layer, the hidden layer(s), and the output layer. Nodes, also called units, are connected by links. Each link propagates the activation α_j from node j to i. Moreover, each link has numeric weight $W_{j,i}$, which describes the strength of the connection between the nodes [38]. As said before, an artificial neural network is a non-linear model. Therefore, a node is active

if all correct inputs are given and inactive if at least one of the inputs is incorrect. Otherwise, the model would be a simple linear function. In a feedforward neural network, bidirectional connections between the units do not exist. Therefore it is also called an acyclic model, in which the information is only directed towards one direction. Namely, from the input layer, via the hidden layer(s), to the output layer. In contrast to feedforward models, recurrent models contain links between units from the same layer. The connections between the nodes form a directed cycle, from which the model obtained its name. Assuming that the parameters of the model are selected appropriately, an artificial neural network can be very robust. It does however take considerable time to train this type of network.

Support vector machine

Originally, support vector machines were designed to solve binary classification problems [19]. The observations of the training set are represented in n-dimensional space, so that the examples belonging to different classes can be separated by a hyperplane. As illustrated in Figure 2.3, there is a margin around the hyperplane from the nearest observations of one class to the nearest observation of the other class. The support vectors are the observations that are closest to the margin hyperplane. The optimal hyperplane is described as the hyperplane that causes the widest border. In other words, the maximum margin hyperplane has to be found. In this example, a linear classification is illustrated. However, by applying the kernel trick, a non-linear classification can also be performed.

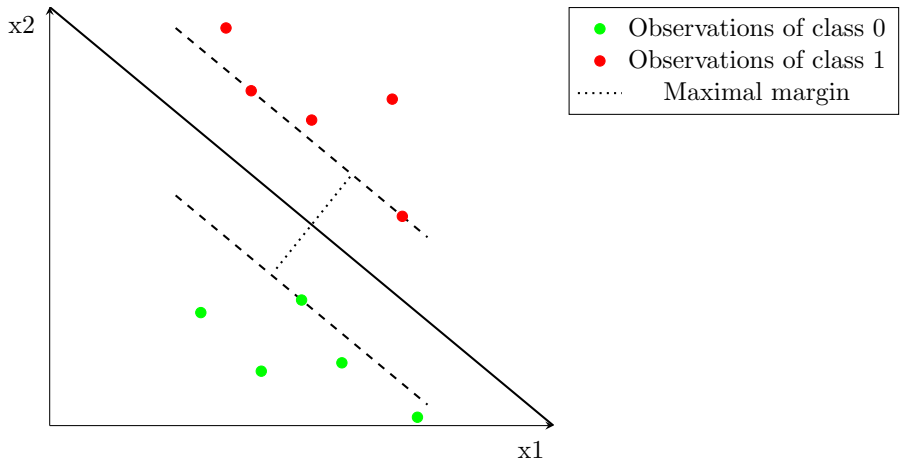

Figure 2.3: Maximal margin hyperplane

Support vector machines are able to produce very accurate classifiers. Furthermore, they are robust for noise and the relative amount of required positive examples is often lower than in other learning algorithms such as neural networks. However, the computation is expensive such that training the algorithm can be time consuming [42].

3 Data

"You can use all the quantitative data you can get, but you still have to distrust it and use your own intelligence and judgment."

Alvin Toffler

Now that the main theoretical background of text mining has been discussed, the data set will be examined here. As said before, medical texts will be analysed in order to define whether a previously assigned activity code was correct or not. Although the final goal is to build an application that is able to determine whether labelling a patient's health trajectory with a specific activity code was justified or not, it is beyond the scope of this thesis to include all activity codes. The aim of this thesis is to build a proof of concept that evaluates whether prior assignments of observations to activity codes were correct or not. The focus is on a specific subset of activity codes, namely activity codes that regard suturing incisions.

3.1 Data set

To define medical activities related to stitching up a wound, three distinct codes can be chosen by physicians. To be specific, activity code 38941, 38942, or 38943 should be used when an incision is sutured. However, to assign these codes to a patient's health trajectory, more detailed medical activities are required, which are explained by the NZa. To start with, the physician has to both inspect and disinfect the wound. Moreover, the wound has to be sutured or excision has to be performed, combined with local anaesthesia. Summarising, there are 4 requirements: inspection, disinfection, local anaesthesia, and suturing. Although the required medical activities are exactly the same for these three codes and one of the three codes is justified if all requirements are met, the situation in which the patient was treated is decisive. Activity code 38942 is about inpatient care. In contrast, both code 38941 and 38943 concern treatments performed in an outpatient department of a hospital. The difference between those two being whether or not the patient was referred to

the hospital or not. In outpatient care, the information that is required for the classification can be found in forms that are filled in when treating a patient at the Emergency Room (ER). Since both code 38941 and 38943 require the same medical activities in outpatient treatments, and the difference in referral can not be found in the texts that will be analysed, the information of the Emergency Room treatment of these two codes will be combined.

To extract the relevant information from a set of 146.814 Emergency Room forms, a list of activity codes that were assigned to patients is used. From this list, all patient numbers along with the treatment date that corresponded to either code 38941 and 38943 were selected. Next, the ER forms of these patients were selected from the total set of ER forms. However, since the activity code is not stored in the ER form and patients might have been treated at the ER for different injuries as well, the ER forms corresponding to other activity codes were also selected in some cases. Therefore, the ER forms corresponding to the correct treatment dates were extracted. Subsequently, the patient numbers were removed from the selected ER forms in order to anonymise the data. The patient numbers are unnecessary in the analysis and this way, it is not possible to deduce the privacy sensitive information to the patients. As can be seen in Table 3.1, a total of 2.659 patients have been assigned either activity code 38941 or 38943. Respectively 1.134 and 1.326 corresponding ER forms could be extracted from the total set of ER forms. Thus, the data set that will be used in this thesis contains 2.460 Emergency Room forms of patients that have been treated for stitching up wounds.

Activity code	**Description**	**Patients**	**ER forms**
38941	Outpatient care, without referral	1.224	1.134
38943	Outpatient care, with referral	1.385	1.326

Table 3.1: Distribution of activity codes

An Emergency Room form is for instance used to inform a patient's general practitioner (GP) after an ER visit and therefore it is written as a letter. However, the information that such a letter consists of is automatically extracted from the electronic health record. Since the structure of these fields contains information that might be useful in classification, not the entire letters but the separate text fields will be extracted from the EHR. Thus, the structure of the text fields is retained. However, it is not uncommon that not all fields are filled in. Therefore, an overview of the number of missing attributes is given in Table 3.2.

Text field	Missing	Percentage
Memo	9	0.4
Anamnesis	0	0.0
Examination	0	0.0
Prescribed medication	1533	62.3
Diagnosis	2	0.1
Therapy	2	0.1
Additional information for GP	2453	99.7
Further treatment	2460	100.0
Occasion	68	2.8

Table 3.2: Available text fields of ER forms

3.2 Class label distribution

As mentioned above, the current data set merely includes examples that were assigned to code 38941 or 38943. These classifications are done by the physicians themselves, but when the health insurance company asks for proof, the classifications are checked by other employees of the hospital. Checking the assigned activity codes is done manually, by searching for the requirements in the text of the electronic health record. The outcomes of these checks are used as the class labels in the current data set. Thus, for each observation it is decided whether or not the requirements were met. If it became clear from the texts that the wound was inspected, disinfected and stitched up with the use of local anaesthesia, then the example was labelled as correct. However, if not all requirements could be found, the observation was classified as incorrect. This is for instance the case if the wound is glued instead of stitched up, or if the patient did not want local anaesthesia for example because of pregnancy. It should be noted that it is not always immediately clear from the medical text whether the requirements have been met because physicians can describe the performed treatment in many different ways of varying clarity. Therefore, human interpretation is an important factor in checking the correctness of the assigned activity codes. In many cases, the essential information was found in the *Therapy* text field and the decision could be made after just investigating this field of text. However, there were also cases in which the text regarding the therapy was inconclusive and other text fields had to be investigated as well.

In addition to the class label, the reason of rejection is listed as well when the observation was

classified as incorrect. As can be seen in Table 3.3, the most common reason of rejection is that the wound is glued together instead of stitched up. In total, 8 reasons of rejection were given. However, since some reasons do not occur often, they are paired up into three global categories. The first being that no local anaesthesia is used. The second category consists of cases in which the wound is not sutured, but glue, staples or plasters were used instead of stitches. The last group consists of observations in which the wound was, for varying reasons, not treated at all.

Class label	**Reason of rejection**	**Number**	**Total**	Percentage
No anaesthesia	*Just one suture*	111	126	12.9
	Anaesthesia not desired	15		
No stitches	*Glued*	644	761	77.9
	Stapeled	67		
	Plaster sticked	50		
Wound not treated	*Just disinfected or bandaged*	52	90	9.2
	Expectative	25		
	Translocated	13		

Table 3.3: Reasons of rejecting assigned activity code

Even more important than the general statistics of the data is the distribution of correctly and incorrectly classified cases. As can be seen in Table 3.4, 60.3 percent of the cases in the data set is labelled as correct, which means that the activity code was justified. A staggering 39.7 percent of the cases is assigned to the incorrect class, which means that the activity code was not justified based on the textual information in the EHR. There exist cases in which it is possible that an observation is rejected since it did not meet the requirements of a specific activity code, but then it can be assigned to another activity code because it does have all requirements for that activity code. However, for the current activity codes, it is an all-or-nothing decision. If the activity code was unjustified, the hospital has to refund the money they received because there are no other possible activity codes that might fit.

Since this data set only contains observations that were assigned to the codes corresponding to suturing, only true and false positives could be detected in the set of cases that were classified by physicians. However, a data set that contains observations that should have been assigned to this code instead of the code that they were assigned to is not available, and thus it is not possible to detect true and false negatives.

Classification	Number	Percentage
Correct	1485	60.3
Incorrect	977	39.7

Table 3.4: Distribution of binary class labels

3.3 Most frequent words

Since the word frequencies are very important in text mining, in Table 3.5 the top 10 of words that occur most frequently in the examined data set are shown. In addition to the most frequent words of the whole data set, the most frequent words for both the correct and the incorrect class are given as well. The difference between these lists can be very informative.

All classes		**Correctly assigned**		**Incorrectly assigned**	
Word	*Frequency*	*Word*	*Frequency*	*Word*	*Frequency*
wond	4439	wond	3188	wond	1251
hand	2846	hand	2066	hoofd	1229
gevallen	2367	intact	1515	wondje	1124
hoofd	2128	gevallen	1324	gevallen	1043
intact	2063	snijwond	1300	hand	780
snijwond	1892	ethilon	1180	snijwond	592
tetanus	1588	hechtingen	1143	intact	548
wondje	1497	tetanus	1107	bewustzijn	483
huisarts	1357	huisarts	1013	tetanus	481
ethilon	1267	gehecht	992	geplakt	471

Table 3.5: Most frequent words in EHR descriptions

As can be seen in Table 3.5, the word *wond* occurs most frequently, namely 4439 times. Since this is the number of times that the word is written down in the EHR forms and not the number of EHR forms that the word exists in, this number can be higher than the total number of EHR forms. The differences between the top features of the correctly assigned class (1) and the incorrectly assigned class (0) seem to be informative. To start with, the word *wondje* is in the top 10 of the incorrectly assigned class, but not in the top 10 of the correctly assigned class. This might be due

to the fact that smaller wounds are more often glued or stapled instead of stitched up and thus the activity code was incorect. The same holds for the word *hoofd*. A wound on the head will more frequently be glued instead of sutured to diminish the risk of scars. On the other hand, the word *gehecht* logically occurs more often in the correctly assigned class instead of the incorrectly assigned class since it explicitly states that a wound is stitched up. Since *ethilon* is the brand name of nylon sutures, the occurrence of this words indicates that the wound was sutured as well.

4 Application

"Artificial Intelligence is the attempt to make computers do what people think computers cannot do."

Douglas Baker

Before, both the underlying theory of text mining and classification using different algorithms, and the data set that is used in this thesis have been discussed. Now, applying the aforementioned algorithms to the data and creating an application that conducts the experiment will be discussed.

4.1 Pre-processing

The general idea of pre-processing textual data has already been discussed in Chapter 3. It is important to execute the steps of the pre-processing phase in right order, since prior modifications may affect further transformations. Figure 4.1 visualises the sequence of executed tasks.

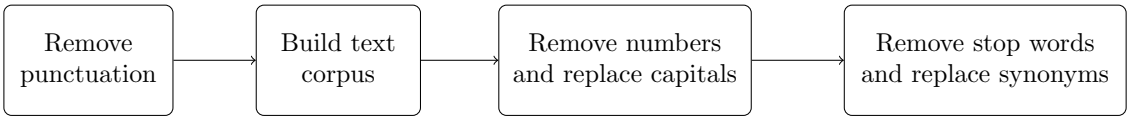

Figure 4.1: Sequence of executed pre-processing tasks

First, punctuation is replaced by white space. Although it would be obvious to simply remove punctuation, this was not sufficient for the current data because punctuation was extensively used without white space around it. Therefore, simply removing punctuation would cause merged words, while they had to remain separate. Adding white space in order to prevent words from being merged does not have a negative impact, since overflowing white space will be removed in a subsequent stage regardless. After removing punctuation, a text corpus, a structured set of texts, was made. Subsequently, numbers are removed and capital letters replaced by lower case letters. At first sight, some numbers in the text seemed to be important, for instance if they said something about the

number of stitches. However, too many numbers were used in a non-predictive way and therefore the choice was made to remove them all. Subsequently, stop words are removed and synonyms are replaced. The list of stop words can be found in Appendix I, and the list of synonyms can be found in Appendix II. For both of these steps, it is important that the capital letters already have been replaced by lower case letters, since searching for stop words and synonyms is case sensitive. The lists of stop words and synonyms are constructed manually, both based on a list of word frequencies. Words that occurred in more than 50 ER forms and that were not assumed to be informative were added to the list of stop words. Since the stop words in medical texts are quite different from stop words in general texts, an existing list of stop words could not be used for this purpose. Therefore, the list of stop words is constructed manually. Unfortunately, the same holds for medical synonyms. A general list of medical synonyms could not be found in Dutch, so the list of words that occurred in more than 50 ER forms was manually checked for synonyms. Thus, this solution is sufficient for the current research, but it makes it less applicable for other research problems.

Stemming is not performed because all medical texts were written in Dutch and the stemming functions in R are not sufficient to solve Dutch stemming problems. Moreover, stemming might delete predictive features because the difference between two words will disappear. This would for instance be disadvantageous if the word 'hechtingen' is made equal to 'hechting'. The difference between these two words might be very important since one suture is performed without anaesthesia and more than one suture is performed with the use of local anaesthesia. Therefore, this is an example of a potentially distinguishing feature between a correct and an incorrect assignment of the activity code. Another example in which stemming would remove the distinctiveness of features is the difference between 'wond' and 'wondje. 'Wondje' indicates that the wound is small, which will often lead to glueing or stapling the wound instead of using stitches, while 'wond' indicates a larger wound that is more likely to be sutured. Since using local anaesthesia is one of the requirements, removing the difference between singular words and plural words might impede assigning a case to the correct class. Besides, typographical errors are not replaced, since this is inherent to the research problem. Correcting typographical errors would take lots of time and effort, while this application should be used to save time instead. Thus, since typographical errors will occur frequently in ER forms, the algorithm should be able to handle these errors.

4.2 Feature generation and selection

After the corpus is made and transformed, a Document Term Matrix (DTM) is made. This is a matrix in which each row corresponds to a document, in this case an EHR form, and each column

corresponds to a term. Then, in case the 1-gram method is used, the frequency of each term is given for each document. In case a 2-gram method is used, the frequency of every word combination is given for each document. In contrast to the term frequencies, binary term frequencies can also be given. In this binary form, it is merely shown whether or not a term (or combination of terms) is present in a document. It was empirically found that using the binary form of the DTM gave the same results as using the DTM in which the frequency was given. Since using the binary form of the DTM is less complex and therefore faster than using the frequency DTM, it was decided to use the binary form of the DTM.

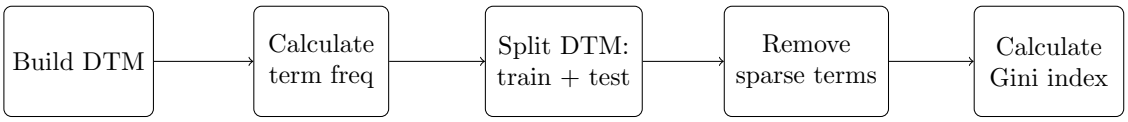

Figure 4.2: Sequence of executed feature generation and feature selection tasks

The DTM is randomly divided into a training set and a test set. Then, sparse terms are removed from the training set, which means that words that do not occur in more than 5% of the documents are not recorded in the DTM. This percentage of 5% might seem very low. However, it was empirically found that this was the most fitting parameter value. A possible explanation for this low value is that some strongly predictive features may not occur frequently. Therefore, taking a higher sparsity value would inadvertently remove these features. Using a low sparsity value is not detrimental, since removing sparse terms is just the first filtering step and remaining non-predictive sparse features will be removed in later stages. The only downside of this approach is that it may slow down the process.

Subsequently, the Gini index was calculated and used to select the most predictive features. The Gini index is at maximum when all observations are evenly distributed over the classes, and it is at minimum when all observations belong to just one single class. Thus, a minimal gini index indicates a strong relationship between the feature and the class label. Only the top k features with the strongest relationship are selected. This cut-off is set to that value that leads to the best accuracy. This value is determined by applying cross-validation, which will be discussed in more detail in Chapter 5. Based on the terms that remain in the DTM of the training set, the document term matrix of the test set is made. Thus, the documents and frequencies of these matrices differ, but the terms of which the frequency is given are exactly the same. Otherwise, if all terms of the test set were also taken into account with the feature selection, the test set is also used to train the algorithm.

As mentioned before, the problem of a skewed class distribution could be solved in three ways. First, examples of the majority class could be removed. Second, cases of the minority class(es) could be added or duplicated. Both of these manners affect the absolute number of cases in order to influence the class distribution. The last way of modifying the class distribution is cost sensitive learning, in which each example gets a weight, based on the relative amount of cases in its class. In the current thesis, the third manner is chosen. In the binary classification part of the problem, the desired class distribution is 50/50. Since 60.3% of the cases is assigned as correctly, each example of this class has a weight of 0.82 and each example of the incorrectly assigned class has a weight of 1.27. In the part of the problem in which the reasons of rejection are predicted as well, each class should have a relative size of 25%. The cases in the correctly assigned class have a weight of *0.41*. The cases of no anaesthesia, no stitches, and wound not treated respectively have a weight of *4.88, 0.81*, and *6.91*.

4.3 Predicting the class label

After generating the features and selecting the most predictive ones, the actual class label can be predicted. As mentioned before, several algorithms can be used to execute this task. In this thesis a logistic regression model, Naive Bayes algorithm, support vector machine, and a neural network are used. All algorithms use the DTM of the training set combined with the class labels of these training examples to predict the class labels of the test set. The Naive Bayes algorithm, which is in the library *e1707*, does not require any other information. The posterior probabilities of the categorial class variable will be be calculated from the independent predictor variables in the training set. The predicted class label is the class with the highest posterior probability. However, the logistic regression model, from the package *glmnet*, requires some additional information. Namely, the parameter called *family*, which is the response type of the evaluated distribution. In case of the binary classification, this parameter has to be set to *binomial*. Then, a binomial logistic regression model is fitted for the log-odds. In case the reasons of rejection are predicted as well, the parameter has to be set to *multinomial*, which will fit a logistic multinomial regression model [16]. The support vector machine algorithm, *svm* from the package *e1701*, is used with a radial kernel and therefore requires a cost and gamma value in addition to the training data. The former is a general penalising parameter for this kind of classification, and the latter is the radial basic function specific kernel parameter [28]. The last learning algorithm, the neural network, which is in the package *nnet*, requires *size* and *decay* as parameters. The size indicates the number of units in the hidden layer, and decay is the weight decay [37].

While some parameters, for instance the response type for the logistic regression model, can merely take one value that is known in advance, the optimal value of other parameters can be dependent of the particular situation. When searching for the optimal parameter values, it is important that the values are only based on the training set and not on the test set. Therefore, cross-validation is applied on the training set and subsequently, the algorithms are tested with the optimal parameter settings on the test set. In this way, it is prevented that the parameter values are fitted on the test data as well. The same holds for selecting the most predictive features. As mentioned before, the top k-features are selected. The question however is what value of k would give the best results. Therefore, cross-validation on the training data is also used to find the best cut-off value for the Gini index.

5 Experimental setup

"The key to Artificial Intelligence has always been the representation."

Jeff Hawkins

As Witten and Frank [43] describe, comparing performances of different machine learning methods on a given classification problem is not as easy as it sounds. To make sure that apparent differences are not caused by chance effects, a sophisticated experimental setup and statistical tests are needed. Just comparing the estimated error is not quite sufficient since this difference may simply be caused by estimation errors. Besides, it may also be important to determine whether one learning method is really better than another on a particular problem. In the current research, it might be the case that the best learning algorithm for the binary classification problem is not the same as the best algorithm for the classification problem in which the reasons of rejection are predicted as well.

5.1 Training and testing

The most common way to divide a data set into a training and test set, is to use approximately 2/3 of the data to train and the remaining 1/3 to test. Thus, in this case the training set consists of 1640 examples and the test set contains 820 examples. Cross-validation helps more effective use of the data set and therefore, 4-fold cross-validation is applied on the training set. With the use of cross-validation, the optimal parameter settings are determined. To start with, the optimal cut-off of the Gini-index is determined. In other words, it is determined which top-k features should be selected to expect the best results. The Gini index is a number between zero and one, and it is increased with steps of 0.1. To be able to properly predict the accuracy, 4-fold cross-validation is applied and therefore four accuracies on the training data for each Gini border can be calculated. Then, the average accuracy on the training data for each Gini border is computed and the value with the highest average accuracy on the cross-validated training data is selected. Furthermore,

the cost and gamma values for the support vector machine, and the number of nodes in the hidden layer and the maximum number of iterations for the neural network are determined. In case of the general linear model and the Naive Bayes algorithm, merely the best expected Gini border is selected. However, for the support vector machine and the neural network, other parameters are meaningful as well. In these cases, the optimal set of both the Gini border and the other required parameter values is selected. For the support vector machine, a radial kernel was chosen and thus, the values of the cost and gamma had to be determined. These values are respectively chosen between 0.1 and 10, and between 0 and 2. The values of the number of nodes in the hidden layer and the decay for the neural network are respectively chosen between 0 and 20, and between 0 and 0.001. The maximum number of iterations, *maxit*, is set to 500 and skip layer connections are permitted. After applying cross-validation on the training set to select the optimal parameter settings, the algorithms can be trained on the whole training set with the selected parameter values as input and subsequently it can be tested on the test set.

In case of the binary classification, the algorithm is trained to predict whether or not the activity code of an example was correct. When testing the algorithm, the predictions of the algorithm are compared to the actual class labels and the accuracy of the algorithm can be calculated. Although predicting the reason for rejection as well is a bit more complicated, the experimental setup is equal to the setup of the binary classification problem. The algorithm is trained to assign an example to one of the four categories. In the testing phase, the predictions of the algorithm are compared to the actual class labels. Thus, a non-hierarchical approach was used. In other words, the examples are directly divided into the four classes. In contrast to this, the hierarchical approach would have been that the algorithm first predicted whether or not the activity code was correct and then, if the activity code was predicted as incorrect, it predicted the reason for rejecting the activity code. In Figure 5.1, the non-hierarchical model is schematically shown and in Figure 5.2, the hierarchical model is illustrated.

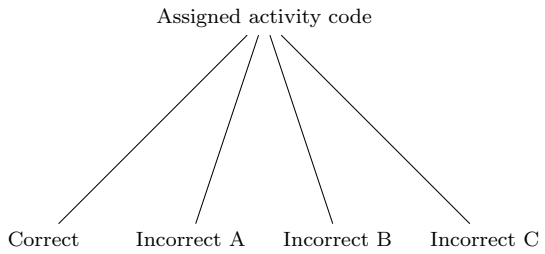

Figure 5.1: Non-hierarchical model

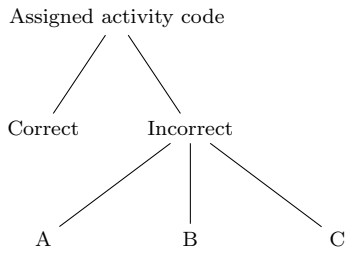

Figure 5.2: Hierarchical model

Since the features that are predictive for a case being correct or incorrect could differ from the features that are decisive in predicting the reason for rejecting a case, it was expected that the hierarchical approach would be more beneficial than the non-hierarchical approach. However, preliminary experiments showed that this was not the case for the current classification problem. Therefore, it was chosen not to extend the experiment with the hierarchical approach.

As explained in Chapter 3, each EHR form consists of several fields of text. Although in data mining it often seems to be the case that the more information the better, observing the process of manually labelling this data set has shown the tendency that examining a particular text field might be more beneficial than examining all information. As said, the current data set is labelled manually. In a considerable number of these cases, the information that was required to assign an example to a category appeared to be in the therapy text field. It might be the case that the most important information is in this particular text field and the information in the other text fields is not valuable or it might even deteriorate the results. As the name indicates, this text field is about the performed medical treatment. Since all four requirements (inspection, disinfection, local anaesthesia, and suturing) are part of the treatment, it seems to be convenient that this information can be found in this particular text field. However, sometimes significant information had to be stored in another text field, or a physician failed to comply with the structure of the text fields. Although the information in the other text fields was only needed in a minority of the examples, it would be of great value to know whether or not a machine learning algorithm needs all information there is or that is just needs the part that seems to be most valuable for humans. To be able to compare these situations, the learning algorithms are trained and tested on both the whole data set and the data set in which only the therapy text field was selected.

5.2 Balance between accuracy and human work

Up until now, it was proposed that the application would entirely replace human work in classifying the activity codes corresponding to the text in ER forms. However, the accuracy of the model could be increased if cases are redirected to a human if the confidence of the classification is below a certain value. Combining the power of both humans and computers can lead to the best results. When expanding the model with the possibility to address classifications to a human, the balance between the number of correct classifications and the amount of residual work is important. The accuracy of classifications should be maximised, while simultaneously the number of cases that has to be categorised manually should be minimised. The level of confidence can be used to decide whether or not a human opinion is necessary. Thus, a boundary should be set for the algorithm

to know when to redirect a case. In this, there are two extremes possible. On the one hand, the boundary is set in a way that every case is redirected. In other words, even if the algorithm is 100% sure that the classification is correct, the case will be redirected. This will lead to a 100% accuracy, assuming that humans categorise everything correctly. Nevertheless, the amount of human work will not decrease compared to the amount of work without using the algorithm. On the other hand, the boundary can be set in a way that not a single case will be redirected. Thus, even if the model has a very low level of confidence, the case will not be redirected. This might lead to a low accuracy, but no human work remains. The question now is what the optimal split is. When does the algorithm have to redirect a case, based on it's level of confidence? The optimal balance should be found in order to reduce the amount of human work and simultaneously maximise the number of correct classifications.

6 Results

"Intelligence is the art of good guesswork."

Horace Barlow

6.1 Binary classification

6.1.1 Performance with feature selection

In Table 6.1 the results of the binary classification are given. For each algorithm, the accuracy is given for both the model in which all text fields are analysed and the model in which only the therapy text field is included. In the last column, the p-values of the binomial probability test are shown to define whether there was a significanct difference in performance between analysing all text fields and analysing the therapy text field.

	All text fields	**Therapy text field**	**p-value**
Binomial Logistic Regression	0.910	0.922	0.110
Naive Bayes	0.905	0.915	0.200
Support Vector Machine	0.915	0.918	0.743
Neural Network	*0.926*	*0.933*	0.405

Table 6.1: Performance on the test set for binary classification. P-value $*$ indicates a significant difference between all text fields and the therapy text field

The best accuracy, 0.933, is achieved when a neural network is used to analyse only the therapy text field. For analysing all text fields, a neural network also yields the best result with an accuracy of 0.926. To compare performances, statistical tests were performed with $\alpha = 0.05$. Although for all models the performance of analysing the therapy text field was better than the performance of analysing all text fields, no significant differences were found.

For both approaches, analysing all text fields and analysing the therapy text field, the neural network model yields the best performance. For analysing all text fields, the neural network model performed significantly better than the binomial logistic regression model and the Naive Bayes model. In case only the therapy text field was included, the neural network model performed significantly better than all other models. An overview of all p-values of the performed binomial probability tests is shown in Table 6.2

Model 1	**Model 2**	**All**	**Therapy**
Binomial Logistic Regression	**Naive Bayes**	0.597	0.327
Binomial Logistic Regression	**Support Vector Machine**	0.585	0.711
Binomial Logistic Regression	**Neural Network**	0.024*	0.049*
Naive Bayes	**Support Vector Machine**	0.302	0.250
Naive Bayes	**Neural Network**	0.012*	0.003*
Support Vector Machine	**Neural Network**	0.233	0.012*

Table 6.2: P-values of comparing all models for binary classification. P-value $*$ indicates a significant difference between model 1 and model 2.

To analyse the errors of the model, in Table 6.3 and Table 6.4 the confusion matrices of the best performing models are given. For analysing all text fields as well as for analysing just the therapy text field, the numbers of false positives and false negatives are almost the same. In other words, if the model is used to check whether the assigned activity code was correct or not, both approving incorrectly assigned activity codes and rejecting correctly assigned activity codes occur almost as often.

		Predicted	
		Correct	*Incorrect*
Label	*Correct*	439	25
	Incorrect	36	320

Table 6.3: Confusion matrix of the binary classification for all text fields

		Predicted	
		Correct	*Incorrect*
Label	*Correct*	445	25
	Incorrect	30	320

Table 6.4: Confusion matrix of the binary classification for the therapy text field

6.1.2 Performance without feature selection

Since selecting the most predictive features could be more important than selecting the best fitting algorithm, it is also interesting to compare the above mentioned accuracies in which feature selection is applied, with the performance of models in which no feature selection is applied. In

Table 6.5, the results of all models without feature selection are shown, along with the p-value of the comparison between feature selection and no feature selection.

	All	p-value	Therapy	p-value
Binomial Logistic Regression	0.851	$1.66e^{-6*}$	0.920	0.832
Naive Bayes	0.862	$3.88e^{-5*}$	0.889	0.001^{*}
Support Vector Machine	0.872	$1.11e^{-5*}$	0.929	0.093
Neural Network	0.900	0.012^{*}	0.923	0.115

Table 6.5: Performance on test set for binary classification without feature selection. P-value $*$ indicates a significant difference between feature selection and no feature selection.

For analysing all text fields, all models that select the most predictive features have a significantly better performance than the models that do not use feature selection. However, for analysing the therapy text field this significant difference was only found for the Naive Bayes model.

6.1.3 Comparing features

If feature selection is not applied, the number of features that is used as input is 242 in case all text fields are included and 49 in case the therapy text field is analysed. In Table 6.6, the number of features in case feature selection is applied is shown for each algorithm. The list of all features for each model can be found in Appendix III.

	All text fields	Therapy text field
Binomial Logistic Regression	15	14
Naive Bayes	15	7
Support Vector Machine	56	7
Neural Network	15	14

Table 6.6: Number of features used as input for binary classification

In Table 6.7, the coefficients of the binomial logistic regression model are given. Since all features have the same possible values, namely 0 or 1, the coefficients are standardised and can therefore be used to compare the importance of the features.

Feature	**Coefficient all**
Intercept	-2.083
Hechtingen	3.812
Gelijmd	-3.482
Ethilon	2.473
Lidocaine	2.462
Klinibond	-2.316
Vicryl	1.787
Gehecht	1.686
Geplakt	-1.680
Hechten	1.389
Augmentin	1.113
Distaal	1.071
Hechting	-0.891
Timmers	0.714
Verwijderen	-0.641
Verdoving	0.086

Table 6.7: Binomial logistic regression model for all text fields

Feature	**Coefficient**
Intercept	-2.278
Hechtingen	4.224
Gehecht	2.871
Ethilon	2.608
Lidocaine	2.479
Klinibond	-2.320
Vicryl	1.893
Geplakt	-1.687
Hechten	1.450
Augmentin	0.890
Timmers	0.821
Verwijderen	-0.809
Dagen	0.687
Hechting	-0.615
Maal	0.544

Table 6.8: Binomial logistic regression model for the therapy text field

As shown in Table 6.8, the binomial logistic regression model for the therapy text uses almost the same features as for all text fields. Features that have a positive sign are most predictive for the correct class, while features that have a negative sign are most predictive for the incorrect class. *Distaal*, *gelijmd* and *verdoving* are not used when the therapy text field is analysed, while *dagen*, and *maal* are not used in case all text fields are analysed. The features used in both models differ in magnitude, but the sign is always the same. According to the coefficients, *hechtingen*, *gelijmd*, and *ethilon* are the most important features in case all text fields are included. In case only the therapy text field is analysed, the most important features are *hechtingen*, *gehecht*, and *ethilon*. According to both the sign and the magnitude of the coefficients, *hechtingen* is the most important feature for the correct class for both analysing all text fields and analysing the therapy text fields. For the incorrect class, *gelijmd* is most predictive in case all text fields are included, while *klinibond* is most predictive when only the therapy text field is analysed.

6.2 Multi-class classification

6.2.1 Performance with feature selection

In Table 6.9, the accuracies of the multi-class classification are given. A sifnigicant difference between all text fields and the therapy text field was only found for the Naive Bayes and the support vector machine models.

	All text fields	**Therapy text field**	**p-value**
Multinomial Logistic Regression	0.751	0.728	0.224
Naive Bayes	0.791	*0.874*	$9.29e^{-11*}$
Support Vector Machine	*0.830*	0.862	0.001^{*}
Neural Network	0.820	0.815	0.794

Table 6.9: Performance on the test set for multi-class classification. P-value $*$ indicates a significant difference between all text fields and the therapy text field.

With the exception of between the support vector machine model and the neural network, significant differences were found between all models in case all text fields were included. In case only the therapy text field was included, significant differences were found between all models, except for comparing the Naive Bayes and the support vector machine models. In Table 6.10, an overview of all p-values of the performed binomial probability test is shown.

Model 1	**Model 2**	**All**	**Therapy**
Multinomial Logistic Regression	**Naive Bayes**	0.014^{*}	$6.91e^{-20*}$
Multinomial Logistic Regression	**Support Vector Machine**	$4.66e^{-9*}$	$3.12e^{-15*}$
Multinomial Logistic Regression	**Neural Network**	$1.46e^{-5*}$	$4.43e^{-10*}$
Naive Bayes	**Support Vector Machine**	0.004^{*}	0.164
Naive Bayes	**Neural Network**	0.043^{*}	$2.96e^{-6*}$
Support Vector Machine	**Neural Network**	0.444	0.001^{*}

Table 6.10: P-values of comparing all models for the multi-class classification. P-value $*$ indicates a significant difference between model 1 and model 2.

In Table 6.11, for all text fields, and Table 6.12, for the therapy text field, the confusion matrices of the best performing models are given. It is notable that no case is predicted as 'wound not treated'.

		Label			
		No anaesthesia	*No stitches*	*Wound not treated*	*Correct*
Predicted	*No anaesthesia*	5	0	0	2
	No stitches	6	233	30	30
	Wound not treated	0	0	0	0
	Correct	38	30	3	443

Table 6.11: Confusion matrix of the multi-class classification for all text fields

		Label			
		No anaesthesia	*No stitches*	*Wound not treated*	*Correct*
Predicted	*No anaesthesia*	27	0	0	7
	No stitches	4	247	33	25
	Wound not treated	0	0	0	0
	Correct	18	16	0	443

Table 6.12: Confusion matrix of the multi-class classification for the therapy text fields

6.2.2 Performance without feature selection

As well as for the binary classification, comparing the results of models that use feature selection and models that do not use feature selection is interesting for the multi-class classification. In Table 6.13, the accuracies of all models without feature selection are given along with the the p-values of the comparison between applying feature selection and no feature selection.

	All	**p-value**	**Therapy**	**p-value**
Multinomial Logistic Regression	0.751	1.000	0.727	1.000
Naive Bayes	0.780	0.049*	0.861	0.061
Support Vector Machine	0.827	1.000	0.867	0.219
Neural Network	0.804	0.241	0.801	0.185

Table 6.13: Performance on test set for multi-class classification without feature selection. P-value $*$ indicates significant difference between feature selection and no feature selection.

In contrast to the binary classification, no significant differences between feature selection and no feature selection when analysing all text fields were found, except for the Naive Bayes model. In

case only the therapy text field is analysed, no significant differences were found between feature selection and no feature selection for any model.

6.2.3 Comparing features

In case feature selection is not applied, the total number of features is 242 for all text fields and 49 for the therapy text field. Table 6.14 shows the number of features in case feature selection is applied for each algorithm. In Appendix IV, the features itself are listed.

	All text fields	**Therapy text field**
Multinomial Logistic Regression	231	49
Naive Bayes	231	11
Support Vector Machine	231	49
Neural Network	115	49

Table 6.14: Number of features used as input for multi-class classification

For both all text fields and the therapy text field, the multinomial logistic regression gives the worst performance. However, the number of features that is used as input is similar to the input of other models. The model that performs best, the Naive Bayes model, uses the fewest features as input.

6.3 Optimal balance between performance and human work

As mentioned before, expanding models with the opportunity to redirect cases to humans can increase the accuracy of the model. In Figure 6.1 and Figure 6.2, the balance between human work and the accuracy of the model is shown. Both graphs are based on the best performing model. On the x-axis, the percentage of manual work is shown, and on the y-axis the performance can be seen. If all cases are classified by the algorithm, the level of confidence cut-off is 0, and no manual work has to be done. In that case the performance is the same as the performance as given above. For the binary classification this means that the accuracy is 92.6% in case no classifications are done by hand and for the multi-class classifications the accuracy is 84.3%. If no case is classified by the algorithm, both the amount of work and the accuracy will be 100%, assuming that a human makes no mistakes. Increasing the level of confidence cut-off means that the amount of manual work increases as well. In the graphs, the amount of manual work, corresponding to a given level of confidence cut-off, is

plotted against the performance of the algorithm.

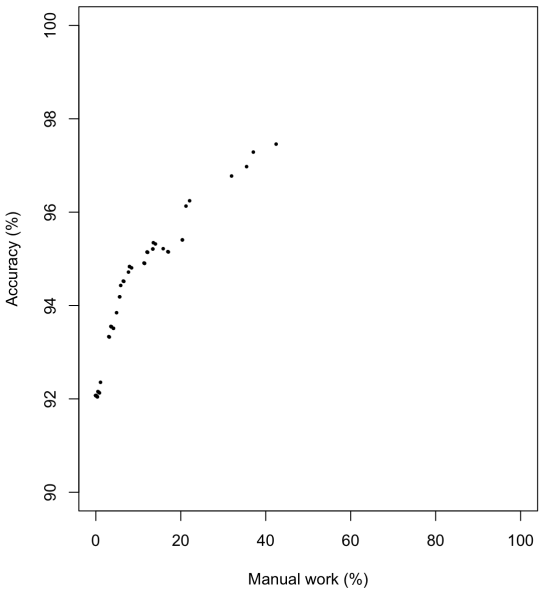

Figure 6.1: Manual work against accuracy for binary classification with neural network

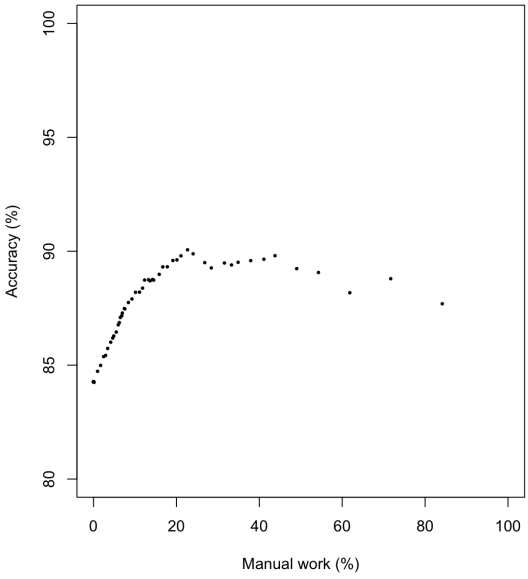

Figure 6.2: Manual work against accuracy for multi classification with support vector machine

If the optimal cut-off value is defined as a maximal increase in accuracy with a minimal increase of manual work, for the binary classification the optimal cut-off value seems to be 0.87. This leads to an accuracy of 95.3% corresponding to a manual amount of work of 13.5%. For the multi-class classification, the optimal cut-off value is 0.74, corresponding to an accuracy of 88.7% and a manual amount of work of 12.3%. However, determining the exact optimal balance between performance and human work is up to the hospital. It depends on both the adverse effects of assigning cases incorrectly and the desire to diminish the amount of human work.

7 Conclusion & Discussion

"By far the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."

Eliezer Yudkowsky

7.1 Performance on the binary classification problem

The main research question of this thesis was: *Is it possible to develop an automatic classification system, using supervised machine learning algorithms, that determines whether an activity code was assigned correctly or not, based on digital unstructured text stored in the electronic health record?* The main research question of this thesis assessed the binary classification problem. When analysing all text fields, as was the initial approach, the best performance that a model achieved was an accuracy of 0.926. This performance level affirmed that it is possible to develop an automatic classification system that determines whether or not an activity code was assigned correctly. Moreover, in case not all text fields but only the therapy text field is analysed, the performance increased to an accuracy of 0.933.

Although such high accuracy reveals opportunities for the future, this performance is possibly not proficient enough to convince health insurance companies that a machine learning algorithm can replace humans in checking activity codes. However, making mistakes is inherent to being human. Therefore, it should be questioned what the average error rate of a human being is. It would have been ideal if the error rate of the manually constructed data set had been available. Unfortunately, this was not the case and hence it remains unknown whether manual and automated error rates differ significantly or not. Besides, it is nearly impossible that the manually constructed data set that was used is free of errors. Therefore, it can be safely assumed that the algorithm was trained on a data set that contained some errors and hence did not reach its full potential yet. Moreover, the confusion matrix can be used to convince health insurance companies. Although mistakes were made by the

algorithm, the number of false positives and false negatives were almost the same. Therefore, health insurance companies pay for incorrect assigned activity codes, but they also save money because hospitals have to refund the money they received for correctly assigned activity codes. The mistakes balance each other out.

7.2 Performance on the multi-class classification

In addition to the binary classification, it was also investigated whether the reason of rejection for incorrectly assigned activity codes could be predicted as well. This question expands the binary classification problem to a multi-class classification. Although the number of labels increased from 2 to 4, the performance remained promising. In case only the therapy text field was analysed, the Naive Bayes model performed best with an accuracy of 0.874. When all text fields were included the best accuracy was 0.830, achieved by a support vector machine model. Therefore, it can also be concluded that it is possible to build a classification system that is able to predict the reason of rejection for incorrectly assigned cases. While hospitals use the reasons for rejection as additional information, they are not required to deliver it to health insurance companies. This part of the research was hence conducted to see whether or not we could deliver this supportive information. An accuracy of 0.874 is sufficiently high to be informative.

The best performing model for all text fields and the optimal one for the therapy text field both never predicted a case as *wound not treated*. At first glance, it might seem that this was caused by not using appropriate weights to solve the skewed class distribution problem. However, the *no anaesthesia* class is also very small, but models do assign examples to the *no anaesthesia* class. Because there exist several subreasons why a wound was not treated, it is most likely that no common characteristics could be found between these cases and therefore no cases were assigned to the class.

7.3 Comparing models

Regarding the subquestion in which it was questioned which supervised machine learning algorithm would achieve the best performance, the neural network turned out to perform best on the binary classification problem. For the multi-class classification problem the support vector machine performed best on analysing all text fields and the Naive Bayes model performed best on analysing

the therapy text field.

In case all text fields are included, the neural network only performed significantly better than the logistic regression model and the Naive Bayes model. In case the therapy text field was analysed, the neural network performed significantly better than all other models. However, no significant differences were found between other models. The fact that only these significant differences in performance were found indicates that the choice between algorithms is not strongly influencing performance.

The accuracy on the test set was also influenced by the analysed text. Although feature selection was applied, models that included only the therapy text field outperformed most of the models that analysed all text fields, for both the binary and the multi-class classification. However, for the binary classification it did not reach significance. For the multi-class classification only a significant difference was found for the Naive Bayes and the support vector machine model. The small number of significant differences can be explained by the fact that most information is indeed extracted from the therapy text field if assigning class labels manually, but in a lot of cases information from other text fields is also needed for definite proof.

7.4 Importance of feature selection

We have focused on the differences between several machine learning algorithms as well as the difference between analysing all text fields and only the therapy text field. However, generating and selecting predictive features is even more important than choosing the right learning algorithm and text field. Therefore, models have been developed that take all features as input. For the binary classification problem, all predictive models with feature selection performed significantly better than models without feature selection if all text fields where analysed. However, for the therapy text field, this significant difference could only be found for the Naive Bayes model. This can be explained by the difference in numbers of features between feature selection and no feature selection. The total number of features for the therapy text field is 49 while it is 242 for all text fields. Therefore, the difference between the number of features in case the therapy text field is analysed is likely to be smaller than in case all therapy text fields are analysed. For the multi-class classification, only a significant difference was found between feature selection and no feature selection in case the Naive Bayes model analysed all text fields. No significant differences were found for the therapy field. The fact that the impact of feature selection is lower for the multi-class classification than for the binary classification can be due to the number of features that was used if feature selection was applied.

If feature selection was applied on the multi-class classification, many features were selected and therefore the number of features was almost the same if no feature selection was applied. Therefore, the models and thus the performances did not differ significantly.

The number of features that each model used as input is diverse. It varied between 15 and 56 for the binary classification, and between 115 and 231 for the multi-class classification. To answer the subquestion in which it was questioned which features are most predictive, the best performing models are compared to the simplest models, namely the logistic regression models. The binomial logistic regression model uses 15 features. The coefficients of the regression models describe the relationship between the feature and the class label. Because binary feature frequency was used, we can compare the coefficients because the values of the features are all on the same scale. In the binomial logistic regression model, *gelijmd*, *geplakt*, *hechting*, *klinibond*, and *verwijderen* have a negative coefficient, which indicates that they are predictive for the incorrectly assigned class. All of these negative features are indeed important when manually searching for incorrectly assigned activity codes. If the features are ranked according to their coefficients, *hechtingen*, *gelijmd*, and *ethilon* turned out to be the most important features. This suits with the keywords that are searched for if assigning the class labels was done manually.

The difference between the weight of *hechting* (-0.891) and *hechtingen* (3.812) supports the idea that stemming could be disadvantageous to the current research aims. Whereas *hechting* is a predictive feature for the incorrect class label, the feature *hechtingen* is a predictive feature for the correct class label. Another notable feature that is used as input for binomial logistic regression model is *Timmers*. This is the name of a physician and since the weight of this feature is positive, it is possibly someone who is often asked to suture large wounds.

7.5 Redirecting cases to humans

The performance of the current models could be improved by redirecting difficult cases to humans. A remaining question concerns the optimal balance between human labor and predictive, automated performance. It is up to hospitals to decide, but examining the labor plotted against accuracy shows that improvements in performance could be achieved with a relatively small amount of human work. For the binary classification, assigning 13.5% of the cases manually will lead to an accuracy of 95.3%. With a 13.5% increase of the amount of manual work, the performance can be increased with 3.1%. In case of the multi label classification, the amount of work would be around 12.3% to achieve an accuracy of 88.7%. In other words, increasing the amount of work with 12.3% would lead to a

performance improvement of 4.3%. However, it should be taken into account that these estimates are based on the assumption that humans classify every case correctly. Since in reality this is not plausible, the improvement in performance would be lower in real world scenario's.

7.6 Future work

The highest accuracy was 0.933, which was achieved with a neural network on the binary classification problem analysing only the therapy text field. The goal of the experiment conducted in this thesis was to give a proof of concept for using machine learning and text mining to predict whether activity codes were assigned correctly. Although it was beyond the scope of this thesis to examine other activity codes, it would be interesting to examine EHR forms corresponding to other activity codes as well. Since health insurance companies adapt the rules and regulations every year, it would be of great advantage if the verification part of their process could be automated. In order to be able to extend the current experiment to investigate alternative activity codes, the pre-processing phase is kept quite general. The only pre-processing approach that can be beneficial for other activity codes but not for the current activity code, is stemming. However, including it would require minimal changes.

In case the current results fail to convince health insurance companies to fully automatise manual work, the application might be useful in preventing the problem of incorrectly assigned activity codes. In the current situation approximately 40 percent of the assigned activity codes is incorrect. It might be advantageous to use the application to prevent physicians from making these mistakes in the first place instead of correcting them later. As in the current situation, the physician would fill in the EHR form and would assign the performed treatment to a particular activity code. In contrast, the algorithm could subsequently be used to predict whether this activity code is correct or not. In this, a cut-off level of confidence could be set. In case the level of confidence is above the cut-off level, everything is fine. Otherwise, there will be a pop-up on the screen to warn the physician. Then the physician is able to change the activity code or add missing information to the EHR form, possibly based on the predicted reason for rejection.

References

[1] American Association for the Advancement of Science (2013). Cancer Knife Sniffs Out Tumor Cell, *Science*, **341(6143)**, 221-222.

[2] Aggerwal, C. C., and Zhai, C. (2012). *Mining Text Data*. New York, NY: Springer.

[3] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

[4] Bodenheim, T. (2005). High and Rising Health Care Costs. Part 2: Technologic Innovation. *Annals of Internal Medicine*, **142(11)**, 932 – 937.

[5] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram Models of Natural Language. *Computational Linguistics*, **18(4)**, 467-479.

[6] Caruana, R., and Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. In Cohen, W., and Moore, A., *Proceedings of the 23rd International Conference on Machine Learning* (pp. 161-168). New York, NY: ACM.

[7] Cios, K. J., and Moore, G. W. (2002). Uniqueness of Medical Data Mining. *Artificial Intelligence in Medicine*, **26(1)**, 1-24.

[8] Custers, T., Arah, O.A., and Klazinga, N. S. (2007). Is there a Business Case for Quality in the Netherlands?: A Critical Analysis of the Recent Reforms of the Health Care System. *Health Policy*, **82(2)**, 226-239.

[9] Das-Purkayastha, P., McLeod, K., and Canter, R. (2004). Specialist Medical Abbreviations as a Foreign Language. *Journal of the Royal society of Medicine*, **97(9)**, 456.

[10] Dasg, M., and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis* **1(3)**, 131-156.

[11] Doi, K. (2014). Current Status and Future Potential of Computer-aided Diagnosis in Medical Imaging. *The British Journal of Radiology*, **78**, 4-19.

[12] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive Learning Algorithms

and Representations for Text Categorization. In Pissinou, N., Nicholar, C., French, J., and Gardarin, G. *Proceedings of the 7th International Conference on Information and Knowledge Management* (pp. 148-155). New York , NY: ACM.

[13] Falkenauer, E. (1998). On method overfitting. *Journal of Heuristics*, **4(3)**, 281-287.

[14] Feldman, R., and Sanger, J. (2007). *The Text Mining Handbook*. New York, NY: Cambridge University Press.

[15] Fellbaum, C. (2005). *WordNet and wordnets*. In Brown, K. (eds.), *Encyclopaedia of Language and Linguistics (2nd ed.)* (pp. 665-670). Oxford: Elsevier.

[16] Friedman, J., Hastie, R., Simon, N., and Tibshirani, R. (2016). *Lasso and Elastic-Net Regularized Generalized Linear Models Version 2.0-5*. On 22 December 2016 retrieved from: https://cran.r-project.org/web/packages/glmnet/glmnet.pdf

[17] Van Ginneken, E., Schäfer, W., and Kroneman, M. (2011). Managed Competition in the Netherlands: an Example for others? *European Union Law and Health*, **16(4)**, 23-26.

[18] Greenes, R. A., (2014). *Clinical Decision Support: The Road to Broad Adoption (2nd ed.)*. San Diego, CA: Academic Press.

[19] Hsu, C. W., and Lin, C. J. (2002). A Comparison of Methods for Multiclass Support Vector Machines. *IEEE Transactions of Neural Networks*, **13(2)**, 415-425.

[20] Huang, M. J., Chen, M. Y., and Lee, S. C. (2007). Integrating Data Mining with Case-based Reasoning for Chronic Disease Prognosis and Diagnosis. *Expert Systems with Applications*, **32(3)**, 856-867.

[21] Huiskes, A. (2014). Designing a Device is Like Painting a Picture. *Webmagazine Maastricht University*, On 24 May 2016 retrieved from: http://webmagazine.maastrichtuniversity.nl/index.php/research/technology/item/479-designing-a-device-is-like-painting-a-picture.

[22] James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York, NY: Springer.

[23] Koh, H. C., and Tan, G. (2011). Data Mining Applications in Healthcare. *Journal of Healthcare*

Information Management, **19(2)**, 64-72.

[24] Kusiak, A., Dixon, B., and Shah, S. (2005) Predicting Survival Time for Kidney Dialysis Patients: A Data Mining Approach. *Computers in Biology and Medicine*, **35(4)**, 311-327.

[25] Lewis, D. (1998) Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. *European Conference on Machine Learning*, 4-15.

[26] Lovins, J. B. (1968). *Development of a Stemming Algorithm.* Cambridge: MIT Press.

[27] Matheus, C. J., Piatetsky-Shapiro, G., and McNeill, D. (1996). *Selecting and Reporting What is Interesting: The KEFIR Application to Healthcare Data.* In Fayyad, U, M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (eds.), *Advances In Knowledge Discovery and Data Mining* (pp. 495-515). Cambridge, MA: MIT Press.

[28] Meyer, D. (2015). *Support Vetor Machines: The Interface to libsvm in package e1071.* On 22 December 2016 retrieved from: ftp://cran.r-project.org/pub/R/web/packages/e1071/vignettes/svmdoc.pdf

[29] Milovic, B., and Milovic, M. (2012). Prediction and Decision Making in Health Care using Data Mining. *Kuwait Chapter of the Arabian Journal of Business and Management Review*, **1(2)**, 69-78.

[30] Monard, M. C., and Batista, G. E. (2002). Learning with Skewed Class Distributions. *Advances in Logic, Artificial Intelligence, and Robotics: LAPTEC 2002*, **85**, 173 – 182.

[31] Nederlandse Zorgautoriteit (2013). *Verrichtingenlijst.* On 24 May 2016 retrieved from: https://www.nza.nl/1048076/1048155/BR_CU_2081_Bijlage_3__Zorgactiviteitentabel.xls

[32] Nederlandse Zorgautoriteit (2015). *Prestaties en Tarieven Medisch Specialistische Zorg.* On 24 May 2016 retrieved from: https://www.nza.nl/1048076/1048090/BR_CU_2136__Prestaties_en_-tarieven_medisch_specialistische_zorg.pdf.

[33] Nederlandse Zorgautoriteit (2016). *Handleiding DBC-systematiek.* On 18 May 2016 retrieved from: http://werkenmetdbcs.nza.nl/zz-releases/algemeen-4/nu-geldende-documenten/menu-id-1954.

[34] Nictiz (2016). *SNOMED CT.* On 29 June 2016 retrieved from:

https://www.nictiz.nl/terminologiecentrum/snomed-ct

[35] Popowich, F. (2005). Using Text Mining and Natural Language Processing for Health Care Claims Processing. *ACM SIGKDD Explorations Newsletter*, **7(1)**, 59-66.

[36] Rich, E. and Knight, K. (1991). *Artificial Intelligence (3rd ed.)*. New York, NY: McGraw-Hill.

[37] Ripley, B., and Venables, W. (2016). *Feed-Forward Neural Networks and Multinomial Log-Linear Models*. On 22 December 2016 retrieved from: https://cran.r-project.org/web/packages/nnet/nnet.pdf

[38] Russell, S.J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach (2nd ed.)*. Upper Saddle River, NJ: Prentice Hall.

[39] Soni, J., Ansari, U., Sharma, D., and Soni, S. (2011). Predictive Data Mining for Medical Diagnosis: An overview of Heart Disease Prediction. *International Journal of Computer Applications*, **17(8)**, 43-48.

[40] Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

[41] Tan, A. H. (1999) Text Mining: The State of the Art and the Challenges. *Proceedings of the PAKDD: Workshop on Knowledge Discovery from Advanced Databases*, **8**, 65-70.

[42] Tong, S., and Koller, D. (2001). Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine learning Research*, **2**, 45-66.

[43] Witten, I. H., and Frank, E. (2005). *Data Mining: Practical Machine Learning tools and Techniques (2nd. Ed.)*. San Francisco, CA: Morgan Kaufmann Publishers.

Appendix I - Stopwords

a	daarom	ergens	hier	liet	neemt
aa	daarvan	erin	hierbij	linker	nergens
aan	dan	erop	hierdoor	links	net
al	danwel	ertussen	hierna	m	niet
aldaar	dat	eruit	hiervan	maar	nieuw
alhier	de	etc	hiervoor	mag	nieuwe
all	der	even	hij	man	niks
alle	deze	evenals	hoe	med	nog
allen	dezelfde	eventueel	hoeft	mede	nogmaals
alles	dhr	evt	iemand	mee	nooit
als	die	gaan	iets	meer	nu
altijd	digimapje	gaarne	in	met	of
ander	dit	geen	indien	mevr	oma
andere	diverse	gegaan	indien	mevrouw	omdat
anders	door	gehad	inmiddels	mg	omver
bdz	dus	gekomen	iom	middels	ondanks
beide	e	gekregen	ipv	midden	onder
bekend	echter	geleden	is	mij	onduidelijk
bij	een	geweest	ivm	mocht	onduidelijke
bijna	eerder	goed	jhk	moeder	ong
blanco	eerdere	gr	kan	moest	ongeveer
boven	eerst	graag	kg	moet	ook
buiten	eigenlijk	haar	klein	moeten	op
cc	eigenlijk	had	kmh	mogelijk	opa
circa	elders	heb	komt	mogelijke	opeens
cm	elkaar	hebben	kon	na	os
cwk	elke	heeft	kreeg	naar	over
daar	en	heel	krijgen	naast	overig
daarbij	enige	heen	krijgt	nadat	overige
daardoor	enof	hele	kwam	nadien	p
daarmee	er	helemaal	l	name	paar
daarna	eraf	hem	langs	nav	pas
daarnaast	erbij	het	li	nee	pat

patient	te	uur	vindt	want	zeker
patiente	tegen	vader	volgens	was	zelf
per	ten	van	vond	wat	zich
plots	ter	vanaf	voor	weer	zichzelf
plotseling	terwijl	vanuit	vooraf	wegens	zie
pte	tevens	vanwege	vooral	wel	zij
re	thv	vanzelf	waar	welke	zijn
rechter	tijdens	veel	waarbij	werd	zoals
rechts	toch	ver	waardoor	wil	zojuist
reeds	toe	verder	waarna	wilde	zonder
rond	toen	verdere	waarom	willen	zou
rvk	tot	vervolgens	waarop	word	zover
seh	totaal	vg	waarschijnlijk	worden	zowel
sinds	totale	via	waarvan	wordt	
sindsdien	tussen	vind	waarvoor	wou	
tal	uit	vinden	wanneer	zal	

Appendix II - Synonyms

word	replaced by
agraves	aggraves
blok	block
dgn	dagen
ethylon	ethilon
exp	expectatief
fraxi	fraxiparine
hibiset	hibicet
ha	huisarts
lido	lidocaine
pcm	paracetamol
proleen	prolene
steriestrips	steristrips
tetanusinjectie	tetanus
tetanusvaccin	tetanus
tetanustoxoid	tetanus
staples	nietjes

Appendix III - Binary features

Binomial logistic regression, Naive Bayes and Neural Network

augmentin	gehecht	hechten	klinibond	verdoving
distaal	gelijmd	hechting	lidocaine	verwijderen
ethilon	geplakt	hechtingen	timmers	vicryl

Support Vector Machine

augmentin	dorsale	hechten	lidocaine	roodheid	vingers
been	drukverband	hechting	lijkt	sensibiliteit	volaire
bewegen	ethilon	hechtingen	nagel	snijwonden	volledig
chirurg	extensie	hechtwond	neurovasculair	timmers	vue
dagen	flexie	huisarts	onderarm	tpv	week
desinfectie	forse	hypertensie	ono	verband	zijde
diclofenac	gehecht	intact	ossale	verdoving	
diepe	gelijmd	klinibond	phalanx	verwijderen	
distaal	geplakt	koorts	poli	vicryl	
distale	gevoel	laten	pulm	vinger	

Appendix IV - Multi features

Multinomial logistic regression model, Naive Bayes and Support Vector Machine

aangezicht	bijgeluiden	duim	gevallen	kleine	normaal
aanw	bloed	eigen	gevoel	klinibond	normale
abd	bloedend	erg	gezicht	knie	observatie
abdomen	bloedende	ethilon	gezien	koorts	onbekend
achter	bloeding	extensie	gezond	kracht	onderarm
achterhoofd	braken	fiets	glas	laatste	onderzoek
actief	brein	flexie	goede	lab	ongestoord
advies	buik	flink	grote	laceratie	ono
afwijkingen	capitis	fors	hand	laten	oog
alcohol	cerebrum	forse	hechten	lateralisatie	open
alert	chirurg	fractuur	hechting	letsel	opgelopen
alleen	contact	functie	hechtingen	licht	opname
ambu	controle	gebeurd	hechtwond	lidocaine	oppervlakkig
amnesie	cor	gebloed	helm	ligt	oppervlakkige
anamnese	dagen	gebraakt	hematoom	lijkt	oraal
arm	desinfectie	gebruikt	hoofd	maal	orbita
armen	deur	gedaan	hoofdpijn	max	ossale
arts	diclofenac	gedronken	hoofdwond	mes	paracetamol
augmentin	diepe	gegeven	huid	min	phalanx
auto	digi	gehecht	huis	misselijk	pijn
avd	digimap	gelaat	huisarts	morgen	pijnlijk
barstwond	direct	gelijmd	hypertensie	motoriek	pijnlijke
been	distaal	gemaakt	infectie	nag	pijnstilling
behaarde	distale	geplakt	instructies	nagel	poli
bekken	dorsale	geslagen	intact	nek	pols
benen	dossier	gesloten	isocoor	neuroloog	pulm
bewegen	drukpijn	gesneden	jaar	neurovasculair	pupillen
bewustzijn	drukpijnlijk	gestoten	kin	neus	retour
bewustzijnsverlies	drukverband	gestruikeld	klachten	nodig	roodheid

sat	teen	trap	verdoving	vue	wondjes
scherpe	tekenen	trauma	verwijderen	waar	wondranden
schoongemaakt	terecht	twee	vicryl	week	zichtbaar
sens	tetanus	uitleg	viel	weet	ziek
sensibiliteit	thorax	val	vinger	werk	zijde
snijwond	thuis	vanavond	vingers	wijkend	zit
snijwonden	timmers	vandaag	volaire	wijkende	zwelling
soepel	top	vanmorgen	volledig	wond	
steristrips	toxoid	verband	voorhoofd	wonden	
symmetrisch	tpv	verbonden	vrij	wondje	

Neural Network

abd	diepe	geplakt	koorts	pijnlijke	verwijderen
alcohol	distaal	gestruikeld	laten	pijnstilling	vicryl
alert	distale	gevoel	letsel	poli	vinger
ambu	dorsale	gezond	lidocaine	pols	vingers
arm	drukverband	glas	lijkt	retour	volaire
arts	duim	goede	maal	roodheid	volledig
augmentin	eigen	grote	min	sat	vue
auto	ethilon	hand	motoriek	scherpe	waar
been	extensie	hechten	nag	sens	week
bewegen	fiets	hechting	nagel	sensibiliteit	wijkende
bijgeluiden	flexie	hechtingen	neurovasculair	snijwonden	wonden
bloedende	flink	hechtwond	onderarm	tekenen	wondranden
brein	fors	huid	ongestoord	tetanus	zichtbaar
chirurg	forse	huisarts	ono	timmers	ziek
controle	fractuur	hypertensie	open	toxoid	zijde
cor	functie	infectie	opgelopen	tpv	
dagen	gebruikt	intact	oraal	twee	
desinfectie	gedronken	jaar	ossale	verband	
deur	gehecht	klinibond	paracetamol	verbonden	
diclofenac	gelijmd	knie	phalanx	verdoving	