DEGREE PROJECT IN INFORMATION AND COMMUNICATION TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Multi-view versus single-view machine learning for disease diagnosis in primary healthcare

ALEKSANDAR LABROSKI

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Abstract

The work presented in this report considers and compares two different approaches of machine learning towards solving the problem of disease diagnosis prediction in primary healthcare: single-view and multi-view machine learning. In particular, the problem of disease diagnosis prediction refers to the issue of predicting a (possible) diagnosis for a given patient based on her past medical history. The problem area is extensive, especially considering the fact that there are over 14,400 unique possible diagnoses (grouped into 22 high level categories) that can be considered as prediction targets. The approach taken in this work considers the high-level categories as prediction targets and attempts to use the two different machine learning techniques towards getting close to an optimal solution of the issue. The multi-view machine learning paradigm was chosen as an approach that can improve predictive performance of classifiers in settings where we have multiple heterogeneous data sources (different views of the same data), which is exactly the case here. In order to compare the single-view and multi-view machine learning paradigms (based on the concept of supervised learning), several different experiments are devised which explore the possible solution space under each paradigm. The work closely touches on other machine learning concepts such as ensemble learning, stacked generalization and dimensionality reduction-based learning. As we shall see, the results show that multi-view stacked generalization is a powerful paradigm that can significantly improve the predictive performance in a supervised learning setting. The different models performance was evaluated using F1 scores and we have been able to observe an average increase of performance of 0.04 and a maximum increase of 0.114 F1 score points. The findings also show that approach of multi-view stacked ensemble learning is particularly well suited as a noise reduction technique and works well in cases where the feature data is expected to contain a notable amount of noise. This can be very beneficial and of interest to projects where the features are not manually chosen by domain experts.

Sammanfattning

Arbetet som presenteras i denna rapport beaktar och jämför två olika metoder för maskininlärning för att lösa problemet med prognos för sjukdomsdiagnos i primärvården: single-view och multi-view maskininlärning. I synnerhet avser problemet med sjukdomsdiagnos prediktion av en (möjlig) diagnos för en given patient, baserat på dennes tidigare medicinska historia. Problemområdet är omfattande, i synnerhet med tanke på att det finns över 14 400 unika möjliga diagnoser (grupperade i 22 högkvalitativa kategorier) som kan betraktas som förutsägbara. Tillvägagångssättet i detta arbete betraktar kategorierna i hög-nivå och försöker använda de två olika maskininlärningsteknikerna för att komma nära en optimal lösning på problemet. Multi-view maskininlärningsparadigmet valdes som ett tillvägagångssätt som kan förbättra prediktiv prestanda för klassifikationer i inställningar där vi har flera heterogena datakällor (olika visningar av samma data), vilket är det exakta fallet här. För att jämföra single-view och multi-view maskininlärning paradigmerna (baserat på begreppet övervakat lärande), är flera olika experiment utformade som undersöker det möjliga lösningsutrymmet under varje paradigm. Arbetet berör noga andra koncept för maskininlärning, som ensembleinlärning, samlad generalisering och dimensioneringsreduktionsbaserat lärande. Som vi kan se visar resultaten att multi-view samlad generalisering är ett kraftfullt paradigm som kan förbättra den prediktiva prestandan avsevärt i en övervakad inlärningsinställning. De olika modellernas prestanda utvärderades med hjälp av F1-poäng och vi har kunnat observera en genomsnittlig ökning av prestanda på 0,04 och en maximal ökning av 0.114 F1 poäng. Resultaten visar också att tillvägagångssättet för multi-view stacked ensemblelärande är särskilt väl lämpat som en brusreduceringsteknik och fungerar bra i fall där funktionsdata förväntas innehålla en anmärkningsvärd mängd brus. Detta kan vara mycket fördelaktigt och av intresse för projekt där funktioner inte manuellt väljs av domänexperter.

Acknowledgments

First and foremost I would like to thank my academic supervisor Anne Håkansson for her very constructive feedback throughout the whole thesis process. Additional thanks goes to my examiner Šarūnas Girdzijauskas who pushed me to find the perfect match for my thesis topic and succeeded. An additional expression of gratitude towards Inovia AB and my industrial supervisor David Buffoni without whose machine learning lessons this thesis would not have been possible. Last but not least, I would like to thank my colleague Viet-Anh Phung for reviewing my work and providing me invaluable feedback towards improving both myself as a writer and the work presented here.

Aleksandar Labroski
Stockholm, September 16, 2018

Contents

Chapter 1

Introduction

1.1 Problem area

Is there a way to predict the current health of an individual or a group of people? As you may imagine, there is no straightforward way to provide an answer. This is mostly due to the fact that there are many factors which can affect the health of an individual [1]. On a high level, these include:

1. The social and economic environment;
2. The physical environment;
3. The persons individual characteristics and behaviors.

Each of these high level categories are comprised of multiple sub-categories which in turn include even more fine-grained factors that can be important indicators of health. For example, considering the physical environment determinants, we can include variables such as airborne pollutants, water pollutants, food-borne hazards and even things such as the climate of the region in question (e.g. the outside temperature [2]). All of these can affect the individual's health in a negative or positive way.
In an ideal case, if we had access to reliable information regarding all of these health factors for a particular inhabited environment we would be able, with a high level of certainty, to determine the future health

of an individual or a group of people. However, in a reality we rarely have access to this kind of detailed information, especially considering the fact that it is very difficult to quantify many of these factors. Figure 1.1 visualizes the different determinants of health and their mutual correlation in a diagram.

It is important to mention why we need to strive to get closer to a solution for this issue. As the world population grows, so does the spending on disease prevention and healthcare. More specifically, in the EU, for example, we have seen a rise of 2% of GDP on health-related expenditure between 2000-2015 [3]. This may not seem like much, but if we consider the fact that EU has a budget of €15.3 trillion, then this figure starts to get its real meaning [4]. A non-negligible part of this is spent on primary healthcare which is typically the first step for a patient to be diagnosed. If we can reduce the time that both patients and doctors spend on getting to the right diagnosis, we can potentially reduce a large part of this spending figure and give doctors more time to treat more patients and potentially increase lifespan for the general population.

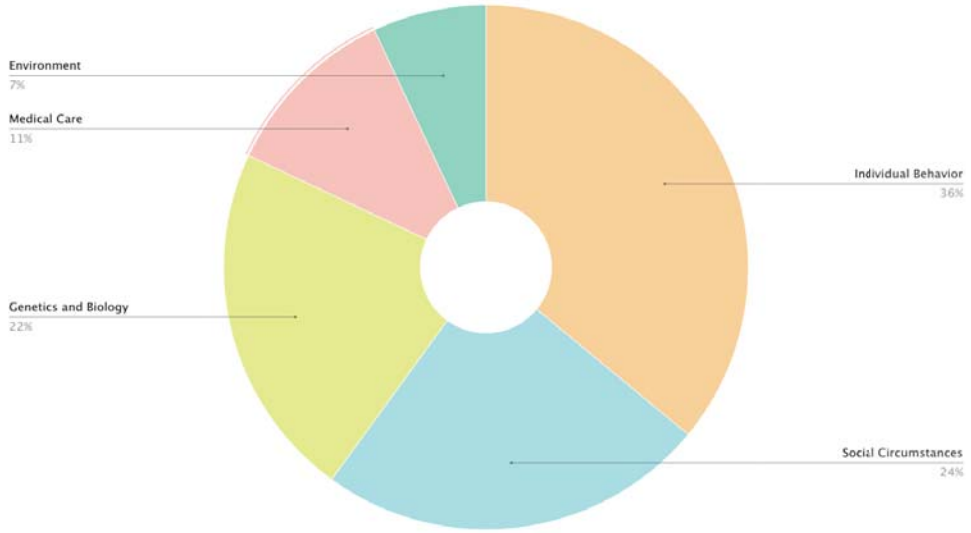

Figure 1.1: Correlated health factors for an individual.
Source: [5]

The work done in this thesis shall explore a potential direction

towards solving this problem. A standard approach that is considered within the field of AI-augmentation of healthcare: predicting the health of individuals based on the information that we have in their medical history. This is normally data that is collected routinely in common medical settings, and is not specifically meant for research purposes.

The problem of predicting diagnoses in healthcare has been typically tackled by researches using machine learning (and more specifically, supervised learning [6]). This popularity has been strongly driven by the acceptance and wide adoption of Electronic Health Records (or simply EHRs) [7]. An EHR is the full medical-related history and data available for a given patient. The EHR contains different aspects of an individual's health status such as diseases (chronic or not), prescriptions, treatment plans, imaging data (such as MRI or X-rays) etc [8].

With the wide adoption and availability of EHRs the supervised machine learning approach in healthcare has been used on many occasions, with high or low success [9, 10, 11, 12, 13, 14]. The reasoning behind this method is sound, taking into account the following: considering the medical history/record of a patient, we can say that it has in itself implicitly included information regarding many of the factors that we discussed earlier. For example, if a patient has been diagnosed with hypothyroidism, then it is more likely that she may have been exposed to excessive amounts of iodine at some particular point in her past. Additionally, other authors have also suggested and identified a link between past diseases and future complications of the same or similar nature and used this fact to drive disease predictions [9]. Even though this is a somewhat simplified way of looking at the issue, one can argue that it is still the optimal way that we can approach a solution based on the information that we have available. In addition to this information, a set of other easily accessible factors can be included to improve the predictions: personal information (age, sex, gender), environmental information (weather conditions at the time of diagnosis, the week/month at the time of diagnosis etc.), however this is rarely done in practice.

Typically, the work done on predicting disease diagnoses is focused on one particular morbidity or a small set of similar diseases which

makes the proposed solutions difficult to adapt in general healthcare settings. Another interesting issue to look into is how to predict disease diagnoses considering a set of multiple, potentially unrelated diseases. This is not at all simple, especially considering the vast spectrum of known diseases.

1.1.1 The classification problem

In current health settings worldwide, the most widely used classification system when it comes to diagnoses is the ICD-10 (the 10th version of the International Classification of Diseases standard) [15]. Incorporated in this classification standard, there are over 14,400 unique possible diagnoses (and over 70,000 in the ICD-10-CM variation which is most widely used). With a categorical system of this size, it is obvious that we cannot, in any simple way, train an algorithm to even get close to getting considerable accuracy on such a large classification problem.

The ICD-10-CM codes, however, are grouped in distinct high-level 22 categories (Table 1.1), and predicting these groups would be a much more viable task then attempting to predict the 70,000 low level diagnoses. The reasoning behind the idea is to provide doctors with some insights to know what to further look into. For example, if we get a prediction that a patient, based on her medical history and other included factors, may have a diagnosis which belongs to the group of endocrine, nutritional and metabolic diseases (group IV), then the doctor may decide to order further blood tests examining the levels of hormones.

Considering the fact that healthcare data is more and more widely available to researchers and engineers worldwide, this may seem like a relatively straightforward issue within a machine learning context. However, it is important to stress out that the same issues that are present in most machine learning tasks are also present here: the class imbalance problem [16], the feature extraction problem [17], the curse of dimensionality [18] etc. The class imbalance problem refers to the issue of having imbalanced (target) classes in a given dataset. What this means is that if we, for example, have 3 target classes, one class might be present with 100 examples, the other one with 1000 and the third

Codes contained in group	Description
A00-B99	Certain infectious and parasitic diseases
C00-D49	Neoplasms
D50-D89	Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
E00-E89	Endocrine, nutritional and metabolic diseases
F01-F99	Mental, Behavioral and Neurodevelopmental disorders
G00-G99	Diseases of the nervous system
H00-H59	Diseases of the eye and adnexa
H60-H95	Diseases of the ear and mastoid process
I00-I99	Diseases of the circulatory system
J00-J99	Diseases of the respiratory system
K00-K95	Diseases of the digestive system
L00-L99	Diseases of the skin and subcutaneous tissue
M00-M99	Diseases of the musculoskeletal system and connective tissue
N00-N99	Diseases of the genitourinary system
O00-O9A	Pregnancy, childbirth and the puerperium
P00-P96	Certain conditions originating in the perinatal period
Q00-Q99	Congenital malformations, deformations and chromosomal abnormalities
R00-R99	Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
S00-T88	Injury, poisoning and certain other consequences of external causes
V00-Y99	External causes of morbidity
Z00-Z99	Factors influencing health status and contact with health services
U00–U99	Codes for special purposes

Table 1.1: ICD-10 high level diagnosis categories.

one with only 10. We can easily conclude that the machine learning algorithm will be able to better predict the more frequent classes than the less frequent ones. This is a common occurrence in machine learning problems and there are several solutions which we shall discuss later. The feature extraction problem refers to the issue of finding an appropriate set of features that shall drive the training and prediction using the machine learning algorithm. The curse of the dimensionality is an issue which appears when we have a large number of dimensions in our extracted feature set, which can increase the search space for the optimal hypothesis exponentially, up to a point where the available data becomes sparse.

Another important issue when it comes to the classification process that we need to consider is how we use the available data towards the final predictions. Namely, in most cases, the diagnosis prediction problem is approached by merging all of the available health data from the EHRs for patients and inputting this merged data to the machine learning algorithm in the training phase [9, 10, 11, 12, 13]. However, if we take into account the diversity of the available data in a typical EHR [8], this approach is not always the most appropriate one. For example, there is no straightforward way to extract and use the data available from doctor's notes within the EHR along with the data from medical analyses (such as blood tests). These different data types typically have different statistical properties which are not captured if the data is simply merged together and used as a whole. This is where multi-view learning comes into play. This relatively new learning paradigm, pioneered with the rise of co-training [19] attempts to capture the different statistical properties of the diverse aspects (views) present in the data available for a given machine learning problem. In the classical multi-view approach, the idea is to learn a hypothesis per given view and jointly optimize all functions towards improving the final generalization performance. If we compare this to the standard approach of concatenating all distinct views into a single view to accommodate the data to the single-view paradigm, we can typically notice a (significant) improvement [20].

When it comes to using machine learning in healthcare, not much progress has been done on the issue of using multiple views in a way

that is exploits the strengths of each one to optimize the hypothesis search process. Instead, typically the data being adopted to suit the single-view learning paradigm, instead of the other way around (accommodating the learning approach to the data). This is especially true in the setting of multi-class classification, such as more general disease prediction (such as the work done in this thesis) opposed to binary classification of predicting whether a given disease is present or not, which is an area where a lot more work has been done.

1.1.2 The feature extraction problem

In a machine learning problem in order to find the optimal set of features to use we would either rely on the knowledge of experts or do automatic feature selection [21].

The biggest problem that we face with the usage of EHRs in a machine learning context is structuring the available data in a way that will make it suitable for the problem of diagnosis representation. In the reviewed literature on using EHR data for medical diagnosis decision support we can see feature representation variants ranging from simple ones, such as binary vector representations [9], more complex ones including bagging of events [22], random sub-sequencing and symbolic aggregate approximations [23] as well as temporal weighting of events using variable importance [24]. All of these have been considered as potential techniques for the feature extraction phase of this work and will be explained at a later stage.

The focus of the work done in this thesis is on the machine learning techniques applied and a comparison of their performances and not on the feature extraction process used. However, since the feature set used is imperative in order to get good performance of the machine learning algorithms applied, a significant amount of the preliminary part of the thesis work shall be focused on this aspect. Including, but not limited to, exploration of feature representation space and optimization of the feature extraction techniques used towards finding an optimal set of features.

1.2 Focus and goal

The focus of this thesis work is to compare the performances of multi-view and single-view machine learning approaches towards predicting disease diagnoses in the general setting of primary healthcare. The scope around the data is important in this case since, for example, primary healthcare has different sorts of cases to deal with compared to intensive or acute healthcare. Intensive care conditions are more sensitive and can be life-threatening. In addition to this, in primary healthcare, treatment of diseases is more dispersed over longer time periods compared to intensive care where treatment is more condensed in a given time period.

The goal of this work, on the other hand, is to see whether the multi-view approach is something worth considering in healthcare multi-class classification settings and if it yields significantly improved results or only marginal performance gains. The conclusion on this aspect can give researchers and machine learning engineers valuable insight as to whether the multi-view is worth exploring when implementing decision support systems in this problem area.

1.3 Ethics and sustainability

Usage of medical data for research purposes is a sensitive topic. Efforts have been continuous for years in order to introduce legislation that will enable patients to feel safe when disclosing personal information for research purposes in medical settings. Additionally, many countries (among which are notably Sweden and the UK) allow free usage of medical data for research only if the data has been completely anonymized [25].

In this particular study, the available data has actually been collected in normal medical settings whose primary purpose is not research projects. The data has been completely de-identified and anonymized as to prevent any unnecessary risks to individuals whose data is being used. It is important to mention that this type of data is allowed to be used for research purposes under the new GDPR [26] regulation that

was introduced in April 2016 by the legislative bodies of the European Union and went into effect in May 2018.

In this case, the data has only been used in research purposes and as such, the particular use case carries no danger to individual patients. If the data and the research findings from the work are to be applied in a real medical settings, then we need to consider the repercussions of the application of the data to predict diagnoses. We will need to be very careful who has access to the findings identified by the machine learning models and how these may affect a patient if the findings are incorrect. It has to be stressed out that the medical professional will always have the final say in the context of identifying the condition (or an absence of one) for the patient of interest. Additionally, in the cases where the machine learning models are not particularly certain regarding the condition of a patient (i.e. we have low predicted probability), we can consider removing the findings completely and giving the medical professional no information that any particular condition has been identified.

We will also need to address the issue of doing biased predictions in cases when the model is possibly overfit to the training data. This is why the models trained for these types of studies should always adhere to standard machine learning approaches - training/validation/testing phases before finally choosing the right model that can be used in a real life context.

When it comes to the sustainability aspects, it is important for all research areas to try to adhere or even improve our standing as a society in regards to one or more of the 17 sustainable development goals introduced by the United Nations [27]. The work done in this thesis has a goal to find a better (more accurate) way of predicting diseases in a primary healthcare setting. Considering this, the goal targeted by the research is number 3: Ensuring healthy lives and promoting well-being for all at all ages. By utilizing the predictive power of machine learning algorithms, morbidities in individuals can be discovered in potentially very early phases, thus reducing the risk of serious complications and even death. A potential complication in this respect is the fact that relying only on algorithms to diagnose patients can sometimes lead to wrong conclusions, so the fact that this is not a final diag-

nosis (instead, it is just a possibility) has to be clearly communicated to medical personnel.

1.4 Research question

The goal of the work done and presented in this thesis is getting closer to finding an optimal combination of machine learning algorithms as well as feature extraction techniques that will take us closer to a business-context usable solution for this problem. In order to achieve this, as we already mentioned, it is important to take a step back and consider non-standard approaches towards dealing with this issue. Standard approaches in this case are the single-view machine learning classification methods that are typically used in these cases opposed to using multi-view machine learning approaches which in many cases capture the problem area in a better way (section 1.1.1).

The research question that shall drive the process is the following: **What performance does the single-view machine learning approach have in comparison with a multi-view approach when it comes to predicting diagnoses in primary healthcare?**

1.5 Research methodology

In the context of the research methodology used in this work, the suitable approach was chosen by consulting the work "Portal of Research Methods and Methodologies for Research Projects and Degree Projects" [28] where common methods and methodologies are presented. The portal structures the research methods and methodologies on several levels, and it is important for all of them to be specifically defined for a given research project in order for the work to be carried out in a structured and correct manner.

The first distinction that we have to define is the main research class, in this case that would be the *quantitative approach* where we use real world experiments as well as datasets to reach a certain conclusion. The question of whether the multi-view or the single-view approach is better in terms of performance can be answered by quantifi-

able metrics.

The philosophical assumption is the second part of the research approach definition that we have to look into. In this case, the *positivism* philosophical assumption is chosen since this fits well with our goal of testing performances of approaches within the information and communication technology.

Besides these two parts, we also have to consider the main research method that will be used in the work. In this case, that would be the *applied research method* where we build on existing knowledge (single-view and multi-view paradigms) and use that to develop practical applications.

The research approach is *deductive*, since a generalizable conclusion is drawn based on observations on large amounts of quantitative data (performance metrics) while the research strategy is *ex posto facto*, since the research is done after the data has already been collected. This fact technically does not allow us to make strong assumptions based on the results due to the post-factum appliance of the data towards answering our research question.

1.6 Delimitations

The main consideration within this research is to see whether the multi-view machine learning approach is an appropriate method towards improving the performance of works dealing with this problem area. The work presented considers the multi-view approach as one that is suitable for usage with heterogeneous data sources, but that is not normally used in the context of disease diagnosis. Within the multi-view machine learning category there are also different approaches that can be considered as candidates towards disease classification, however some of the more popular ones have been attempted within some niche areas of disease diagnosis, while others have not been considered. The research presented here considers one standard multi-view approach: Canonical Correlation Analysis (considered as a multi-view baseline) and one non-standard approach: multi-view stacked ensemble learning. When it comes to single-view the standard way of apply-

ing supervised machine learning is by taking all data and feeding it into a machine learning algorithm. This is precisely how the majority of the works presented within the problem area of diagnosis prediction tackle the issue. There are other approaches that can be considered (for example, single-view stacked ensemble learning [29] or kernel approaches combined with SVMs [30]), however these approaches are not the standard when it comes to disease diagnosis classification and will not be considered in this work.

1.7 Outline

The second chapter: *Relevant theory* presents the theory that the research presented here is based on. The third chapter on Related work presents the current state of the art in diagnosis/disease prediction (both the problem classification and feature representation). The fourth chapter on *Data set and processing* provides a complete picture of the data that has been used in this research as well as the whole process (data cleaning, preprocessing and feature extraction) that was used in order to get the data to a state where it is usable in a machine learning context. The chapter on *Experiments and evaluation*, presents the specific experiments along with the steps involved in each one and provides a brief discussion into how each one differs from the rest. The chapter additionally presents the evaluation methods that were used to compare the different experiments and approaches. Chapter six (*Multi-view versus single-view machine learning*) delves deeper into the results obtained from the described experiments and gives suitable interpretations and comparisons between the data. The last chapter: *Conclusions and future work* provides a more focused discussion on the conclusions obtained from the experiments before presenting some pointers on future research possibilities within the problem area.

Chapter 2

Relevant theory

This section explains the relevant theory behind the machine learning algorithms and high-level approaches used in this work along with any important aspects that are important for the reader to be familiar with to follow the specific experiments done in the work presented.

2.1 Concepts and approaches

The issue of predicting diagnoses can be formulated as a supervised learning problem, which we can formally define as follows [6]:

Given a set of data points of the format:

$$(x_1, y_1), (x_2, y_2)...(x_n, y_n)$$

the objective of the supervised learning algorithm is to find (learn) a function (called a **hypothesis** function) that will approximate the mapping of x to y:

$$y = f(x)$$

An optimal or near optimal hypothesis function is found through a search on the space of possible hypothesis functions. There are two types of supervised learning based on the final target output: classification (the output is one of a final set of values) or regression (the output is a number, a continuous value). Typically, within health-care it is much more common to have supervised learning

with classification, both binary (e.g. predicting the presence or absence of a disease) or multi-class (e.g. predicting subtype of a certain disease).

The two main machine learning supervised approaches that are of interest to us in this project can be categorized under two high-level groups:

1. Single-view learning: the standard approach towards solving machine learning problems. Explained in section 2.1.1.

2. Multi-view learning: we would like to see whether applying a multi-view approach to the problem will result in improved predictive performance of the final disease targets. The two sub-approaches that are used in the experiments are: multi-view dimensionality reduction using CCA (explained in section 2.1.2) and multi-view stacked ensembles (explained in section 2.1.3).

2.1.1 Single-view learning

The single-view learning approach is the standard method used in many classification and regression tasks where a machine learning algorithm is applicable. It is typically used to find a hypothesis function that approximates the mapping of x (dependent variables) to y (target variable(s)). A brief overview of this approach was already presented in the Introduction chapter of this report (Section 1.1). This approach has additionally been widely studied in the area of healthcare diagnosis (Section 3). In the predictive work done under the scope of this thesis, the single-view approach is taken as a baseline method to which the other applied approaches are compared performance-wise.

The single-view approach in this case is a straightforward application of machine learning towards the goal of disease classification. This is done by taking all available dependent variables (features) as one, i.e. concatenating all of the different views into one single view (converting a multi-view into a single-view problem) and using this to find the best possible hypothesis that maps the feature set X to the disease diagnosis targets Y.

2.1.2 Multi-view learning

The concept of multi-view learning has been getting attention in the machine learning community since the first work on co-training was published in 1998 [19]. Multi-view learning takes advantage of the fact that data from multiple, heterogeneous sources (views) is available that describe a given problem. This is the main fact that makes multi-view learning a successful and popular machine learning approach. Co-training [19], as one of the earliest algorithms within the area, has shown significant results in successfully using unlabeled data for classification problems and is one of the most widely used approaches for semi-supervised learning. Multi-view learning can be applied in all stages of the machine learning process depending on the approach chosen for the specific problem and the type of multi-view learning:

- **Late combination of views:** the co-training approach (that we mentioned previously) and all of the derived co-training style algorithms belong to this group. The idea here is that multiple learners are applied (each one per view) which are then forced to be consistent across the different views (visualized in figure 2.1). All of the views are considered separately from one another. It is important to note that this approach relies on several assumptions:

  1. Sufficiency: each view is sufficient for classification on its own.
  2. Compatibility: the target function of both views predict the same labels for co-occurring features with high probability.
  3. Conditional independence: the views are conditionally independent given the label.

- **Intermediate combination of views:** the algorithms that belong to this group are the multiple kernel learning methods. Using this approach, the views are combined just before or during the training process. The idea behind this approach is to calculate separate kernels [30] per view which are then combined with a kernel-based method (visualized in figure 2.2).

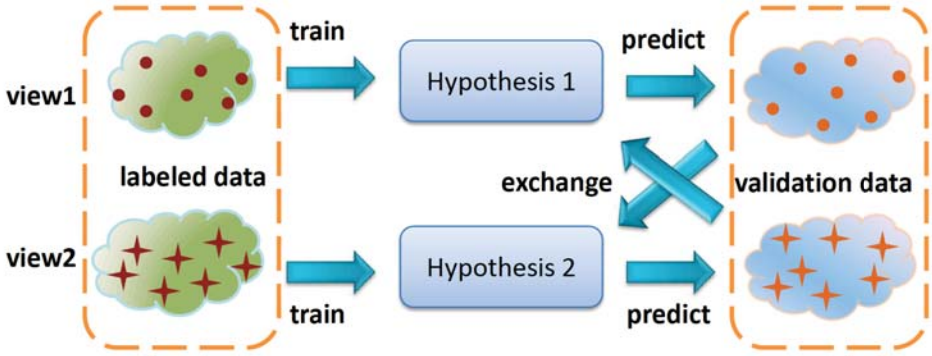

Figure 2.1: A visual representation of the late combination of views.
Source: [31]

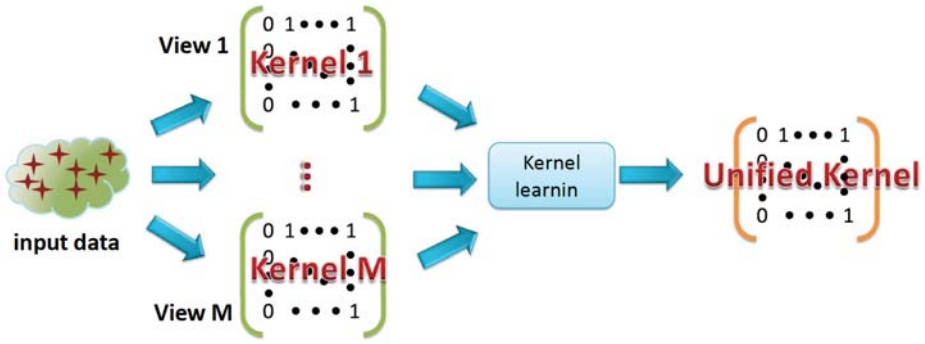

Figure 2.2: A visual representation of the intermediate combination of views.
Source: [31]

- **Prior combination of multiple views:** This group is comprised of subspace learning approaches which aim to obtain a common subspace between the different views by assuming that they have been generated from this latent view (visualized in figure 2.3). The assumption here is that the views were initially generated from some latent subspace.

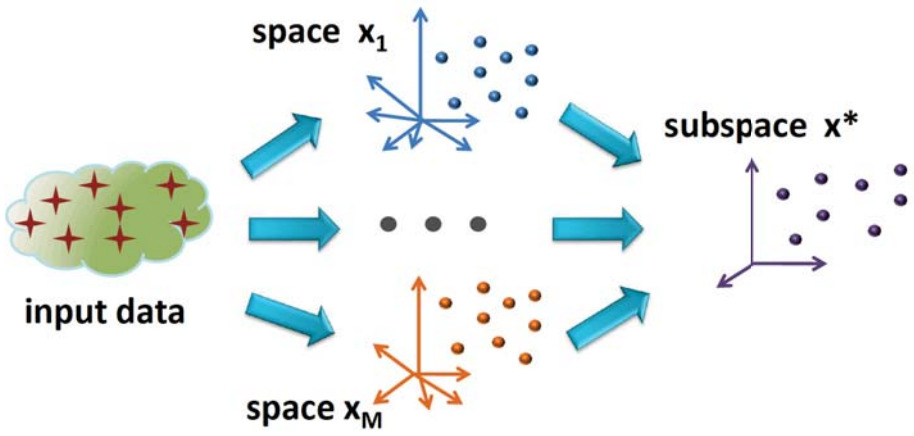

Figure 2.3: A visual representation of the intermediate combination of views.
Source: [31]

In an extensive survey of multi-view learning [31], Chang Xu et al. have identified two significant principles that are common across all multi-view algorithms and ensure their success:

1. **Consensus principle:** This principle aims to maximize the agreement on the multiple, distinct views that are used to train the algorithm. To be more specific, the connection between two hypotheses on two different views and their error rates demonstrate the following inequality [32]:

$$P(f^1 \neq f^2) \geq max\{P_{err}(f^1), P_{err}(f^2)\} \tag{2.1}$$

   In other words, we can conclude that the probability of disagreement of two independent hypotheses upper bounds the error rate of either hypothesis. This means that through minimization of the disagreement of the two hypotheses, we can minimize the error rate of each hypothesis respectively.

2. **Complementary principle:** Each view of the data may contain some information that the others do not have, so using multiple views will more accurately and thoroughly describe the data. The concept of co-training relies on this principle extensively: by labeling unlabeled data using a classifier trained on one view, and then using this (labeled) data to train a classifier on another view these classifiers share the complementary information contained within each view.

Besides these two principles, in another more recent survey on multi-view representation learning [33], the authors have identified another principle which is behind the approach of multi-view representation learning (where the concept of subspace learning approaches belongs): the correlation principle. The essence behind the correlation principle is maximizing the correlations of variables between multiple views. This can be witnessed by looking at the idea behind Canonical Correlation Analysis [34], which is one of the earliest methods for discovery of latent subspaces generalized to multiple views (it can be seen the multi-view version of Principal Component Analysis (PCA) [35]). In particular, the proposition behind CCA is to find linear projections w_x and w_y for two views X and Y, respectively which make the corresponding examples of the two datasets maximally correlated in the projected subspace:

$$\rho = \max_{w_x, w_y} corr(w_x^T X, w_x^T Y) \tag{2.2}$$

If we look at different concepts that comprise the whole machine learning area we can see that multi-view learning is related to other concepts: ensemble learning, active learning and domain adaptation [31]. Active learning [36] is a concept whose idea is minimizing the amount of labeled data required for training. Ensemble learning [37] is using multiple learners and then combining their predictions. Domain adaptation [38] is adapting a model trained on one source domain to another domain, with a different data distribution.

When it comes to healthcare data, we can consider all of the multi-view approaches as potential candidates for our classification problem. However, considering the data diversity and the extent of the

data available (especially within the unstructured domain), more interesting approaches to apply would be co-training style algorithms (to attempt to maximize the agreement between different views) as well as multi-view representation learning algorithms, or subspace learning algorithms to reduce the dimensionality of the data and solve the problem of the curse of the dimensionality.

In the known literature, Canonical Correlation Analysis as a dimensionality reduction technique for classification [39, 40] or regression [41] purposes has not been widely applied even considering the fact that the approach is very suitable for multi-modal, multi-view data. In this particular project, the multi-view approach will be applied on the extracted features as a way to reduce the dimensionality of the available data using the canonical correlation analysis approach and this low-dimensional feature space will be used for the final classification task. The data will be logically split into unstructured vs structured data as the two distinct view groupings required for the method to be successful.

2.1.3 Stacked generalization

An interesting concept which is part of ensemble learning as a broad approach is stacked generalization. First, several algorithms are trained on the available data and then a meta-learner is trained using the outputs of the trained models as input data to the final meta-learner. It has been shown that stacked generalization, yields significantly better performance than any single trained model. The original paper on the idea [29] argues that for a given set of data X, Y we there are many possible learners (generalizers) that can be modeled. The issue that appears in that case is how to address the multiplicity of all these possible learners. The algorithmic approaches such as cross-validation and bootstrapping are winner-takes-all strategies. What this means is that multiple learners are modeled on the available data and the one which yields the best generalization accuracy is considered the winner. In comparison to this, stacked generalization attempts to combine the set of modeled learners instead of choosing one amongst all of them. This is done by taking the outputs of the predictions of the set of mod-

eled learners as points in a new input space and then modeling a new generalizer on the newly extrapolated space.

Even though the concept of stacked generalization is to combine learners on a single set of data, the concept can also be applied to the case of multi-view learning. This can be achieved by modeling learners on the separate data view subsets and then combining their outputs using a separate learning process. The main difference between standard stacked generalization and multi-view stacked generalization is that the standard approach splits the original data into subsets based on the row (index) axis while the multi-view stacking approach splits the original data into subsets based on the column (data dimensionality) axis. As one can expect, while the data splits based on the row axis exhibit similar statistical properties based on the attributes contained, the data splits based on the columns axis generally exhibit different statistical properties since the splits have different attributes instead of distinct samples. This generalization approach is named multi-view stacked ensemble learning and has recently gotten some attention in the machine learning community [42, 43, 44]. Even though this approach is not considered a part of the standard multi-view approaches, it is very closely related to co-training [19] in the sense that it captures the statistical properties behind multiple views in a set of data. Additionally, it seems that the principles that are behind co-training are highly important in the case of multi-view stacking ensembles as well. This was be one of the approaches applied on the classification problem in this thesis and it will be compared to the other approaches that are considered (single-view and multi-view dimensionality reduction).

2.2 Algorithms

The specific learners (machine learning algorithms) that were used in this project are ones which inherently support multi-class classification. The reason for this is mostly due to limited technical capabilities, since algorithms which do not inherently support multi-class classification can only be applied to such problems by applying strategies

such as one-versus-all or all-versus-all [45], both of which require significantly more computing power in the case of large classification problems (many classes) such as the one we have here. In particular, Random Forest [46], the simple Naive Bayes [47] (specifically the Multinomial extension which is typically used for text classification [48]) classifier were the main learners applied to the specific problem in this study. Additionally, the experiments were attempted using a pairwise ranking technique [49] under a gradient boosting framework [50], but since this is a special case and differs significantly from the normal learning algorithms applied here, the description of this approach is presented in the appendix C.

2.2.1 Random forest

The idea of behind the random forest classifier is presented next. A large set of K decision (CART [51]) trees are built out of the training data by sampling the dataset using a technique called bagging [52]. Bagging as a technique is used to draw training sample sets from the original complete training set Θ (with replacement). Each tree is grown on the data subset Θ_k using random feature selection. The trees are not pruned. Each data subset $\Theta_k \epsilon \Theta$ has the same distribution. After a certain number of trees is grown (the number of trees is highly dependent on the nature of the problem and the data), they vote for a class in a given set of classes. The class which has received the most votes wins and it is considered the resulting prediction.

In the original paper [46], two reasons are given towards the usage of bagging:

- Bagging in combination with random feature selection seems to increase accuracy in random forests.
- Bagging provides a way to measure the generalization capability of a classifier. This is done by taking samples from the training set and aggregating the votes on the final prediction of each of the taken samples only on the trees in the forest which did not contain that specific sample in their bootstrap subset. This is also called out-of-bag error estimate.

2.2.2 Multinomial Naive Bayes

The Naive Bayes classifier is based on the Bayes' theorem. It is a simple classifier that has been frequently applied and works well in the case of text classification. The concept is the following [53]: considering an example data point X and its corresponding probability $P(X)$ and a class Y then $P(Y|X)$ denotes the conditional probability that X belongs to class Y. The classifier is a direct application of the Bayes theorem:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \tag{2.3}$$

The assumption made here is a rather "naive" one: the attributes are independent given the class they belong to. Considering the fact that this assumption rarely holds, the Naive Bayes is not an optimal learner for most machine learning problems. However, considering the fact that the concept behind is very simple, it does not require a lot of memory and has low running time (which are desirable properties for a machine learning algorithm) it has been successfully applied on many occasions.

The Naive Bayes classifier comes in several flavors. One such example is the Multinomial version of the Naive Bayes algorithm, which makes the assumption that the feature data distribution is of multinomial distribution [54]. This version of the Naive Bayes learner works rather well in problems with data that has been transformed into event counts, such as word counts in text. This is the version of the classifier that is being applied on the classification problems in this work.

Chapter 3

Related work

The focus of this section is on presenting the current state of the art in available literature when it comes to diagnosis predictions, multi-view machine learning and the fusion of these two disciplines. In current research settings, to the best of the author's knowledge, there has not been much research done on applying the benefits of multi-view learning when it comes to diagnosis classifications. The state of the art is presented in a chronological way and aims to provide an incremental general non-exhaustive overview of the current knowledge within the problem area.

When it comes to the problem of diagnosis prediction research in the currently available literature we can find works dealing with different sort of problems related to the classification process. However, it is important to mention that most of the studies which deal with the problem of diagnosis prediction generally attempt to tackle the issue of classifying a very specific niche of health issues (i.e. whether a certain condition is present or not: binarization of the problem), such as adverse drug reactions, mild stroke, palliative care, haematological diagnosis, heart failure, with very few approaches trying to generalize the learning process to multiple distinct and unrelated diagnoses.

Besides the issue of classification, one additional important aspect is the issue of representing the available medical data (EHRs) in a way that is suitable for machine learning algorithms. To that goal, several authors have presented different approaches in order to maximize the potential of the data representation (capturing statistical properties

of different types of available data, handling the temporality of the data etc.) towards the final goal of diagnosis prediction/classification. These approaches will also be presented here.

It is important to note that this section of the report is not meant to serve as an exhaustive literature survey of the work done within the area, but rather a very brief introduction of the issue from the point of view of existing research done in the past. The section has a goal to provide the basics of the standard approaches towards tackling this problem. This basic knowledge is required for the reader to be able to follow the process presented in this report. Curious readers are encouraged to take a look at more comprehensive surveys and overviews done on the problem area, such as: ***An analytical method for diseases prediction using machine learning techniques*** [10], ***Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review*** [55], or when it comes to the use of deep learning in the context of EHRs, the following survey: ***Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis*** [56].

The reviewed prediction studies are presented in three groups: single-disease, multiple-disease and general disease predictions.

1. **Single-disease prediction:** Predicting the presence or absence of a condition or a specific disease. [14, 57, 58, 59, 13, 60, 61, 12]
2. **Multiple disease (set) prediction:** Predicting the diagnosis of a given set of related diseases. [62, 22, 24, 23, 11]
3. **General disease prediction:** Predicting the diagnosis of multiple, unrelated diseases for a given patient. [9, 12]

3.1 Single-disease prediction

2014 - Supervised heterogeneous multiview learning for joint association study and disease diagnosis [14] - S. Zhe et al. in their study have recognized the fact that most disease diagnosis studies ignore potentially important relationships between heterogeneous views of the

data used in the classification process. To mitigate this, in particular for the case of Alzheimer disease, they propose a new Bayesian method that unifies multi-view learning with sparse ordinal regression (regression is used in order to capture the disease severity with the prediction process). In their approach, the assumption is that the data is generated from common latent features based on separate sparse projection matrices and suitable link functions (similarly to CCA [34]). These common latent features are then used to predict the disease status based on Gaussian process ordinal regression. They got better prediction accuracy by 10% compared to the second best method (plain GP ordinal regression) and 22% better accuracy compared to the worst method (CCA + lasso).

2014 - A framework for feature extraction from hospital medical data with applications in risk prediction [57] - in this study, T. Tran et al. present a framework for automatic feature extraction from complex medical records (i.e. EHRs). The framework applies a set of filters onto event sequences whose goal is to extract a feature set capturing the temporal and data diversity specifics in the original dataset. They have applied the extracted feature set to a task predicting readmission risk and compared this to a baseline feature set extracted using Elixhauser co-morbidities [63] [64]. They used machine learning with logistic regression to predict the readmission risk of 4 different diseases: diabetes, chronic obstructive pulmonary disease (COPD), mental disorders and pneumonia. The extracted feature sets outperformed the baseline by 8% AUC on average.

2015 - Early detection of Heart Failure with varying prediction windows by structured and unstructured data in electornic health records [58] - in this paper, Y. Wang et al. used a combination of feature extraction methods and varying prediction windows (60 to 365 days before the diagnosis) for early prediction of heart failure. They extracted features from two sources:

1. Structured data extraction methods: using counts of events for categorical variables and simple aggregation functions (such as mean) for numeric variables.

2. Unstructured data extraction methods: NLP techniques were used

to extract Framingham HF signs [65] and symptoms from patient notes.

2016 - Deep Patient- An unsupervised representation to predict the future of patients from the electronic health records [59] - R. Miotto et al. in this work present a technique for unsupervised feature representation extraction based on a deep learning architecture. The idea is to automatize the feature extraction process from EHRs by using a stack of denoising autoencoders [66]. The idea of autoencoders is to extract a distributed representation that captures the coordinates along the main factors of variation within the data. The feature sets learned through the usage of the technique were compared to feature sets learned through other feature learning techniques such as principal component analysis (PCA), K-Means clustering, Gaussian mixture model (GMM) and independent component analysis (ICA). The authors have applied the extracted feature sets onto several different disease diagnosis machine learning problems: Diabetes diagnosis, rectal cancer diagnosis, liver cancer, congestive heart failure etc. with notable performance improvements compared to the other feature learning techniques.

2016 - Development of a multivariate clinical prediction model for the diagnosis of mild stroke/TIA in physician first-contact patient settings - Bibok et al. [13] in their study in on development of a multivariate model to predict occurrence of mild stroke [67] used logistic regression to attempt to predict the absence or presence of mild strokes in first-contact patient settings. Their idea is to improve the differentiation between diagnoses with symptoms that mimic mild stroke and actual occurrences of the morbidity.

2017 - Disease prediction by machine learning over big data from healthcare communities [60] - The study within this paper by M. Chen et al. presents a way to predict disease risk for chronic diseases - more specifically, the authors use the cerebral infarction morbidity as a showcase. The authors present 3 different approaches towards predicting the disease risk based on the type of features used: structured data (demographics, living habits, examination items and results and past diseases), unstructured text data (patient's readme illness, doc-

tor's records) and the fusion of these two data types: structured + unstructured. They have applied different machine learning algorithms for each approach:

- Structured data: Naive Bayes (NB), K-Nearest Neighbors (KNN) and Decision Tree (DT)
- Unstructured data: Convolutional Neural Network-based unimodal disease risk prediction (CNN-UDRP) - original algorithm presented in the paper
- Structured + unstructured data: CNN-based multimodal disease risk prediction (CNN-MDRP) - original algorithm presented in the paper

They have obtained notable results with the CNN-based algorithms compared to the standard approaches: 94.2% accuracy for the CNN-UDRP and 94.8% for the CNN-MDRP algorithm while the standard approaches (NB, KNN and DT) have an accuracy of around 50%.

2017 - Improving palliative care with deep learning [61] - The work presented in this study by A. Avati et al. has the aim to improve palliative care by trying to solve a "proxy problem": given a patient and a date predict the mortality of that patient within the next 12 months. The system would then use this prediction to make recommendations for palliative care referral (palliative care as a term refers to care for terminally ill patients and their families according to the National Cancer Institute [68]). The authors are approaching the problem as a binary classification goal and have built a supervised deep learning model to solve it. The model is a deep neural network comprising 18 hidden layers of 512 dimensions with the Scaled Exponential Linear Unit (SeLU) activation function. Due to class imbalance, the evaluation is done using the Average Precision (AP) score, also known as the AURPC (Area Under Precision-Recall Curve). The model achieves AP score of 0.69 with a recall of 0.34 at 0.9 precision. The AUROC is also evaluated with a performance of 0.93.

3.2 Multiple disease (set) prediction

2015 - Temporal weighting of clinical events in electronic health records for Pharmacovigilance [62] - the author of this study (J. Zhao) presents 9 different approaches for weighting of temporal events based on given curve functions. The assumption is that events which are closer to the target label/diagnosis are more important than those which have occurred further in the past. In order to analyze the effect of each weighting strategy on the final predictive performance when it comes to disease diagnosis (in particular for the case of adverse drug reactions (ADEs)), the approaches were assessed using a random forest model for each strategy. The results obtained show that all strategies improve the performance compared to the baseline approach which did not use any event weighting strategies.

2015 - Handling temporality of clinical events for drug safety surveillance [22] - the authors (J. Zhao et al.) of this work present three distinct strategies to handle the temporality aspect within the medical data when performing feature extraction:

1. Bag of events - the number of occurrences of a given clinical event is counted within D days prior to the diagnosis event that is being predicted. This number (x) is taken as the feature:

$$x = \sum_{d=1}^{D} n_d$$

2. Bag of binned events - the number of occurrences of a given clinical event is counted each day within a D days period prior to the diagnosis event being predicted. The features extracted in this case are:

$$n_1, n_2, ...n_D$$

3. Bag of weighted events - different weights are assigned to the clinical event x that occurred at different days. The weights are calculated according to the time distance between the clinical event and the diagnosis event being predicted (events that are

further away in time have lower weight than more recent ones). Accordingly, the value of the feature x is:

$$x = \sum_{d=1}^{D} (n_d/d)$$

This is an advanced way of handling data temporality within electronic health records and even though the authors present this as a way to represent data for adverse drug reaction predictions, this concept can easily be applied in a more general setting of primary healthcare diagnosis predictions. Additionally, even though these strategies are meant to improve the performance of diagnosis classification, the authors stress out that they have not observed significant improvements in predictive performance between the different strategies. Instead, the biggest change in the performance stemmed from the different number of days (D) that were considered as feature candidates prior to the adverse drug reaction diagnosis event being predicted.

2016 - Learning temporal weights of clinical events using variable importance [24] - J. Zhao et al. present a way to use variable importance values learned from a random forest algorithm to extract temporal weights. Each weight learned is used by applying one of two techniques on the features:

1. Weighted aggregation - the corresponding weights are applied to the value of each event from different time windows and then the weighted values of the same event are aggregated.

2. Weighted sampling - weights are used as probabilities with which features are sampled to be considered as possible splitting nodes in the random forest algorithm.

The conclusion of the authors is that the weighted sampling technique provides an improvement when compared with the model trained without weighted sampling.

2016 - Learning from heterogeneous temporal data in electronic health records [23] - the authors (J. Zhao et al.) in this paper present an advanced way of handling temporal data in EHRs. The idea is to

take a time series dataset and apply the following approaches: Piecewise Aggregate Approximation (PAA), Symbolic Aggregate Approximation (SAX) and Sequence shapelets generation to prepare the time series format data for input into a machine learning algorithm. Based on the approaches mentioned, the authors present 3 different ways of transforming time series data into features:

1. Sequence generation: using pure SAX transformations on the time series.

2. Sequence clustering: applying clustering on the sequences generated by using pure SAX transformations and using this as input to a classifier.

3. Random subsequence selection: reducing the diversity of the feature sequences within a given class by finding one sequence that is representative of the common symbol alignment of that class.

4. Random dynamic subsequence: Same as the method above but here the subsequence length is determined automatically.

To evaluate the different approaches the authors used the random forest algorithm on the disease diagnosis problem of adverse drug reactions (ADEs). Their results show that random dynamic subsequence (RDS) method notably improves the performance over a technique presented in one of their earlier works - sequence length (SL), taken as baseline performance [69].

2018 - An application of machine learning to haematological diagnosis - Another related, very recent study is the one presented by Gunčar et al. [11] where data from laboratory blood test results is used in an attempt to predict haematological diseases. The authors present two models which use different sets of features (one as a complete set of all available features, and another one as a reduced set) and have promising results.

3.3 General disease prediction

2017 - Diagnosis prediction from electronic health records using the binary diagnosis history vector representation [9] - The idea of this paper is perhaps closest to what the goal of the research presented in this thesis is. Namely, the authors (I. Vasiljeva and O. Arandjelovic) present a framework for "high-level" disease diagnosis modeling. The concept of "high-level modeling" is to predict/discover co-morbidities based on an extension of the Markov process assumption for disease diagnosis [70] (regarding disease progression as a discrete sequence of events). They argue that this approach is unsuitable in this case due to the fact that older than n-1 (diagnoses that are older than the most recent one that the patient has had) information is lost in the predictive process (obviously a problem with chronic diseases). To mitigate this issue, they propose a different representation of the patient's state, more suitable for disease progression modeling than a Markov assumption representation. Their approach represents the current patient state as a vector of binary values, where each vector element corresponds to a specific diagnosis code and its value is 1 if the corresponding disease is present in the patient's medical history (or 0 otherwise). In this way, the modeling problem is reduced to learning the transition probabilities between the different vectors in the patient's history:

$$p(v(H) \rightarrow v(H'))$$

The authors have applied this approach on a set of 30 diagnoses which were most prevalent (explaining 75% of the data) in an EHR dataset from a hospital in Scotland. They have compared their approach to the previously considered pure Markov process and have found significant improvements in the predictive accuracy: 82% accuracy at rank-1 predictions of the next disease, and 90% at rank-2 vs. 35% accuracy at rank-1 and 50% accuracy at rank-2 for the Markov process baseline.

2018 - Scalable and accurate deep learning for electronic health records - Most recently, Rajkomar et al. [12] in their study on applying deep learning in a clinical setting with EHR records yielded promising results. They are trying to predict the following outcomes for a pa-

tient in an ICU (intensive care unit) setting: inpatient mortality, risk of readmission after 30 days, expected length of stay and the discharge diagnosis. This is probably the only study found in machine learning literature that tries to predict such a large scale of disease (ICD-9) diagnoses: 14,025. They have achieved very promising results, with an F1-micro score of 0.4. However, their study differs from the work presented here in the aspect of what we are trying to predict - namely, the work presented by Rajkomar et al. attempts to classify ICD-9 diagnoses only for hospitalized patients, which is not considered a part of primary healthcare.

One last interesting related work to be reviewed in this report is the one done by Baxter et al. [71] - here the authors present 3 different feature extraction methods for clustering individuals based on data that does not naturally occur in vector form. They discuss that this kind of data frequently appears in medical cases in an event sequence form with irregular events taking place. The authors discuss three options in order to tackle this problem:

1. Converting the sequence data into feature vectors
2. Using a distance based clustering method which allows non-vector data
3. Using a mixture of generative probabilistic models

The paper focuses on the first option and attempts to minimize the loss of information (relevant to their clustering data mining task) when creating feature vectors from. They present three distinct feature vector representations:

1. Count approach of having one feature for each type of service representing the number of times the service was used.
2. Average, residual, deviance feature vector for capturing the required temporal information missed by the count feature vector approach.
3. Gap feature vector - describe the total length of time when the regular required tests are not carried out.

Even though many of the studies above have achieved significant results, they have still not attempted to make use of the different statistical properties that are captured by the different views of the data. Only the deep learning modeling methods some way try to infer derived features exploiting multiple views within the hidden layers of the network, which we cannot reason about in an straightforward way. Additional limitation of most of the studies above is that they cannot easily be applied in a real-world setting due to the fact that most models are built to target single diagnosis prediction and they would not provide insight as a decision support system.

Chapter 4

Data set and processing

4.1 Context

The data that has been used in this work comes from one medical center (vårdcentral in Swedish) from the southern part of Sweden (Skåne). The data has been completely de-identified to protect the privacy of the patients.

The data contains information such as analysis results, past diagnoses, prescriptions, vaccines given, sick leave data etc. from the primary healthcare medical records of patients. The fact that this is primary healthcare data is important in this case due to the fact that this type of data typically contains more general patient health information compared to other type of medical records (such as emergency or acute case records where the data is more specific and related to a condition that was/is being treated) [72].

4.2 Data understanding

In order to understand the data that is available for this project, it is necessary to explore the data from several different aspects - both from a statistical and practical point of view.

The data is semantically split into 6 distinct data subsets. These can be considered as different "views" of the patients' data: **patient notes**, **analysis results**, **diagnoses**, **standing prescriptions**, **sick leave**

and **products**. Each of these data views represents a time series standalone dataset with its own statistical properties and semantics which we discuss more in detail here.

Even though all of the data is used in the predictive part of this work, not all time series data subsets carry equal information or contain the same number of records or patients. Additionally, there is no easy way to connect the events of the different views of the data (unless we consider the dates and the patient names as some sort of a connection) and this makes it hard to reason which event in which view is related to which events in the other views. There are, however some basic statistics that show us what kind of data we are dealing with, and those are presented in this section.

Data view	**Records**	**Patients**
Patient notes	598,863	15,445
Analysis results	243,741	9,273
Diagnoses	144,389	13,320
Standing prescriptions	51,689	9,027
Sick leave	1,014	307
Products	6,403	2,775

Table 4.1: The different data views

The whole dataset with all the different views has 15,957 unique patients. However, not all of these patients have a complete medical history (such as diagnoses) so they cannot be considered for the predictive part of this work. Most of these patients who are missing part of their medical history are either patients who have been entered into the system due to a phone call they have made to the medical center (vårdcentral) for advice or patients who have, for example, received mandatory vaccinations but have not had any other visits to the hospital. Additionally, there is some part of patients who have only had filled prescriptions within this medical center.

After taking out all patients that are not suitable for any predictive tasks (have no assigned diagnoses) we are left with a set of 13,320 patients who have at least one medical diagnosis in their EHR. Most of these patients (75%) have at least 12 diagnoses recorded within their

EHR, while 25% of patients have 2 medical diagnoses. In general, the count of diagnoses that we have for each of these 13,320 patients ranges from 1 up to 241 diagnoses in their unique medical history (Table 4.5).

Besides the view-specific data (such as diagnoses or analysis results), we also have some basic information for each patient such as the birth year, the gender and a de-identified name designation (1-4 digits). The data contains records between 2010-04-11 and 2017-07-31, however most records are from the years 2016, 2015 and 2017 with very few records dating back to 2010.

As we already discussed, the diagnosis predictive targets (labels) can be considered on two different levels: the diagnosis group (22 groups) or the diagnosis code (more than 14,000 distinct diagnoses). This particular dataset contains a set of 1,490 unique diagnosis codes (out of those 14,000), however 80 individual diagnoses explain 70% of the diagnosis records (Figure 4.1). This fact has been used in a similar study where the authors trained a model to predict the top 30 diagnoses, which in their case explained 75% of the EHR data available [9].

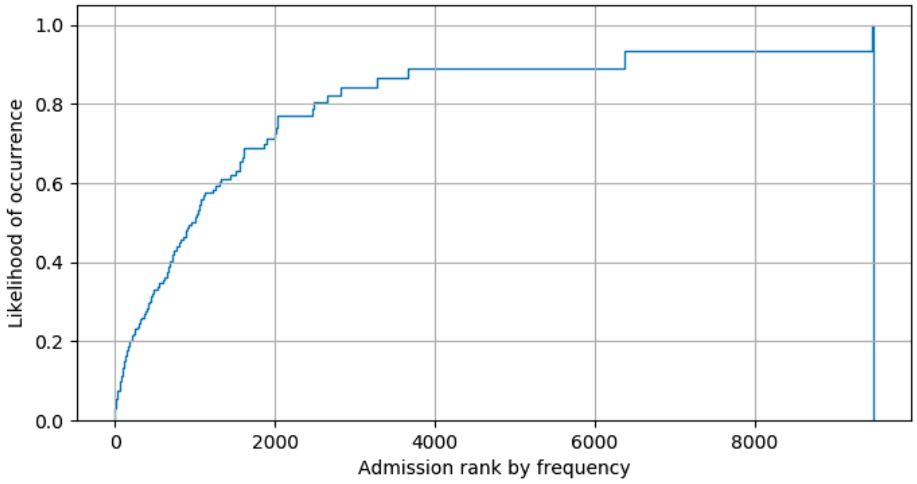

Figure 4.1: A cumulative frequency histogram

One interesting fact is that female patients are more prevalent when

it comes to hospital visits, and thus diagnosis frequency. In particular, out of all individual diagnosis records, female patients hold 92,769 diagnoses while male patients hold 51,610 diagnosis records - a significant portion of 55% more records belong to female patients (Appendix A). This fact is important, since the fact that the machine learning algorithm sees more data of one type can make it more precise on this sort of data. However, this statement can be questioned in this case, since many diseases can be generalized to both female and male patients, excluding diseases of pregnancy and the reproductive system.

When it comes to the actual labels that shall be used as prediction targets, this particular dataset contains records with 21 out of 22 ICD-10 diagnosis groups. The only diagnosis group which is not found in the dataset is the special codes group, U00-U99. It is important to note that even though all of the other groups are present in the dataset, not all of them are found in an equal proportion (we have a class imbalance problem). The labels in the dataset are originally not represented by their diagnosis codes, but instead they are represented by the name of the group in Swedish.

Table 4.2 shows the distribution of the diagnosis groups along with their ratio of the total dataset.

Several possibilities were considered to resolve the class imbalance problem. One possible solution is to remove classes which are below a certain threshold in the dataset (e.g. below 0.01 percent), however this would cause the data to not represent a real-case scenario. Other possibilities of dealing with the class imbalance issue is the usage of under-sampling or over-sampling [73]. Under-sampling is a technique which eliminates examples of the majority classes, while over-sampling replicates examples of the minority classes. Both techniques are an attempt to cause a minimization of the class distribution imbalances. However, both techniques have known drawbacks. The drawback of under-sampling is that we would typically lose some potentially important information and thus artificially lower the performance of the classifier. Over-sampling, on the other hand may lead to severely overfit models, especially if the minority classes are much less frequent than the majority classes. It is important to mention that models which take into account the information contained in exam-

Codes contained	Group name in Swedish	Frequency	Ratio of total
A00-B99	INFEKT SJD	2,911	0.020161
C00-D49	TUMÖR SJD	1,965	0.013609
D50-D89	BLOD & IMM	1,141	0.007902
E00-E89	ENDOKRIN	10,088	0.069867
F01-F99	PSYK SJD	13,824	0.095742
G00-G99	NEUR SJD	2,949	0.020424
H00-H59	ÖGON SJD	1,354	0.009378
H60-H95	ÖRON SJD	3,912	0.027094
I00-I99	CIRK SJD	14,496	0.100396
J00-J99	RESP SJD	9,472	0.065601
K00-K95	MAG-TARM	4,600	0.031859
L00-L99	HUD SJD	6,244	0.043245
M00-M99	MUSK-SKEL	28,740	0.199047
N00-N99	URIN & KÖN	4,698	0.032537
O00-O9A	GRAV SJD	326	0.002258
P00-P96	PERIN SJD	3	0.000021
Q00-Q99	MEDF MISSB	175	0.001212
R00-R99	SYMT & OKL	24,883	0.172334
S00-T88	SKAD & FÖRG	5,780	0.040031
V00-Y99	Y ORS T SJD	3	0.000021
Z00-Z99	KONT M HSV	6,824	0.047262
U00–U99	N/A	0	0.000000

Table 4.2: ICD-10 diagnosis categories distribution

ples when using under-sampling [74]. However, these techniques may artificially impact the performance of a model and provide us with incorrect scores. Even though looking into the right solution for the class imbalance problem in this case is an interesting topic, it does not fall under the scope of the work done here.

One should be aware and note that class imbalance issues such as this one typically lower the performance of a machine learning algorithm. The most acceptable way of dealing with this issue is finding more data. This was attempted in this project, however due to restrictions and security concerns data cannot/could not be made available in any short period of time without extensive preprocessing such as de-identification and anonymization. Due to this, the problem of the class imbalance was not solved. However, it is important to mention that suitable performance metrics will be used, that correctly account for the target class imbalance in the dataset (such as F1 score).

4.3 Sources

This particular dataset (as we already mentioned), has the following 6 distinct sub-datasets:

1. **Patient notes:** this sub-dataset contained the notes that have been written down per patient from the medical personnel (such as reports of telephone calls, nurse meeting notes, doctor interpretations etc). This data view contains 7 specific columns, whose detailed description can be found in the Appendix A. The main columns of interest to us are the 'note', 'local_keyword' and 'patient_notes_interpretation_as_string' where the main text, some keywords and an appropriate keyword interpretation of the findings is stored. The merged text of all three columns together is short, averaging 20 words over all of the examples available in this data view. It is important to note down that we can observe a large number of rows with at least one missing value in this data view. Out of 598,863 records, there are only 1,075 records which do not contain any missing values. Most missing values are in the interpretation columns, which means that most notes

do not have any specific numeric interpretation (they are not referring to any analysis results). Statistics regarding this view are shown in Table 4.3.

Statistical measure	**Value**
Patients	
Unique count	15,445
Mean	38.773907
Standard deviation	51.919300
Minimum	1.000000
25%	8.000000
50%	20.000000
75%	48.000000
Maximum	922.000000
Records	
Count	598,863
Containing empty values	597,788
Without empty values	1,075

Table 4.3: Simple statistics of the notes view

2. **Analysis results:** results of different medical analyses done, per patient. As an example we can consider blood-pressure measurements, urine and blood screenings etc. This data view contains 9 specific columns, whose detailed description can be found in the Appendix A. All records within this view have at least one empty column value. Some simple statistics regarding this view are shown in Table 4.4.

3. **Diagnoses:** this part of the data contains information regarding the diagnoses assigned to the patients at different points in time. This data view contains 7 specific columns, whose detailed description can be found in the Appendix A. All records within this view have at least one empty column value. Some additional simple statistics regarding this view are shown in Table 4.5.

4. **Standing prescriptions:** records of all prescription medication given to the patients. The data on the different medications is kept in a structured manner and written down in the official, internationally accepted hierarchical classification system for medicines:

Statistical measure	**Value**
Patients	
Unique count	9,273
Mean	26.285021
Standard deviation	33.455400
Minimum	1.000000
25%	5.000000
50%	15.000000
75%	34.000000
Maximum	659.000000
Records	
Count	243,741
Containing empty values	243,741
Without empty values	0

Table 4.4: Simple statistics of the analysis view

Statistical measure	**Value**
Patients	
Unique count	13,320
Mean	10.840015
Standard deviation	16.398864
Minimum	1.000000
25%	2.000000
50%	5.000000
75%	12.000000
Maximum	241.000000
Records	
Count	144,389
Containing empty values	144,389
Without empty values	0

Table 4.5: Simple statistics of the diagnosis view

ATC codes [75]. Besides the ATC individual codes, the group codes and the date of the start of the prescription, the data also contains information for the end date of the prescription, for medications that are only taken for specific periods of time. This data view contains 7 specific columns, whose detailed descriptions can be found in the Appendix A. Very few records (60 out of 51,689) in this view contain empty values. Some simple statistics regarding this view are shown in Table 4.6.

Statistical measure	**Value**
Patients	
Unique count	9,027
Mean	5.726044
Standard deviation	6.915138
Minimum	1.000000
25%	2.000000
50%	3.000000
75%	7.000000
Maximum	87.000000
Records	
Count	51,689
Containing empty values	60
Without empty values	51629

Table 4.6: Simple statistics of the prescriptions view

5. **Sick leave:** records of all sick leave approved per patient and the period of time (exact from-to dates) that the patient was under sick leave. Each sick leave is recorded under a specific diagnosis noted down as its ICD-10 code. This data view contains 7 specific columns, whose detailed description can be found in the Appendix A of this report. None of the records of this view contain empty values. Some simple statistics regarding this view are shown in Table 4.7.

6. **Products:** these are records of all other billable events provided by the medical center. These include, but are not limited to: vaccinations (free or paid), advice giving over the phone or in person, simple procedures done in the hospital such as cleaning

Statistical measure	Value
Patients	
Unique count	307
Mean	3.302932
Standard deviation	2.989539
Minimum	1.000000
25%	1.000000
50%	2.000000
75%	5.000000
Maximum	13.000000
Records	
Count	1,014
Containing empty values	0
Without empty values	1,014

Table 4.7: Simple statistics of the sick leave view

of injuries etc. These are also noted down using specific codes which are not internationalized and depend on the data system used in the specific medical center. This data view contains 3 specific columns, whose detailed descriptions can be found in the Appendix A of this report. Same as the previous one, this view does not contain any empty values in its records. Some simple statistics regarding this view are shown in Table 4.8.

Statistical measure	Value
Patients	
Unique count	2,775
Mean	2.307387
Standard deviation	1.790868
Minimum	1.000000
25%	1.000000
50%	2.000000
75%	3.000000
Maximum	20.000000
Records	
Count	6,403
Containing empty values	0
Without empty values	6,403

Table 4.8: Simple statistics of the products view

Looking at the data available from a different point of view, we can also classify it as: **structured** ***(analysis results, diagnoses, standing prescriptions, sick leave and products)*** and **unstructured** ***(patient notes)*** data. This categorization of the data was previously used in a similar study where the structured and unstructured data was differently treated towards the final prediction goal [60].

Both these different splits of the data, either by data type (structured/unstructured) or by the semantic data group (diagnoses, analysis, prescriptions etc.) can be considered as different "views" of the patient's medical history, where the final goal is assigning the correct diagnosis to the patient and prescribing the appropriate medication. As we shall see, this fact can be easily applied in a multi-view learning setting to improve the predictive performance of a classifier in a medical setting.

4.4 Preparation

Before we can consider a specific dataset for machine learning we have to "transform" the data and get it into a suitable format. This means that we have to clean the data from any information that is unnecessary for the algorithm to consider, remove redundant, missing and/or corrupted data and finally optimize the data representation for the machine learning process [76]. This is typically a process which takes a large portion of time (even up to 60%-80%) of the whole machine learning process and is considered "a necessary evil" by many machine learning practitioners [77].

Within this section the data preparation process shall be described. This consists of the following two phases: *initial data preprocessing* and the *main feature extraction process* (along with all of the decisions that have been made). The process is visualized in figure 4.2.

4.4.1 Preprocessing

This section explains the first step of the data transformation process - the cleaning phase. In this particular case, each of the above mentioned semantically grouped data sources (Section 4.3) was prepro-

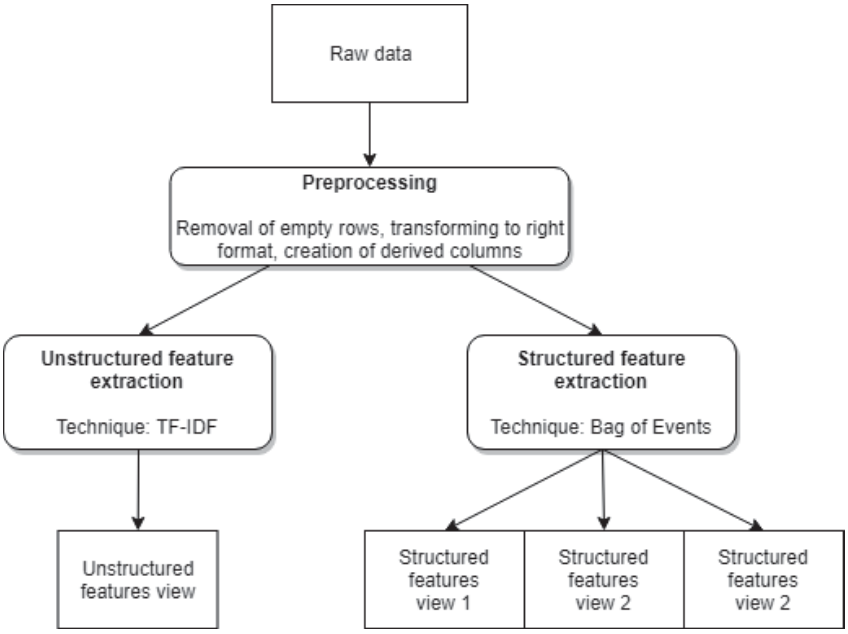

Figure 4.2: The data preparation process.

cessed separately before merging the different information in the feature extraction step.

The following common data preprocessing steps were done for each view (not necessarily in the specific order):

- Removal of redundant, unneeded columns.
- Imputation of empty values: 'U' was used for imputation value of the categorical values, while the continuous values were handled on a case-by-case basis.
- Removal of any leftover empty rows (some columns were not handled in the previous imputation step so it is imperative to remove empty values before training).
- Replacement of categorical alphabetic characters with numeric ones since many machine learning algorithms cannot handle alphabetic characters.
- Conversion of date-time columns to the proper types since they

were initially considered as integer (timestamp) values by the interpreter.

- Creation of derived columns, if needed on a view-per-view basis.

Additional data preprocessing was done on the patient notes view. More specifically, the text from the columns "note", "local_keyword" and "patient_notes_interpretation_as_string" was concatenated into one derived column named "text", while the original columns were deleted. This derived column will be used in the feature extraction process instead of the original ones. The idea is to have all of the information from the different note columns into one place so that the text feature extraction process can be simplified.

4.4.2 Feature extraction

The feature extraction phase processes each view separately and connects it to an appropriate diagnosis within the diagnosis records view. The final feature set has the six data views that we already talked about (*patient notes* a.k.a. unstructured view, *analysis results, past diagnoses, prescriptions, sick leave and products*) along with one derived view: *date trends*. This view was created based on the date and time data that is present within the diagnosis view and is meant to represent the seasonality trends in the data.

It is important to stress out that hard boundary has been drawn when it comes to extracting features from the structured data part vs. the unstructured data part. Several different feature representation approaches have been considered as for either of these data types. Each of these approaches are discussed here in detail along with a justification on the decision to use one over the other techniques.

Extraction from structured data

Several strategies have been considered as main (structured) feature representations for this project:

- The simplest technique that was considered is binary history vector representation [9] where the all of the possible events in a

patient's EHR are represented as either 'occurred' (1) or 'not occurred' (0) before the time point of a certain predictive event (diagnosis). In this case we do not take into account whether the event has occurred multiple times, we only note down the presence or absence of the event.

- Another technique that was considered is the Bag of Events and its variants [22] (Bag of Binned Events and Bag of Weighted Events) which has successfully been applied to the field of adverse drug reactions prediction. Within this approach, we pre-define a window D which is the number of days prior to a given target label (diagnosis) where we consider the events that have happened as the predictive dependents of that particular diagnosis. To be more specific, let us suppose that n_d represents the number of occurrences of an event x on day d which is within the range of D days before our target label (diagnosis) y. The feature that represents x within these D days would then be defined as: $\sum_{d=1}^{D} n_d$ creating the Bag of Events feature representation.

- The Bag of Binned Events version of this approach is similar, with the difference of counting each daily occurrence of event x within D days as a separate feature so that: $x_1, x_2, ..., x_D$ is the set of features representing the event x within the range of D. The most advanced version of this approach is the Bag of Weighted Events where we assign different weights w to the occurrences of event x depending on the time distance between the specific occurrence of x and the target diagnosis y.

- The last feature representation version of the EHR events that was considered is a custom variation of the Bag of Events along with another sort of representation named Piecewise Aggregate Approximation that has also been used in the case of medical diagnosis for adverse drug events [23]. This combined feature representation would use the bag of events approach for events that do not have numerical measurements, i.e. events that are only noted down for their occurrence (such as diagnoses, sick leave, vaccinations, prescriptions etc.), while the Piecewise Aggregate

Approximation (PAA) approach would be used for events that can be measured as numerical values (blood screenings, urine screenings, blood-pressure measurements etc). The essence of the Piece Aggregate Approximation approach is to split a given event of a numerical measurement into w equal sized windows such that the i-th element of the feature representing this event (x) would be calculated as:

$$x_i = \frac{w}{n} \sum_{j=\frac{n}{w}(j-1)+1}^{\frac{n}{w}i} t_j \tag{4.1}$$

where t is a value which belongs to the specific window i. In other words, each $x_1, x_2, ..., x_w$ will be represented as the mean value of the event measurements that belong to the specific window that x_i represents.

After considering all of the above feature representation variants, the Bag of Events was chosen as the simplest, powerful enough feature representation technique for this task. It has been shown that the Bag of Binned Events, the Bag of Weighted Events [22] as well as the Piecewise Aggregate Approximation [23] approaches do indeed yield better results in the case of diagnosis prediction, but they are not suitable in a project such as this one, where a comparison between several approaches is being made, and the feature representation used is not the main issue of interest. The approach that was chosen (Bag of Events) for the structured features part of this study is somewhat more complex than the simple binary event feature vector representation [9] and somewhat simpler than the other techniques described above (PAA or Bag of Binned/Weighted Events) which makes it a perfect candidate in the trade-off between feature complexity and ease of understanding of the approach itself.

In order to extract the features according to the Bag of Events representation technique, we have to choose the D parameter (that we already mentioned). In this particular case, the value of the D parameter was set to 30 days for the structured feature extraction. The choice of this parameter is discussed in detail within the last subsection of this chapter.

In order to distinguish one view from the others, each feature is labeled with a suitable prefix denoting the view that the feature belongs to. Each feature is additionally named according to the main key/code that is counted within that specific view. For example, for the diagnosis and the sick leave views this is the column: diagnosis_group_name which denotes the name of the high level ICD-10 group (that we discussed above) in Swedish. For the products view this is the product_code column, for the analysis this is the analysis_name column and so on. The final set of the structured data features extracted has the following form:

Diagnosis	analys_BP	presc_A01A	diag_HUD SJD	sick_NEUR	prod_DQ017
CIRK SJD	0	1	3	1	1
NEUR	1	2	0	1	0
HUD SJD	0	0	1	0	0

Table 4.9: The final feature representation form.

Extraction from unstructured data (text)

In the case of feature extraction from the unstructured data subset, two different strategies were considered that belong to the Bag of Words group of representations [6]. The concept has been given this name since the process is analogous to putting all of the words of a given text corpus into a bag and pulling the words one by one. While we lose the ordering of the words, we have the count of the words which usually represents the relative importance of the specific word within the corpus. This is also called a unigram bag of words model. A variation of the bag of words model is the bag of n-grams model for text vectorization, since we can assign different thresholds for the number of words that are considered as a whole "piece". For example, instead of a single word, 3 words at once (if next to each other) can be considered as a counting piece.

The simplest representative technique which belongs to the model described above is the count vectorization approach: given a document, the count vectorizer considers each word as a feature and assigns a value to the cell which is the number of occurrences of the

specific word within the processed document.

Another, more advanced technique that was considered as a text extraction approach for this project is the TF-IDF (Term Frequency - Inverse Document Frequency) vectorizer [78] [79]. The idea behind TF-IDF is to solve the problem of very frequent words which carry very little information (such as "the" or "is"). The TF-IDF is calculated as follows:

$$tf - idf(t, d) = tf(t, d) * idf(t) \tag{4.2}$$

where t is the term and d is a document which contains the specific term within a larger corpus. The term frequency (TF) in this case is calculated as the frequency of the term t in document d divided by the maximum number of occurrences of any term k within that same document:

$$tf = \frac{f_{t,d}}{max_k f_{k,d}} \tag{4.3}$$

The idf, on the other hand, is defined as the logarithm of the division between the total number of documents by the ones which contain term t:

$$idf_t = \log_2(N/n_t) \tag{4.4}$$

Finally, for this specific task, the TF-IDF approach was chosen as the primary feature extraction technique from the unstructured data subset since the relative importance of the features (words) can be easily determined which means that less (but more important) features can be used for the predictive part of this project, effectively reducing (however, not eliminating) the curse-of-the-dimensionality problem. Additionally, less features generally means less noise within the predictive variables which usually yields more accurate predictions.

Same as in the case for the structured features, here a decision had to be made regarding the amount of time that shall be considered as the predictive information before a given diagnosis target. For the sake of simplicity, we shall consider this parameter in days (D), same as was the case in the previous section. The choice of this parameter for the unstructured features was $D = 7$. Which means that a whole week of patient notes history were considered before each and every diagnosis event. This differs from the choice of 30 days for the structured features. The details of the choice are discussed in the next section.

The amount of feature variables to be used for the unstructured data representation was chosen through trial and error. Several different feature extractions were done with distinct numbers of variables (10,000; 20,000; 30,000 and 40,000). The predictive performance of the different sets of variables that were chosen was evaluated using a Multinomial Naive Bayes learner [48]. The feature sets that were extracted using several different numbers of variables gave the following F1 performance scores on the diagnosis classification task (using a 3-fold cross validation training method):

Maximum text features	**Test score**	**CV score**
10,000	0.507	0.502
20,000	0.501	0.487
30,000	0.493	0.475
40,000	0.481	0.462

Table 4.10: Different maximum text features experiments F1 performance scores.

As we can observe from the performance scores above, the predictive and generalization performances drop steadily. Most likely, this fact is due to the inevitable inclusion of unnecessary noise in the data as more variables are included into the predictive set. The final unstructured feature set to be used in the predictive part of this work is the version using 10,000 features extracted using TF-IDF from the text data.

Finally, same as the extract structured features, the final set of text variables had a similar structure, e.g:

Diagnosis	text_akuten	text_artrit	text_domningar
CIRK SJD	0.081	0	0.130
ENDOKRIN	0	0.073	0
HUD SJD	0.75	0.543	0

Table 4.11: The final text feature representation.

Choice of D parameter

Initially, 3 candidate D parameter values were considered: 1 week (7 days), 1 month (30 days) and until the start of the EHR (which is 5 years or roughly 1,826 +- 1 days). Each of these D parameter values were evaluated on a representative subset of the medical data (approximately 15% of the records) by modeling an appropriate diagnosis predictive classifier. This was done so that we can get a fair idea of how each D parameter affects the final diagnosis precision of a classifier. Several experiments (using a 3-fold cross validation training method and a Multinomial Naive Bayes classifier) with different samples were done using each value of D and the following average F1 performance scores were obtained:

Data type	**Full history**	**30 days**	**7 days**
Structured features	0.31	0.45	0.45
Unstructured features	0.4	0.43	0.45

Table 4.12: Different socres of the D-parameter experiments.

It was finally decided to use a value of **7 (D) days history for the unstructured data** and **30 (D) days for the structured data**. Even though the experiments revealed similar performance for the 30 and 7 D values for the structured data, the reasoning is that some analysis results or previous diagnoses (e.g.) could be more than 7 days in the patient's history but might still affect the target diagnosis.

Chapter 5

Experiments and evaluation

The following section explains the modeling part of the project along with the decisions regarding the machine learning algorithms used. The first section of this work presents the specific ***experiments*** applied as a process towards answering the specified research question. The second section presents the specific ***evaluation methodology*** towards quantifying the specific results obtained from the experimental work.

5.1 Experiments

The experiments performed within the scope of this project are grouped into the two high-level approaches: *single-view* and *multi-view learning*. Under the scope of single view learning, we use a fusion of multiple data views to accommodate the data to the learning approach, instead of the other way around, while in the concept of multi-view learning, we are accommodating the learning methodology to the available data. Under the scope of single-view learning there is one experiment which is applied on the data and it is taken as the baseline score that will be compared to the other experiments performed.

5.1.1 Single-view learning

The single-view experiment performed in the context of this thesis is the following: the feature extraction techniques described in section 4.4.2 were used to extract a given set of features into multiple views.

Since the single-view approach does not support learning from multiple views, the final feature set was a simple merge of all extracted views together. This merged resulting feature set was afterwards used as input into a machine learning algorithm to find a suitable hypothesis that maps the data to the target values (diseases to be predicted). Figure 5.1 depicts a visual representation of this single-view experiment. For the sake of simplicity, we will denote the learned model (hypothesis) of this experiment SV.

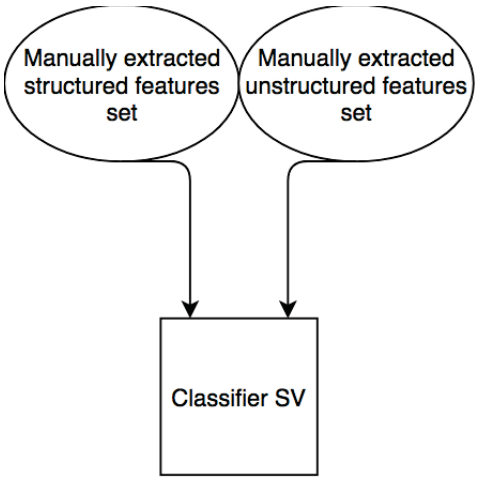

Figure 5.1: The single-view classification experiment.

5.1.2 Multi-view learning

Canonical Correlation Analysis for dimensionality reduction

Canonical correlation analysis [34] as one of the best-known methods for multi-view learning can be applied for dimensionality reduction of high dimensional data [41]. In this case, it is being applied to extract the main components of the two views of data: structured and unstructured. Since we can explicitly define how many component variates C we would like to extract, experiments using different values of C were performed to see the effect of the different numbers of components on the final prediction scores of the models. The experiments performed used the following C values: 2, 4, 6 and 8. Three

separate hypotheses were evaluated under each C value: hypothesis found using the variates extracted from the structured dataset, hypothesis found using variates extracted from the unstructured dataset and a final hypothesis found by combining both variate components extracted from the structured and unstructured data fused together. The scope and the steps of this particular experiment are visualized on figure 5.2

Will denote the learned hypotheses under the scope of this experiment as CCAS for the structured features, CCAU for the unstructured features and CCASU for the structured and unstructured low-dimensional features fused together.

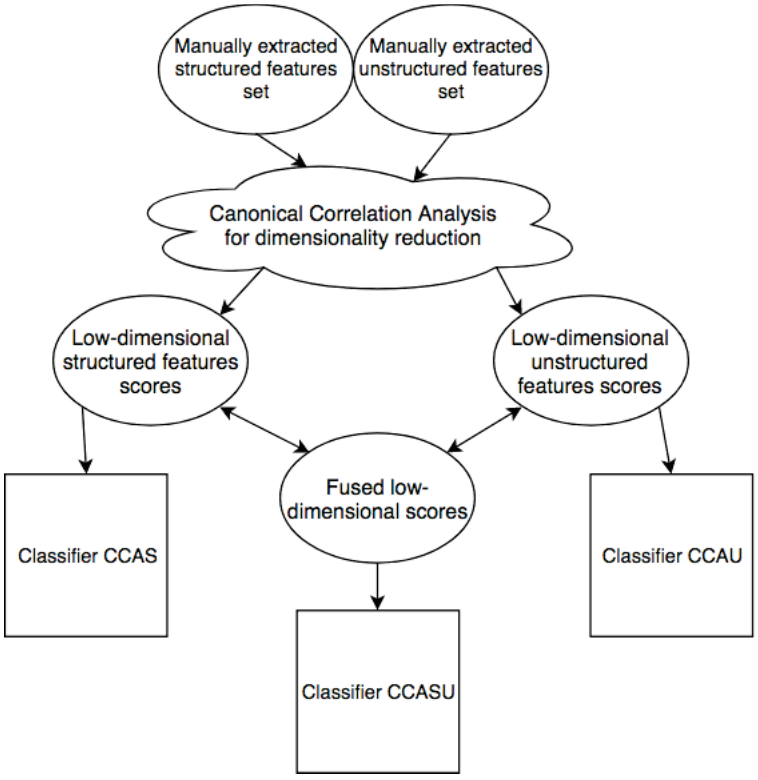

Figure 5.2: The multi-view canonical correlation analysis dimensionality reduction classification experiment.

Multi-view stacked ensemble per data type

In the context of this thesis there were two experiments done using the approach of multi-view stacked ensemble models (explained in detail in section 2.1.3). The first experiment that was performed using this method is by combining two data views categorized by the data type that the features were extracted from (structured and unstructured data). This is similar to the experiment performed using the CCA approach for dimensionality reduction. The experiment idea is to combine the low-dimensional prediction outputs of the two classifiers trained on the structured and unstructured data views separately and use these to learn a new, final classifier. The experiment steps are visualized in figure 5.3.

We shall denote the models trained under the scope of this experiment as PTS for the structured features and PTU for the unstructured features. The final stacked classifier shall be denoted as PTSU.

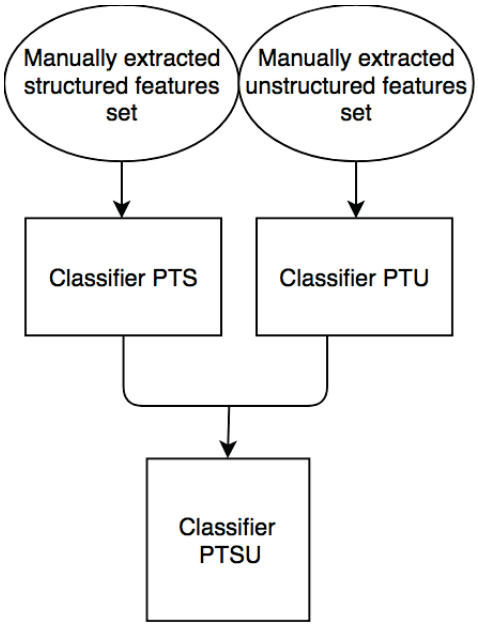

Figure 5.3: The multi-view stacking ensemble per data type classification experiment.

Multi-view stacked ensemble per data view

In the experiment that was performed using the approach of multi-view stacking we are training a classifier per data view where the data is categorized onto different groups semantically instead of by type. As a reminder, we have the following semantic data views: date trends view, diagnosis view, analysis view, prescriptions view, sick leave view, products view and notes view where the notes view is equal to the unstructured view discussed in the previous section. Each view's data is used to train a different classifier and the prediction probabilities of each classifier are fused together to create one low level dimensional dataset which is then used to train one final classifier. This steps of this second multi-view stacked ensemble experiment are visualized in figure 5.4.

The potential hypotheses learned in this experiment shall be denoted as follows:

- PVDT: Date trends classifier
- PVD: Diagnosis classifier
- PVA: Analysis classifier
- PVP: Prescriptions classifier
- PVS: Sick leave classifier
- PVPR: Products classifier
- PVN: Notes (unstructured data) classifier

The final stacked classifier that has been trained with the outputs of the predictions of the previous classifiers shall be denoted as PVF.

5.2 Hyperparameter optimization

In order to get to find an optimal hypothesis under each experiment, the machine learning algorithms used had to be optimized to accommodate the specific use-case that we are dealing with. Both of the

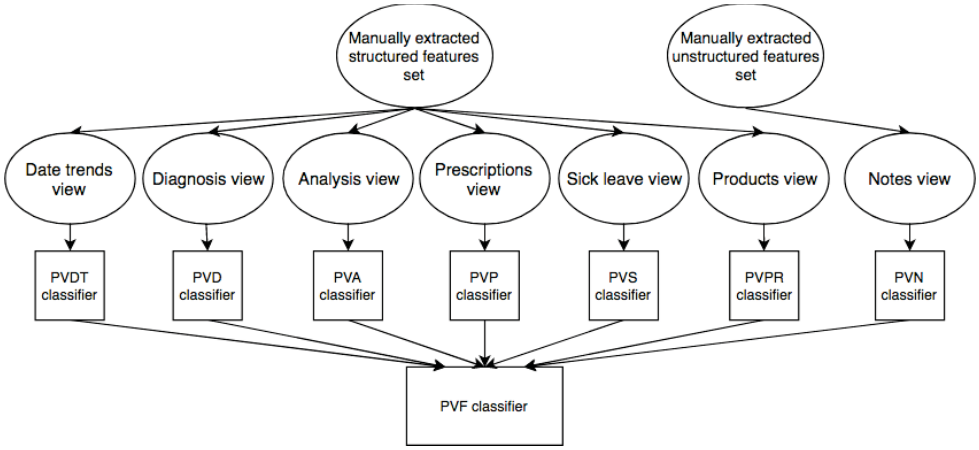

Figure 5.4: A visual representation of the multi-view stacking ensemble per data view classification experiment.

main classifiers used (Multinomial Naive Bayes and Random Forest) has had their hyperparameters [80] optimized using a grid search approach.

5.2.1 Random forest

Building a robust and accurate random forest models requires tuning of numerous hyperparameters.

Even though there is a large number of parameters that can be tuned in the case of the random forest implementation used in this project [81], among the more important hyperparameters to be optimized in this specific case are the following:

- Number of estimators (decision trees): this is the most difficult hyperparameter to be optimized. The number of estimators to be trained highly depends on the training data distribution, the number of attributed (variables) and the number of target classes. The best approach towards finding the optimal number of estimators is to perform a grid search on the whole possible hypotheses space, which is the approach taken in this work.
- Maximum node depth: specifies what should be the maximum depth of the individual trees in the random forest model. The

experiments done on a sample of the total disease diagnosis prediction data available showed that the optimum depth highly depends on the sampling. However the default setting: expanding nodes until all leaves are pure or they contain less than the minimum number of samples specified (explained next) seems to provide reasonable results in most cases.

- Minimum samples split: this hyperparameter specified what is the minimum number of samples required to split an internal node (non-leaf). If the number of samples that are at that node is less than the minimum, in that case the node will be designated as a leaf and the expansion stops.

- Node split criterion: this hyperparameter specified the criterion used to measure the information gain that a certain split will provide. The two possible values are Entropy or Gini impurity. Gini impurity was chosen because it is inherently multi-class.

5.2.2 Multinomial Naive Bayes

Considering the fact that the Naive Bayes classifier is relatively simple, the hyperparameters available for its tuning are very few. The three hyperparameters available in the specific Multinomial Naive Bayes implementation used in this project [82] are the alpha parameter (additive smoothing parameter), fit prior (whether to learn class probabilities or not, which is enabled by default) and class prior (an option to specify the prior probabilities of the classes explicitly). The only hyperparameter that was tuned during training using a grid search over the possible hypothesis space was the alpha (smoothing parameter).

5.3 Evaluation

The optimal approach that has been identified in this project shall be used in a business-context solution. As one may suspect, the models in a clinical context should have notable performance so that we can be sure that we are providing the best possible decision support to medical professionals. Considering the above, the evaluation of the

process of each experiment was done under the criteria of **predictive performance** (how accurately the found hypothesis predicts the target labels).

The performance evaluation in terms of predictive power of each hypothesis was done by evaluating both the averaged cross-validation score (training performance) and the test dataset score (which gives a good idea of the generalization performance of the hypothesis). Each experiment performed was done under a data split of 70/30% of training vs. test dataset ratio. In addition to that, each model training process was performed using 3-fold stratified cross validation. Three folds were used due to the fact that the least populated classes in the data have only 3 data examples, and we are thus unable to perform more than 3-fold **stratified** cross validation. Stratified cross-validation is a method of evaluating the performance of learning algorithms based on performing a number of k-fold splits on the set of training data T. Each of the T_k folds is used as a test set to evaluate the performance of the trained classifier on the rest of the training data $T \setminus T_k$. The keyword stratified in this case denotes a specific cross-validation technique which keeps that data class (target) balance in the data folds. This has the potential to reduce variance which makes it easier to identify the best learning algorithm [83].

In order to compare the models we need numerical metrics that will show how the model is performing. To achieve this, we can use numerical measures such as accuracy, precision, recall, ROC curve, AUC as well as the f-measure (and its variations) [84]. Among these, ones which are frequently used in classification projects with class imbalanced data are precision, recall and the harmonic mean [85] of both: the f-measure.

The precision and recall are metrics which focus on the positive examples and their predictions. Even though they do not directly capture how well the models handle negative examples, they are still very well suited for medical settings due to the fact that they show us how well the model predicts cases where we have a positive diagnosis. It is much more important to optimize the prediction of positive cases (a disease is detected) compared to the prediction of the negative cases (there is no actual disease). Essentially, we want to minimize the oc-

currences when the model predicts a negative when it was in fact a positive: even if we make a mistake and say that the patient is sick when he is, in fact, not we can do more tests to determine whether this is in fact true and fix the mistake easily.

The recall (also referred to as the true positive rate) is specifically defined by the following formula:

$$R = \frac{tp}{rp} \tag{5.1}$$

where tp represents the number of true positive cases that were predicted divided by the real number (rp) of positive cases that exist in the data.

The precision (also referred to as true positive accuracy) is defined mathematically as follows:

$$P = \frac{tp}{pp} \tag{5.2}$$

where tp (same as above) represents the number of true positive cases that were predicted divided by the number of all predicted positive cases by the classifier (both correctly and incorrectly).

Finally the performance score which unifies the precision and the recall measures is the F1 score metric [86]:

$$F_1 = \frac{2PR}{P + R} \tag{5.3}$$

The F1 score is suitable for this case due to the problem of the class imbalances and it is only high if both the precision and recall are high. The F1 score is a special case of the more general F-measures which can be specified by the following formula:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{5.4}$$

$$(0 \leq \beta \leq +\infty)$$

It is important to note that the metrics discussed are generally used in binary classification cases where we would typically have a specific target label set of positive and negative examples (2 classes). In the

case of multi-class classification, it is necessary to aggregate the measures to show the relevant score across all classes being predicted. To achieve this, we can apply **macro** or **micro** averaging aggregation of the performances of the classifier [87]. The macro averaging of the performance measure gives equal weight to each class. It will calculate the mean performance of the classifier on each class and may thus emphasize the low performance of infrequent classes. It is however, useful if the infrequent classes are important in the specific classification problem (this is not the case here, since all classes are important to us). Micro-averaging, on the other hand, will give equal weight to each classification decision on a per-example basis instead of per-class basis (micro-averaging, as the name suggests, is thus more fine-grained than macro-averaging). However, in typical scenarios where we have a noticeable class imbalance, the more frequent classes will dominate the infrequent ones. In this case we would like to give equal weight to the predictions on a per-example basis: all diagnoses are equally important to be predicted correctly. Considering this, it is more suitable to use the micro-averaging instead of macro-averaging of the performance scores.

The main performance scores reported on each experiments are always in the micro-averaged F1 measure numerical score (the harmonic mean of the precision and recall scores).

In addition to the averaged final performance scores of the trained models, per-class performance scores (precision, recall and F1-measure) shall also be reported for the best classifiers observed in each of the four experiments.

The confusion matrix [88] is another concept which numerically and visually shows how well a model performs on a per-class basis. The confusion matrix is a two-dimensional matrix which has the true label of an example on one axis, and the predicted label of the example on the other axis. An ideal model would only have values higher than 0 in the diagonal of the matrix, indicating that all examples are classified correctly.

Chapter 6

Multi-view versus single-view machine learning

The specific results obtained from each experiment along with an extensive comparison will be presented in section 6.1. The last section of this chapter (6.2) will present an extensive discussion regarding the obtained results and possible reasons why each one occurred.

6.1 Results

In order to compare the results to a baseline performance, we shall be using a majority class classifier. A majority class classifier is one which always assigns data points to the majority class. In this case, the majority class (ICD10 codes M00-M99 or the label MUSK-SKEL in our dataset) comprises a fraction of 0.199 of the whole dataset. Which means that the accuracy of this majority class classifier would be 0.199, the average precision would be 0.009 (0.199 for the majority class and 0 for all other classes) and the recall would be 1 for the majority class and 0 for all other classes (0.05 on average). The F1 score based on the precision and recall would be **0.015**.

The results presented in the following sections are split by experiment performed. Each experiment's best performing model according to the test data subset F1 score is bolded to give the reader an easy overview of which model performed best in each case.

6.1.1 Single-view learning results

The results obtained by the classifiers trained on the single-view data (all views concatenated into one) were similar for the Multinomial Naive Bayes and the Random Forest classifier. The micro aggregated F1 score on the test set of the Multinomial Naive Bayes classifier was 0.535 while the same score for the Random Forest classifier was 0.572. The per-class F1 scores ranged from 0 for the infrequent classes and up to 0.71 for the more frequent ones when on the Multinomial Naive Bayes classifier. For the Random Forest classifier, on the other hand, this range moves between 0 and up to 0.7. It seems that the Random Forest classifier also had slightly better balanced predictions for the classes: the standard deviation of the per-class micro F1 score was 0.196 for the Random Forest classifier and 0.214 for the Multinomial Naive Bayes. It appears that Multinomial Naive Bayes has excellent performance (and slightly better than the Random Forest) on classes that are very prevalent in the dataset and can be distinguished easily. For example, the classifier scored better on the classes: CIRK SJD, ENDOKRIN, HUD SJD and MUSK-SKEL all of which are represented by more than 5% of the examples in the test dataset. Table 6.1 shows the overall performance scores of the models. It is interesting to note that the fact that the test and cross validation scores have similar values per classifier shows us that both of the models generalize well.

The per-class performance scores and the confusion matrix of the best model (using the Random Forest classifier) are available in table B.1 and figure B.1 in the appendix.

Classifier	MultinomialNB		Random forest	
Score type	Test	CV	Test	CV
F1 mirco avg.	0.535	0.526	**0.572**	0.589

Table 6.1: Single view learning F1 scores results

6.1.2 Multi-view learning results

Canonical Correlation Analysis for dimensionality reduction

When it comes to the usage of multi-view CCA dimensionality reduction for classification, only the Random Forest classifier was used as a learner since the CCA dimensionality reduction can extract scores which are of negative value. This is a problem for the Multinomial Naive Bayes classifier since it cannot handle negative values due to the fact that the Multinomial Naive Bayes classifier assumes multinomial distribution. The results obtained in this experiment show an improvement of the F1 performance score of the classifiers for incremental number of components used; from 2 up to 8 components. The further increase in the number of components used (10) in this setting actually makes yields worse predictive performance: the optimal number of components is essentially 8.

One additional observation that we can witness is the fact that the best performance is achieved on the CCAS model (the structured data low-dimensional features). This can be expected, since the fact that the structured data features have considerably less dimensions than the unstructured features typically means that the feature set also has considerably less predictive noise. It logically follows that the scores extracted using the dimensionality reduction technique would also have less noise for this feature set. The standard deviation on the CCASU models on the micro-F1 scores per class was the following: 0.188, 0.226, 0.203, 0.191, 0.189, using 2, 4, 6, 8 and 10 components, respectively. Based on this, we can see that the balance of the predictions is initially low for the 2 component case - the model scores low on all classes. After this, using 4 and 6 components the model scores better on the more prevalent classes and quite badly on the classes which are not represented by many examples so the predictions are imbalanced. The scores flatten out and get relatively balanced using 8 and 10 components, where the standard deviation is below 0.2. Table 6.2 presents the results obtained using the Random Forest classifier on each number of extracted components 2-10. The per-class scores of the best model trained (CCAS with 8 components) are presented in the ap-

pendix in table B.2. The confusion matrix for the same model is also available in the appendix under figure B.2.

Components	2		4		6		8		10	
Score	Test	CV	Test	CV	Test	CV	Test	CV	Test	CV
CCAS	0.359	0.352	0.501	0.502	0.532	0.541	**0.548**	0.560	0.545	0.559
CCAU	0.301	0.286	0.372	0.368	0.393	0.390	0.424	0.426	0.423	0.426
CCASU	0.373	0.368	0.493	0.497	0.526	0.533	0.542	0.556	0.542	0.556

Table 6.2: CCA multi-view dimensionality reduction classification F1 scores results using a Random Forest classifier

Multi-view stacked ensemble per data type

When it comes to the usage of multi-view stacking per data type extracted, we can notice several interesting points. As expected, the performance of the models based on the two separate data type views showed different results depending on the classifier used (PTS vs. PTU). The structured data model (PTS) performs noticeably better when using the Random Forest classifier (0.573 test F1 score), compared to the Multinomial Naive Bayes (0.484 test F1 score). The unstructured data shows the opposite case - the Multinomial Naive Bayes scores a test F1 measure of 0.508 while the Random Forest scores a test F1 measure of 0.462. The final stacked classifier, as expected, shows a noticeably better performance when using the two strongest predictors (PTS Random Forest and PTU MultinomialNB) compared to any other combination (0.612 F1 test score for the Multinomial Naive Bayes classifier vs less than 0.6 for all others).

An interesting observation to make at this point is the fact that the Multinomial Naive Bayes classifier works better as a final stack model compared to the Random Forest classifier. This could be contributed to the fact that the prediction probabilities which are used as inputs to the final classifier generally have little to no noise compared to the original inputs with very high dimensionality. Additionally, *it seems that stacking measurably strengthens the prediction performance on the classes which we have a lot of examples for, while increasing the prediction performance on the other classes only marginally*. This can be witnesses by observing the increased standard deviation of the per-class F1 scores: the PTS

Random Forest classifier has 0.186 standard deviation, the PTU Multinomial Naive Bayes has a 0.194 standard deviation, and the final PTSU Multinomial Naive Bayes classifier has 0.247 standard deviation - a increase of 0.05 standard deviations.

Table 6.3 shows the performance for each possible combination of the PTS, PTU and the PTSU classifier models. The per-class performance metrics (precision, recall and the F1 score) of the best model (PTSU with Multinomial Naive Bayes classifier) can be observed in table B.3 in the appendix. The confusion matrix of the same model can also be found under figure B.3 in the appendix.

Model				PTSU			
	Classifier			MultinomialNB		Random forest	
		Test	CV	Test	CV	Test	CV
Structured data (PTS)	MultinomialNB	0.484	0.485	0.542	0.549	0.549	0.573
Unstructured data (PTU)	MultinomialNB	0.508	0.503				
Structured data (PTS)	MultinomialNB	0.484	0.485	0.530	0.632	0.484	0.681
Unstructured data (PTU)	Random forest	0.462	0.466				
Structured data (PTS)	Random forest	0.573	0.588	0.562	0.776	0.547	0.687
Unstructured data (PTU)	Random forest	0.462	0.466				
Structured data (PTS)	Random forest	0.573	0.588	**0.612**	0.702	0.565	0.680
Unstructured data (PTU)	MultinomialNB	0.508	0.503				

Table 6.3: Multi-view stacked ensemble per data type F1 scores results

Multi-view stacked ensemble per data view

The most interesting results were obtained under the per data view stacked ensemble experiment. Namely, *the best prediction performance score can be observed under the Multinomial Naive Bayes classifier for the diagnosis view (taken separately)*. In addition to this, the second best model was obtained for the same view using the Random Forest classifier. The worst model performance-wise can be observed under the artificially created date trends view, which has an F1 score of 0.12 for the

Multinomial Naive Bayes classifier. This could be an indication that the specific view contains mostly noisy and redundant data instead of relevant information. Additionally, the diagnosis view consistently scored high with both the Multinomial Naive Bayes classifier and the Random Forest. This consistency indicates that the diagnosis view is the most informative when it comes to the diagnosis predictions and contains a very low amount of feature noise.

One peculiar point that we can note is that the Random Forest classifier consistently scored lower than the Multinomial Naive Bayes classifier on the most informative features: the diagnosis view (0.644) and the unstructured data notes view (0.466). Looking at the performance, it would not be unreasonable to conclude that these two views are the ones which contain the least amount of feature noise. However, the fact that the Naive Bayes consistently scored higher than the Random Forest indicates that the Naive Bayes works better with low noise data compared to the Random Forest classifier. The Random Forest models, however, scored well on the less informative views: date trends (0.221 vs. 0.12), analysis view (0.299 vs. 0.279), prescriptions view (0.322 vs. 0.256), sick leave view (0.207 vs. 0.202) and the products view (0.212 vs. 0.203). This observation indicates that the Random Forest classifier works better than the Naive Bayes with noisy data.

In contrast to the observation regarding the per-class F1 standard deviation on the previous experiment, it seems that the best single view model does not have a noticeably lower standard deviation of the per-class F1 scores compared to the final stacked model (PVD 0.213 std vs. PVF 0.228 std). This is most likely due to the fact that the potential on the diagnosis view is the maximum that can be achieved in this case, and by introducing additional views into the equation we are basically introducing noise, so the standard deviation is increased since some of the per-class predictive scores are lowered, while others remain on more or less the same level.

Table 6.4 shows the F1 performance score per model and per classifier trained. Additionally, table B.4 in the appendix represents the per class performance scores recall, precision and F1 score for the best model trained (Multinomial Naive Bayes on the diagnosis view). The confusion matrix can also be found in the appendix under figure B.4.

Classifier	MultinomialNB		Random forest	
Score	Test	CV	Test	CV
Date trends (PVDT)	0.120	0.117	0.221	0.210
Diagnosis view (PVD)	**0.686**	0.683	0.644	0.643
Analysis view (PVA)	0.279	0.278	0.299	0.298
Prescriptions view (PVP)	0.256	0.257	0.322	0.316
Sick leave view (PVS)	0.202	0.203	0.207	0.207
Products view (PVPR)	0.203	0.203	0.212	0.210
Notes view (PVN)	0.510	0.501	0.466	0.466

Classifier	MultinomialNB		Random forest		MultinomialNB		Random forest	
Score	Test	CV	Test	CV	Test	CV	Test	CV
Final classifier (PVF)	0.631	0.634	0.596	0.611	0.598	0.770	0.545	0.684

Table 6.4: Multi-view stacked ensemble per view F1 scores results

After observing the results obtained on the multi-view per-view ensemble stacking, it was decided that it would be interesting to attempt a different approach: training a stacked classifier using the two best performing views out of all seven - the diagnosis view (model PVD) and the notes view (model PVN). The results obtained are shown in table 6.5. The results show minor performance improvements, which could also be attributed to the test/training random data split. It is obvious that the usage of the two best views as a stacked ensemble does not improve the final predictive performance. This could be expected, considering the fact that, as we observed in the previous trials, the multi-view stacking improves predictive performance in cases where the views contain a notable amount of noisy data (which is not the case here).

6.1.3 Comparison of the multi-view and single-view results

The results that we obtained on the experiments differed significantly from one another. In particular, the micro F1 scores of the best mod-

Classifier	MultinomialNB				Random forest			
Score	Test		CV		Test		CV	
Diagnosis view (PVD)	**0.688**		0.682		0.642		0.642	
Notes view (PVN)	0.510		0.500		0.466		0.466	
Classifier	MultinomialNB		Random forest		MultinomialNB		Random forest	
Score	Test	CV	Test	CV	Test	CV	Test	CV
Final classifier (PVF)	0.634	0.639	0.598	0.614	0.603	0.767	0.546	0.682

Table 6.5: Multi-view stacked ensemble diagnosis and notes views F1 scores results

els trained on each experiment ranged from 0.572 for the single-view learning experiment (worst score) up to 0.688 for the multi-view stacked ensemble per data view experiment (best score trained on the diagnosis view only). This is visualized in figure 6.1. The model trained only using the diagnosis view yielded the best generalization performance if we consider the fact that the test F1 score there was higher than the cross validation F1 score. The worst generalization performance (a case of overfitting) can be observed in the multi-view stacked ensemble per data type case. Here we can observe a cross validation F1 score which is 0.09 points higher than the test score. The overfitting is worse compared even to the single-view learning case, where we can observe a difference between the CV F1 score and the test F1 score of 0.017 in the case of the Random Forest classifier.

The standard deviations of the best models in each experiment were the following: 0.196 (single-view), 0.185 (CCA multi-view dimensionality reduction), 0.247 (multi-view stacked ensemble per data type) and 0.213 (multi-view stacked ensemble per data view). We can observe that the CCA multi-view dimensionality reduction technique lead to more balanced predictions for the classes, however it did not manage to increase the predictive performance in general: it yielded the worst predictive performance, even compared to the single-view case.

If we take out the diagnosis view model out of the equation (which had the best predictive performance overall), comparing the single-view against the multi-view final models we have the following micro F1 score results: **0.572** (single-view) versus **0.548** (CCA), **0.612** (multi-

view stacked ensemble per data type) and **0.631** (multi-view stacked ensemble per data view type). Again, the multi-view stacked ensemble per data view experiment has the best performance. This only reinforces the assumption that semantically splitting the dataset into different views (instead of by the data type such as structured/unstructured) can positively affect the performance scores in a multi-view approach setting.

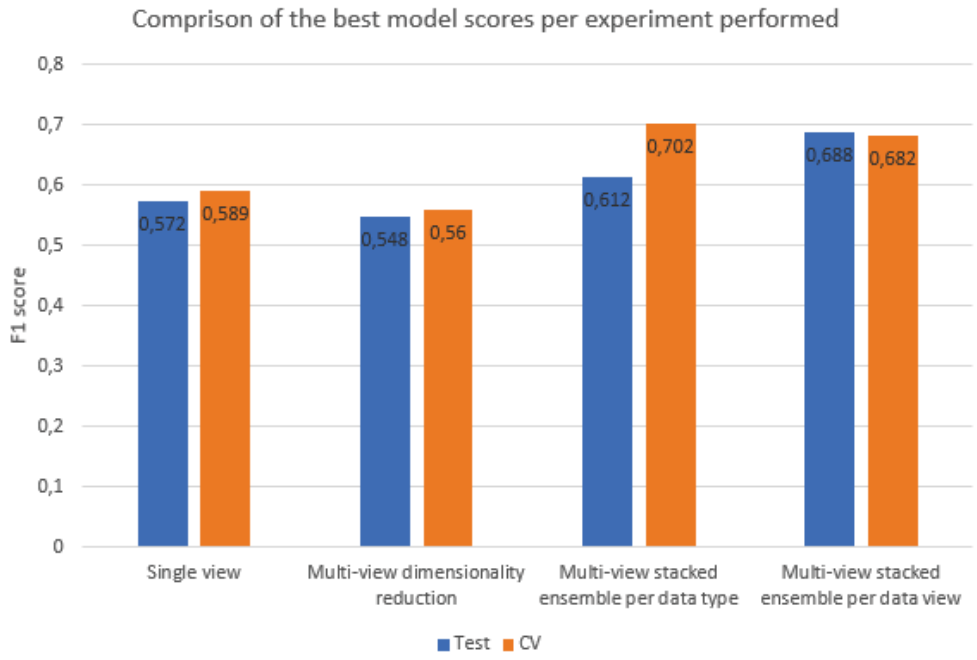

Figure 6.1: Comparison of the best model scores obtained in each experiment.

6.2 Discussion

To answer the question of whether the multi-view approach is superior to the single-view approach we have to consider all aspects of the learning and prediction process. As can be expected, the information-to-noise content in the different data views is the single most important aspect that influences the final performance of a multi-view or a single-view classifier.

If we consider the single-view approach versus the multi-view approach as a general case **we can certainly conclude that using the multi-view approach does yield better results**. Even if we consider the last experiment, where a single view (diagnosis) had the best score (overall), we can still argue that the diagnosis view was taken into account separately from the other features due to the fact that we were using the multi-view approach. As a reminder, the most common way of adopting multiple views to single-view learning is by concatenating all views together into a single dataset (the first experiment presented).

We can also consider the multi-view approach on its own. In that case, we can split the approach onto two different levels: using multi-view stacked generalization ensembles and using multi-view dimensionality reduction:

Using the dimensionality reduction approach, we observe an increase in performance as we increase the number of components used in the dimensionality reduction. This increase is up to a point where the components extracted from the data do not contribute any additional information (all main variates have already been identified). This is the point where we start to add noise to the features instead of valuable information, so the predictive power stalls or goes down. It is important to note that the lower dimensional feature sets extracted using Canonical Correlation Analysis yielded the worst results when it comes to their predictive capability: the highest F1 score obtained was only 0.548 (worse than the single-view approach). However, this approach is still worth considering, especially in cases where we have high dimensions of the feature sets, which is definitely applicable here. The approach yields results comparable to the single-view approach while, at the same time, using much lower dimension feature set: from approx. 10800 dimensions down to only 8 dimensions. This makes the training and the prediction process of the model much simpler and more efficient (time-wise).

In the case of multi-view stacked generalization, we can actually observe that out of the two experiments performed, the best score was obtained from one, single view out of seven possibilities. Considering the fact that one of the ideas of multi-view learning is view relationship analysis and knowledge discovery, we can certainly say that that

particular goal has been achieved. It is easy to conclude that the diagnosis view has the most informative features, closely followed by the unstructured data view (notes) while all other views do not even closely reach the predictive performance of the first two. We can conclude that diagnosis view and the notes view contain mostly valuable information (more than half), while the other views contain more noise than data, as observed by the predictive power of the models trained on those views. In multi-view terms this can be considered as a case where the two aforementioned views satisfy the sufficiency assumption (explained in 2.1.2) that is important for the case of co-training.

It is important to mention though, that all models reached a much better performance compared to the majority class baseline, which had an F1 score of **0.0159**.

Let us now consider the question that we posed at the beginning of this report (section 1.4): **What performance does the single-view machine learning approach have in comparison with a multi-view approach when it comes to predicting diagnoses in primary healthcare?** We can safely argue that *the overall performance of the multi-view approach was better according to all metrics used*, considering all aspects of both approaches.

One additional interesting fact that could be observed in all of the experiments under single-view and multi-view stacking is the fact that the Random Forest classification algorithm yielded notably better scores than the Multinomial Naive Bayes when working with noisy data. The Multinomial Naive Bayes, on the other hand, had noticeably better scores on data with minimal or no noise. This was most readily observed in the cases of single-view diagnosis classification as well as the usage of the Multinomial Naive Bayes classifier as a final stacked ensemble predictor. It had better results in both the per data type and per data view multi-view stacked ensembles. We can conclude that **the multi-view stacking approach works rather well as a noise reducing technique**: only the most informative information is kept in the lower subspace outputs that are given by the learners on the higher levels in the stack (the ones which deal with the data views directly). In addition to this, **multi-view stacking could also help identify sets of redundant and noisy views** - ones which can potentially be re-

moved from the problem as a whole, thus reducing the curse of the dimensionality problem. We could argue that the diagnosis view as a single data subset is all that is needed to reach satisfactory predictive performance and that we can simply discard the other views completely. However, one should be cautious before making such conclusions since this fact may not prove to be correct if we have more data. More research is needed into how the different views, split in this (or a similar) way affect the performance of a classifier, especially if we have more data examples than available in this project. Moreover, we should be careful before using the multi-view stacked ensemble approach in cases where the data views are clean and the sufficiency assumption is satisfied.

Chapter 7

Conclusions and future work

7.1 Discussion

Disease prediction is a complex issue in which a lot of the current machine learning research is focused. There are many problems and areas to be explored on the specific topic, but the scope of this project was to determine whether the multi-view approach could be a sound idea towards improving the standard way of dealing with the issue of disease diagnosis. The best approach identified here will be used in a business-context solution as a part of a digital solution for medical decision support and patient data exploration.

The feature set that was used in this project was constructed by taking several ideas from different studies in the same (or a similar) research area. The final feature dataset was constructed from raw electronic health records data taken from a hospital from the Southern part of Sweden. It is important to mention that each geographical area typically has a different set of diagnosis distributions and different studies might yield different individual prediction results. However, the approach and the idea can be applied to any geographical area worldwide.

The predictive work done in this project attempts to utilize two multi-view approaches: multi-view dimensionality reduction using CCA, and multi-view stacked generalization ensembles. Although the mutli-view dimensionality reduction has been used on multiple

occasions in the area of healthcare (mostly as a knowledge discovery method), the multi-view stacked generalization approach has not been previously applied in the case of primary healthcare. The results that we have obtained show that multi-view learning as an approach is ideal as feature analysis technique to reveal the views which carry the most information. This is especially true in cases where we do not have experts at hand to help with the feature extraction/definition process. E.g. we were able to determine that the diagnosis view indeed had the most information contained that can help us make better diagnosis predictions, and thus provide better decision support to medical professionals.

The data that was used in this project was semantically split into 7 different views based on the attributes contained in each subset. The multi-view approach allowed us to identify the view with the least and most amount of noise. This step - splitting the data semantically into different views) - is not always as straightforward or possible. Other studies utilizing multiple views of data should especially take care of using one of the approaches presented here, since the multi-view approach seems to be working very well as a noise removal technique (in a figurative sense of the word). The multi-view approach is certainly a viable and recommended option for other studies, not involving medical data, and dealing with noisy data.

7.2 Future work

Besides the two multi-view methods that were utilized in the project (multi-view stacking and dimensionality reduction), the multi-view approach in general contains other well known approaches that might also be interesting in the healthcare problem area. One such example is the co-training approach which we briefly mentioned in subsection 2.1.2. The idea behind co-training is to maximally utilize unlabeled data examples to maximize the agreement of a classifier on two distinct views of labeled + unlabeled data. The approach trains a classifier on the labeled dataset and classifies the set of unlabeled examples. A certain number of the unlabeled examples along with their given labels

by the classifier are taken out of the unlabeled dataset and are put into the pool of labeled data. The process is repeated until the unlabeled dataset is essentially empty. It has been shown that this approach can increase the predictive performance of a model compared to only using labeled data [19]. This could be hugely beneficial in the healthcare area, where it is important to have as much predictive power as possible. Additionally, it should be relatively straightforward to generate a set of unlabeled data examples out of EHRs, using a similar approach as we do for the labeled data in this case (section 4.4).

In the case of standard classification approaches, the loss function looks at a single data example at a time when optimizing the loss function of the model (pointwise ranking approach). In contrast to this, a more natural approach towards loss optimization is to look at a pair of data examples and to predict which one is more appropriate in relation to the other [49]. This idea can be easily utilized in classification and there are a number of learner implementations which already provide the possibility to specify the ranking approach for classification. One such example is the XGBoost [50] implementation of the gradient boosting trees [89] learning concept. Even though the scope of this project is not to look into the performance of a classification algorithm utilizing the pairwise ranking approach, this has been done in parallel with the two standard learners mentioned above (Multinomial Naive Bayes and Random Forest). The performance scores and the specific results obtained using the pairwise ranking classification approach can be seen in appendix C. A discussion and an analysis of the performances comparing pairwise vs pointwise classification ranking approaches will be considered in some future work project.

Another important aspect that we have taken into account in this project is the feature set used to drive the diagnosis predictions. There are several well-accepted and known techniques (manual or through the usage of auto-encoders) for feature extraction out of the temporal data contained in the EHRs. In this case we considered a manual feature extraction technique using a simple representation model (Bag of Events). Several different past intervals of the data were considered in the process (feature windows). The prediction window is highly important, since it contains all of the data that shall be used

to point to a specific disease. However, not much research has been done on what is the optimal feature extraction window in the case of primary healthcare. Instead, most authors use arbitrary window intervals chosen through trial and error (which is also the case here). An interesting future study to consider would be a thorough examination of the different possible time intervals specifically for the problem of disease prediction in primary healthcare. This is especially important in the case of primary care due to the fact that patient EHRs contain very general data, and the blueprint of each disease may be contained closer or further away from the actual diagnosis event.

Another interesting possibility for future studies to be considered is to use actual feature values instead of the bag of events paradigm. This can be considered mostly in the case of the patient analyses data (blood and urine screenings, bloodpressure measurements etc.). This idea has already been applied in a study within the healthcare area. More specifically, in predicting haematological diagnoses from a set of laboratory blood test results [11]. However, since that study attempts to predict a small set of diagnoses, the features were prepared by consulting experts from the area and largely by trial and error. The authors of that work show that the idea of using specific feature values (where each feature represents a given blood parameter) work remarkably well when it comes to predicting a set of haematological diagnoses. The study in that case differs from the one presented in this report in final prediction targets (a small set of disease targets vs. a vast spectrum of primary healthcare diagnoses). We can easily conclude that the blueprints of a vast range of haematological diseases (and not limited to) are contained within the blood test results of patients. Since the set of haematological disease is also a part of the set that is being predicted in this study, it would be interesting to see if the usage of the actual numerical results of analyses will make a difference in the predictive performance of a primary healthcare diagnosis system. Some ideas on how to transform a set of temporal EHR data to specific features using numerical analysis results was discussed in the feature extraction section 4.4.2.

Chapter 8

Bibliography

[1] (2018) The determinants of health. World Health Organization. [Online]. Available: http://www.who.int/hia/evidence/doh/en/

[2] A. Schneider and S. Breitner, "Temperature effects on health-current findings and future implications," *EBioMedicine*, vol. 6, pp. 29–30, 2016.

[3] (2014) Global health expenditure database. World Health Organization. [Online]. Available: http://apps.who.int/nha/database/Regional_Averages/Index/en

[4] Eurostat - gross domestic product at market prices. Eurostat. [Online]. Available: http://ec.europa.eu/eurostat/tgm/refreshTableAction.do;jsessionid=vVtY3V33xwX7yLtqv19x9ATq-gxFcPZQhEn9geDbqYIOfBCPuhbn!-510604984?tab=table&plugin=1&pcode=tec00001&language=en

[5] (2017) Determinants of health. GoInvo Design. [Online]. Available: http://www.goinvo.com/features/determinants-of-health/

[6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.

[7] What is an electronic health record (ehr)? Office of the National Coordinator for Health Information Tech-

nology. [Online]. Available: https://www.healthit.gov/faq/what-electronic-health-record-ehr

[8] What information does an electronic health record (ehr) contain? Office of the National Coordinator for Health Information Technology. [Online]. Available: https://www.healthit.gov/faq/what-information-does-electronic-health-record-ehr-contain

[9] I. Vasiljeva and O. Arandjelovic, "Diagnosis prediction from electronic health records using the binary diagnosis history vector representation," *Journal of Computational Biology*, vol. 24, 07 2017.

[10] M. Nilashi, O. bin Ibrahim, H. Ahmadi, and L. Shahmoradi, "An analytical method for diseases prediction using machine learning techniques," *Computers and Chemical Engineering*, vol. 106, pp. 212–223, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135417302570

[11] G. Gunčar, M. Kukar, M. Notar, M. Brvar, P. Černelč, M. Notar, and M. Notar, "An application of machine learning to haematological diagnosis," *Scientific reports*, vol. 8, no. 1, p. 411, 2018.

[12] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, P. J. Liu, X. Liu, M. Sun, P. Sundberg, H. Yee, K. Zhang, G. E. Duggan, G. Flores, M. Hardt, J. Irvine, Q. Le, K. Litsch, J. Marcus, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum, K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. Howell, C. Cui, G. Corrado, and J. Dean, "Scalable and accurate deep learning for electronic health records," *ArXiv e-prints*, Jan. 2018.

[13] M. B. Bibok, A. M. Penn, M. L. Lesperance, K. Votova, and R. Balshaw, "Development of a multivariate clinical prediction model for the diagnosis of mild stroke/tia in physician first-contact patient settings," *bioRxiv*, 2016. [Online]. Available: https://www.biorxiv.org/content/early/2016/11/22/089227

[14] S. Zhe, Z. Xu, Y. Qi, and P. Yu, "Supervised heterogeneous multiview learning for joint association study and disease

diagnosis," *CoRR*, vol. abs/1304.7284, 2013. [Online]. Available: http://arxiv.org/abs/1304.7284

[15] World Health Organization, *International statistical classification of diseases and related health problems*, 5th ed. World Health Organization, 20 Avenue Appia, 1211 Geneva 27, Switzerland: WHO Press, 2016, vol. 2.

[16] C. X. Ling and V. S. Sheng, *Class Imbalance Problem*. Boston, MA: Springer US, 2017, pp. 204–205. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_110

[17] D. Storcheus, A. Rostamizadeh, and S. Kumar, "A survey of modern questions and challenges in feature extraction," *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges, NIPS*, pp. 1–18, 01 2015.

[18] E. Keogh and A. Mueen, *Curse of Dimensionality*. Boston, MA: Springer US, 2017, pp. 314–315. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_192

[19] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT' 98. New York, NY, USA: ACM, 1998, pp. 92–100. [Online]. Available: http://doi.acm.org/10.1145/279943.279962

[20] J. Zhao, X. Xijiong, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, 02 2017.

[21] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968

[22] J. Zhao, A. Henriksson, M. Kvist, L. Asker, and H. Boström, "Handling temporality of clinical events for drug safety surveillance," in *AMIA Annual Symposium Proceedings*, vol. 2015. American Medical Informatics Association, 2015, p. 1371.

[23] J. Zhao, P. Papapetrou, L. Asker, and H. Boström, "Learning from heterogeneous temporal data in electronic health records," *Journal of Biomedical Informatics*, vol. 65, pp. 105–119, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1532046416301654

[24] J. Zhao and A. Henriksson, "Learning temporal weights of clinical events using variable importance," in *BMC Med. Inf. and Decision Making*, 2016.

[25] N. E. Kass, M. R. Natowicz, S. C. Hull, R. R. Faden, L. Plantinga, L. O. Gostin, and J. Slutsman, "The use of medical records in research: what do patients want?" *The Journal of Law, Medicine & Ethics*, vol. 31, no. 3, pp. 429–433, 2003.

[26] Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, vol. L119, pp. 1–88, May 2016. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC

[27] (2015) Sustainable development goals. United Nations. [Online]. Available: https://sustainabledevelopment.un.org/sdgs

[28] A. Håkansson, "Portal of research methods and methodologies for research projects and degree projects," in *The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOMP 2013; Las Vegas, Nevada, USA, 22-25 July*. CSREA Press USA, 2013, pp. 67–73.

[29] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[30] T. Afonja. Kernel functions. Towards Data Science. [Online]. Available: https://towardsdatascience.com/kernel-function-6f1d2be6091

[31] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[32] S. Dasgupta, M. L. Littman, and D. A. McAllester, "Pac generalization bounds for co-training," in *Advances in neural information processing systems*, 2002, pp. 375–382.

[33] Y. Li, M. Yang, and Z. Zhang, "Multi-view representation learning: A survey from shallow methods to deep methods," *arXiv preprint arXiv:1610.01206*, 2016.

[34] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[35] I. Jolliffe, "Principal component analysis," in *International encyclopedia of statistical science*. Springer, 2011, pp. 1094–1096.

[36] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[37] T. G. Dietterichl, "Ensemble learning," *The Handbook of Brain Theory and Neural Networks*, 2002.

[38] B. Wei and C. Pal, "Cross lingual adaptation: an experiment on sentiment classifications," in *Proceedings of the ACL 2010 conference short papers*. Association for Computational Linguistics, 2010, pp. 258–262.

[39] C. Shen, M. Sun, M. Tang, and C. E. Priebe, "Generalized canonical correlation analysis for classification," *Journal of Multivariate Analysis*, vol. 130, pp. 310–322, 2014.

[40] Z. Zhang, M. Zhao, and T. W. Chow, "Binary-and multi-class group sparse canonical correlation analysis for feature extraction and classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2192–2205, 2013.

[41] D. P. Foster, S. M. Kakade, and T. Zhang, "Multi-view dimensionality reduction via canonical correlation analysis," Toyota Technical Institute-Chicago, Tech. Rep., 2008.

[42] X. Li, S. Qian, F. Peng, J. Yang, X. Hu, and R. Xia, "Deep convolutional neural network and multi-view stacking ensemble in ali mobile recommendation algorithm competition: The solution to the winning of ali mobile recommendation algorithm," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 1055–1062.

[43] E. Garcia-Ceja, C. E. Galván-Tejada, and R. Brena, "Multi-view stacking for activity recognition with sound and accelerometer data," *Information Fusion*, vol. 40, pp. 45–56, 2018.

[44] Z. Ouyang, X. Sun, J. Chen, D. Yue, and T. Zhang, "Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things," *IEEE Access*, vol. 6, pp. 9623–9631, 2018.

[45] M. Aly, "Survey on multiclass classification methods," *Neural Netw*, vol. 19, pp. 1–9, 2005.

[46] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[47] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22. IBM New York, 2001, pp. 41–46.

[48] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1. Citeseer, 1998, pp. 41–48.

[49] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 129–136.

[50] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[51] L. Breiman, *Classification and regression trees*. New York: Routledge, 2017.

[52] ——, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[53] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[54] Multinomial distribution. Wolfram MathWorld. [Online]. Available: http://mathworld.wolfram.com/MultinomialDistribution.html

[55] B. A. Goldstein, A. M. Navar, M. J. Pencina, and J. Ioannidis, "Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review," *Journal of the American Medical Informatics Association*, vol. 24, no. 1, pp. 198–208, 2017.

[56] B. Shickel, P. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *arXiv preprint arXiv:1706.03446*, 2017.

[57] T. Tran, W. Luo, D. Phung, S. Gupta, S. Rana, R. L. Kennedy, A. Larkins, and S. Venkatesh, "A framework for feature extraction from hospital medical data with applications in risk prediction," *BMC bioinformatics*, vol. 15, no. 1, p. 425, 2014.

[58] Y. Wang, K. Ng, R. J. Byrd, J. Hu, S. Ebadollahi, Z. Daar, S. R. Steinhubl, W. F. Stewart *et al.*, "Early detection of heart failure with varying prediction windows by structured and unstructured data in electronic health records," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 2015, pp. 2530–2533.

[59] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: an unsupervised representation to predict the future of patients from

the electronic health records," *Scientific reports*, vol. 6, p. 26094, 2016.

[60] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[61] A. Avati, K. Jung, S. Harman, L. Downing, A. Ng, and N. H. Shah, "Improving palliative care with deep learning," in *Bioinformatics and Biomedicine (BIBM), 2017 IEEE International Conference on*. IEEE, 2017, pp. 311–316.

[62] J. Zhao, "Temporal weighting of clinical events in electronic health records for pharmacovigilance," in *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 375–381.

[63] Concept: Elixhauser comorbidity index. [Online]. Available: http://mchp-appserv.cpe.umanitoba.ca/viewConcept.php?printer=Y&conceptID=1436

[64] Term: Comorbidity / comorbidities. University of Manitoba. [Online]. Available: http://mchp-appserv.cpe.umanitoba.ca/viewDefinition.php?definitionID=102446

[65] P. A. McKee, W. P. Castelli, P. M. McNamara, and W. B. Kannel, "The natural history of congestive heart failure: the framingham study," *New England Journal of Medicine*, vol. 285, no. 26, pp. 1441–1446, 1971.

[66] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[67] (2014) Mini-stroke: What should you do? Harvard Health Publishing. [Online]. Available: https://www.health.harvard.edu/heart-health/mini-stroke-what-should-you-do

[68] Palliative care in cancer. National Cancer Institute. [Online]. Available: https://www.cancer.gov/about-cancer/advanced-cancer/care-choices/palliative-care-fact-sheet

[69] J. Zhao, A. Henriksson, L. Asker, and H. Boström, "Detecting adverse drug events with multiple representations of clinical measurements," in *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 536–543.

[70] R. Sukkar, E. Katz, Y. Zhang, D. Raunig, and B. T. Wyman, "Disease progression modeling using hidden markov models," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2012, pp. 2845–2848.

[71] R. A. Baxter, G. J. Williams, and H. He, "Feature selection for temporal health records," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2001, pp. 198–209.

[72] Definition of acute vs primary care. Alaska Nurse Practitioner Association. [Online]. Available: https://anpa.enpnetwork.com/nurse-practitioner-news/60531-definition-of-acute-vs-primary-care

[73] G. M. Weiss and F. Provost, "The effect of class distribution on classifier learning: an empirical study," *Rutgers Univ*, 2001.

[74] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97. Nashville, USA, 1997, pp. 179–186.

[75] WHOCC. Atc structure and principles. [Online]. Available: https://www.whocc.no/atc/structure_and_principles/

[76] R. A. Bifola. Machine learning(ml) — data preprocessing – techspecialist academy – medium. Techspecialist Academy. [Online]. Available: https://medium.com/@techspecialistacademy/machine-learning-ml-data-preprocessing-d968f86b703

[77] G. Press. Cleaning big data: Most time-consuming, least enjoyable data science task, survey says. Forbes. [Online]. Available: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says

[78] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.

[79] 4.2. feature extraction — scikit-learn 0.19.1 documentation. [Online]. Available: http://scikit-learn.org/stable/modules/feature_extraction.html

[80] J. Brownlee. What is the difference between a parameter and a hyperparameter? [Online]. Available: https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/

[81] Random forest. Scikit-learn. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[82] Multinomialnb. Scikit-learn. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

[83] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 49–57, 2010.

[84] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, 2011.

[85] Harmonic mean. Wolfram MathWorld. [Online]. Available: http://mathworld.wolfram.com/HarmonicMean.html

[86] Y. Sasaki *et al.*, "The truth of the f-measure," *Teach Tutor mater*, vol. 1, no. 5, pp. 1–5, 2007.

[87] V. Van Asch, "Macro-and micro-averaged evaluation measures [[basic draft]]," *Belgium: CLiPS*, 2013.

[88] K. M. Ting, *Confusion Matrix*. Boston, MA: Springer US, 2017, pp. 260–260. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_50

[89] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[90] Pointwise vs. pairwise vs. listwise learning to rank. [Online]. Available: https://medium.com/@nikhilbd/pointwise-vs-pairwise-vs-listwise-learning-to-rank-80a8fe8fadfd

Appendices

Appendix A

Data overview appendix

The following appendix lists all of the different columns contained by the distinct data views:

Column	Description
date_time	The date when the note was written down.
note	The note in free text in Swedish.
local_keyword	A keyword tag describing the note.
local_contact_type_name	The reason/context of the note (e.g. phone call, personal visit etc).
patient_notes_interpretation_as_string	A short interpretation (summary) of the note written down by the medical professional.
patient_notes_interpretation_as_float	If the patient note contains information about a particular analysis result (e.g. red blood cells measurement), the value is written here.
patient_notes_interpretation_as_float_2	If the patient note contains information about a particular analysis result which is noted down as two values (e.g. blood-pressure), then the second value is written here.

Table A.1: Patient notes view columns

Column	Description
analysis_name	The name of the analysis performed (in Swedish).
analysis_date	The date when the analysis was performed.
analysis_pathology	A flag indicating whether the value shown on the analysis is out of normal range.
local_range	The normal range for this particular analysis.
analysis_result_interpretation_as_float	The value of the analysis result.
analysis_result_interpretation_as_float2	If the analysis has a two-value result (e.g. blood-pressure), then the second value is written here.
analysis_result_interpretation_as_string	The analysis result noted down as a string (e.g. blood-pressure: "130/80").
analysis_search_word	The search tag for the analysis performed in English, depends on the system in use in the particular hospital.
analysis_bio_search_word	The general type of analysis performed (e.g. blood or urine test).

Table A.2: Analysis results view columns

Column	Description
diagnosis_code	The ICD-10 individual diagnosis code.
diagnosis_code_name_language_phrase	A short description of the diagnosis in Swedish.
diagnosis_group_name	The name of the high level ICD-10 group (1 out of 22) that this particular diagnosis belongs to.
diagnosis_type_id	A technical flag, unused.
diagnosis_text	A more extensive description of the diagnosis in Swedish.
main_diagnose	Unused.
date_time	The date when the diagnosis was assigned.

Table A.3: Diagnosis view columns

Column	Description
atc_code	The medication ATC code.
standing_prescription_start_date	The start date of the prescription.
standing_prescription_stop_date	The end date of the prescription.
atc_group_name	The ATC group name.
atc_group_code	The ATC group description.
atc_group_depth	The ATC hierarchy depth of the medication.
atc_group_id	The ATC group ID.

Table A.4: Standing prescription view columns

Column	Description
actual_sickleave_from_date	The start date of the sick leave.
actual_sickleave_to_date	The end date of the sick leave.
sickleave_percentage	The percentage of sick leave.
diagnosis_code	The diagnosis cause for the sick leave.
diagnosis_code_name_language_phrase	A short description of the diagnosis in Swedish.
diagnosis_group_name	The ICD-10 high level group code.
diagnosis_type_id	A technical flag, unused.

Table A.5: Sick leave view columns

Column	Description
product_code_name	The product code.
product_code_description	The product description in Swedish.
date_time	The date of the event.

Table A.6: Products view columns

Diagnosis group	Gender	Count
BLOD & IMM	F	799
	M	342
CIRK SJD	F	8089
	M	6406
ENDOKRIN	F	6660
	M	3428
GRAV SJD	F	248
	M	78
HUD SJD	F	3490
	M	2753
INFEKT SJD	F	1702
	M	1208
KONT M HSV	F	4321
	M	2503
MAG-TARM	F	2936
	M	1664
MEDF MISSB	F	104
	M	71
MUSK-SKEL	F	19280
	M	9458
NEUR SJD	F	2170
	M	779
PERIN SJD	F	1
	M	2
PSYK SJD	F	9824
	M	4000
RESP SJD	F	6119
	M	3349
SKAD & FÖRG	F	3537
	M	2243
SYMT & OKL	F	16182
	M	8700
TUMÖR SJD	F	1013
	M	952
URIN & KÖN	F	3181
	M	1517
Y ORS T SJD	F	1
	M	2
ÖGON SJD	F	871
	M	483
ÖRON SJD	F	2240
	M	1672

Table A.7: Diagnosis groups per gender

Appendix B

Results appendix

Class	Precision	Recall	F1-score	Support
BLOD & IMM	0.20	0.36	0.26	342
CIRK SJD	0.37	0.66	0.47	4349
ENDOKRIN	0.42	0.39	0.41	3026
GON SJD	0.81	0.38	0.52	406
GRAV SJD	1.00	0.06	0.12	98
HUD SJD	0.73	0.62	0.67	1873
INFEKT SJD	0.22	0.44	0.30	873
KONT M HSV	0.55	0.44	0.49	2047
MAG-TARM	0.44	0.30	0.36	1380
MEDF MISSB	0.04	0.04	0.04	52
MUSK-SKEL	0.71	0.71	0.71	8622
NEUR SJD	0.52	0.26	0.35	885
PERIN SJD	0.00	0.00	0.00	1
PSYK SJD	0.58	0.65	0.61	4147
RESP SJD	0.60	0.64	0.62	2840
RON SJD	0.60	0.66	0.63	1174
SKAD & FRG	0.66	0.62	0.64	1734
SYMT & OKL	0.59	0.32	0.42	7465
TUMR SJD	0.53	0.30	0.39	590
URIN & KN	0.39	0.51	0.44	1409
Y ORS T SJD	0.00	0.00	0.00	1
Micro-avg/total	0.57	0.54	0.53	43314

Table B.1: Single view learning results per class

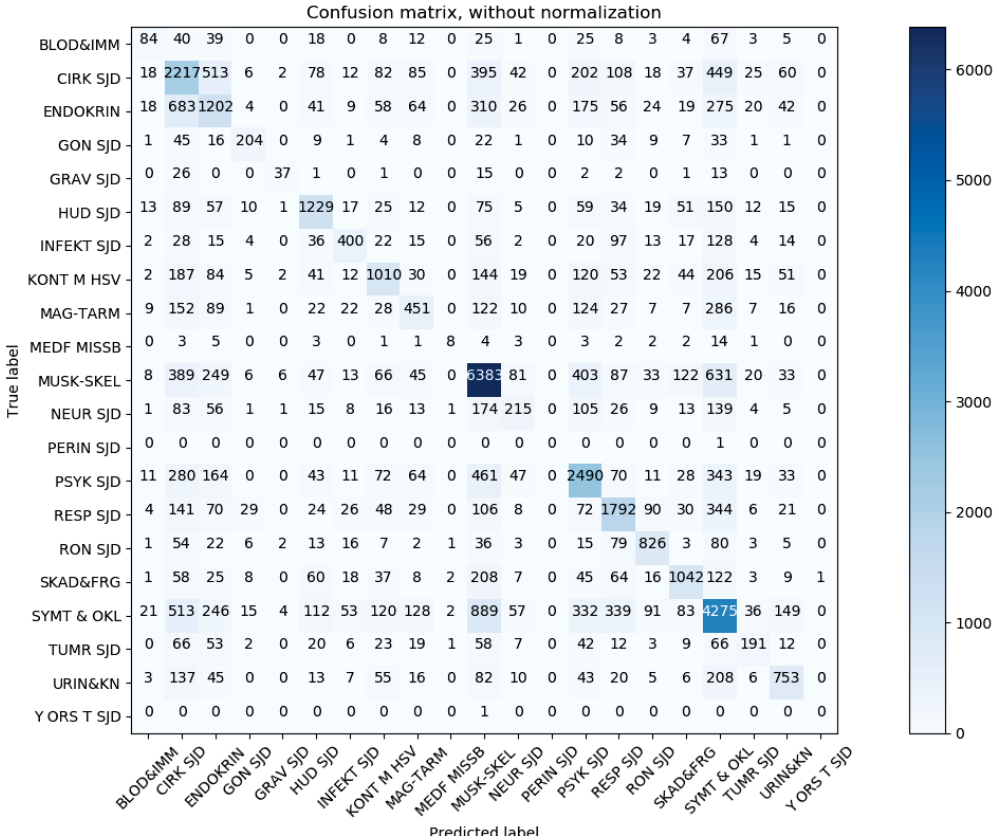

Figure B.1: Single view learning best model confusion matrix.

Class	Precision	Recall	F1-score	Support
BLOD & IMM	0.35	0.40	0.37	342
CIRK SJD	0.43	0.50	0.46	4349
ENDOKRIN	0.38	0.33	0.36	3026
GON SJD	0.70	0.41	0.52	406
GRAV SJD	0.58	0.26	0.35	98
HUD SJD	0.61	0.70	0.65	1873
INFEKT SJD	0.55	0.43	0.49	873
KONT M HSV	0.59	0.36	0.45	2047
MAG-TARM	0.46	0.29	0.35	1380
MEDF MISSB	0.55	0.35	0.42	52
MUSK-SKEL	0.66	0.71	0.69	8622
NEUR SJD	0.45	0.19	0.27	885
PERIN SJD	0.00	0.00	0.00	1
PSYK SJD	0.54	0.65	0.59	4147
RESP SJD	0.57	0.65	0.60	2840
RON SJD	0.65	0.77	0.71	1174
SKAD & FRG	0.64	0.61	0.62	1734
SYMT & OKL	0.52	0.48	0.50	7465
TUMR SJD	0.39	0.44	0.41	590
URIN & KN	0.53	0.53	0.53	1409
Y ORS T SJD	0.00	0.00	0.00	1
Micro-avg/total	0.55	0.55	0.54	43314

Table B.2: CCA classification with dimensionality reduction results per class

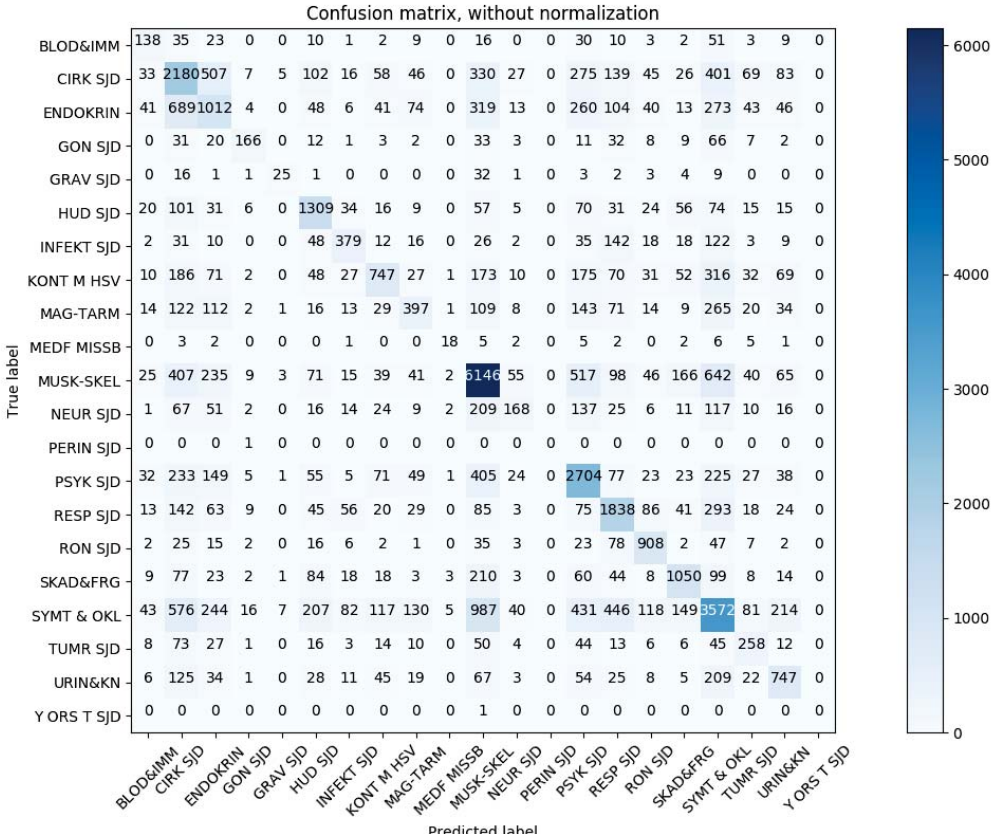

Figure B.2: CCA multi-view dimensionality reduction best model confusion matrix.

Class	Precision	Recall	F1-score	Support
BLOD & IMM	0.79	0.12	0.21	342
CIRK SJD	0.46	0.62	0.53	4349
ENDOKRIN	0.53	0.39	0.45	3026
GON SJD	0.91	0.43	0.59	406
GRAV SJD	1.00	0.08	0.15	98
HUD SJD	0.73	0.69	0.71	1873
INFEKT SJD	0.88	0.37	0.52	873
KONT M HSV	0.72	0.46	0.56	2047
MAG-TARM	0.71	0.25	0.37	1380
MEDF MISSB	0.00	0.00	0.00	52
MUSK-SKEL	0.70	0.80	0.75	8622
NEUR SJD	0.87	0.15	0.26	885
PERIN SJD	0.00	0.00	0.00	1
PSYK SJD	0.64	0.67	0.65	4147
RESP SJD	0.70	0.68	0.69	2840
RON SJD	0.82	0.70	0.75	1174
SKAD & FRG	0.81	0.60	0.69	1734
SYMT & OKL	0.48	0.68	0.56	7465
TUMR SJD	0.87	0.28	0.42	590
URIN & KN	0.76	0.51	0.61	1409
Y ORS T SJD	0.00	0.00	0.00	1
Micro-avg/total	0.64	0.61	0.60	43314

Table B.3: Multi-view stacked ensemble per data type best predictor per class results

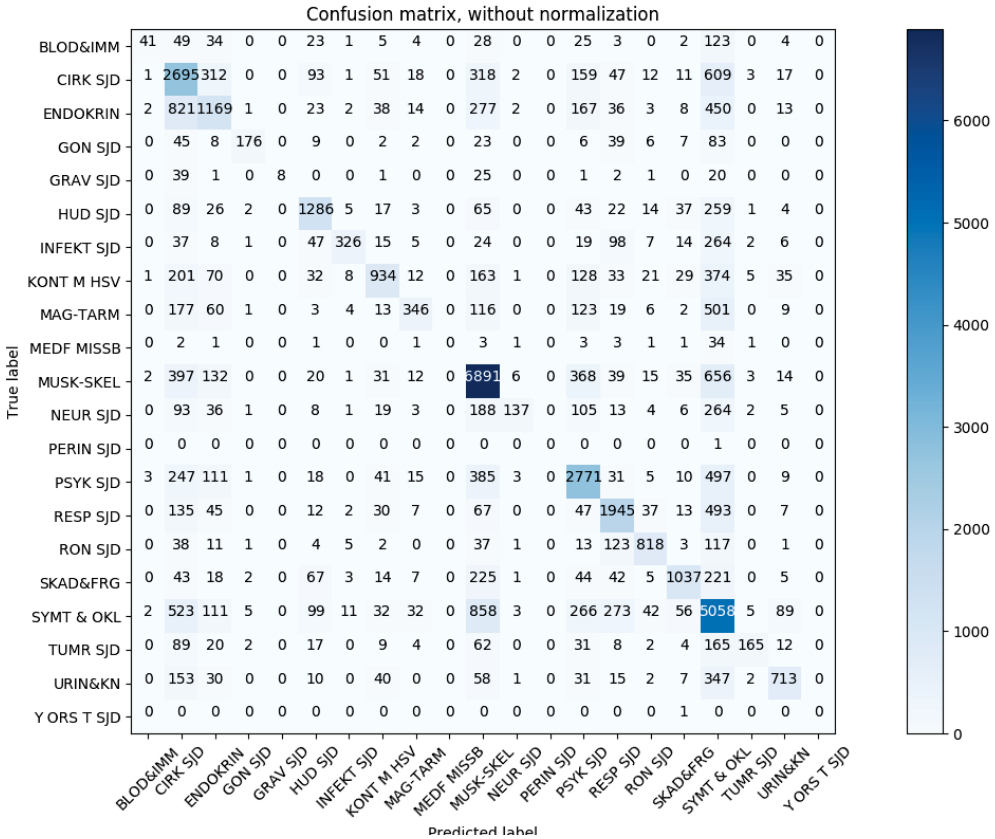

Figure B.3: Multi-view stacked ensemble per data type best model confusion matrix.

Class	Precision	Recall	F1-score	Support
BLOD & IMM	0.55	0.42	0.48	342
CIRK SJD	0.60	0.62	0.61	4349
ENDOKRIN	0.59	0.54	0.56	3026
GON SJD	0.71	0.66	0.68	406
GRAV SJD	0.77	0.73	0.75	98
HUD SJD	0.82	0.70	0.76	1873
INFEKT SJD	0.66	0.73	0.69	873
KONT M HSV	0.63	0.69	0.66	2047
MAG-TARM	0.59	0.47	0.52	1380
MEDF MISSB	0.60	0.40	0.48	52
MUSK-SKEL	0.77	0.81	0.79	8622
NEUR SJD	0.60	0.40	0.48	885
PERIN SJD	0.00	0.00	0.00	1
PSYK SJD	0.67	0.74	0.70	4147
RESP SJD	0.69	0.75	0.72	2840
RON SJD	0.73	0.82	0.77	1174
SKAD & FRG	0.72	0.78	0.75	1734
SYMT & OKL	0.67	0.66	0.67	7465
TUMR SJD	0.66	0.47	0.55	590
URIN & KN	0.69	0.63	0.66	1409
Y ORS T SJD	0.00	0.00	0.00	1
Micro-avg/total	0.68	0.69	0.68	43314

Table B.4: Multi-view stacked ensemble per data view best predictor per class results

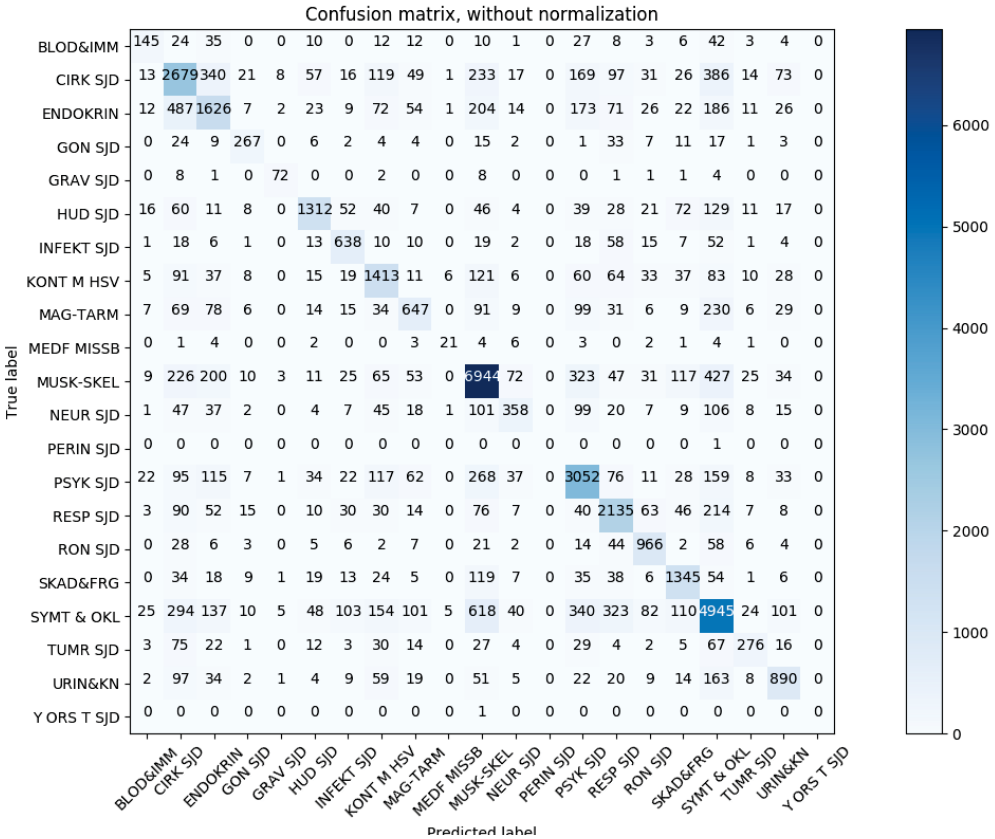

Figure B.4: Multi-view stacked ensemble per data view best model confusion matrix.

Appendix C

Pairwise ranking classification approach

Another classification approach that was attempted in the project is a pairwise ranking [49] classification approach based on the gradient boosting framework XGBoost [50]. The framework relies on the idea of gradient boosting trees [89] and is an ensemble learning method employing the idea of CART trees [51].

XGBoost has one interesting feature: optimizing the objective loss function based on a ranking pairwise approach. What this means is that instead of optimizing the loss function for one single data example, we use a pair of examples, and we optimize the loss function for each pair of these examples. The optimization is done for each separate pair in the dataset, so the search space for the hypothesis is much larger than the one we have in a single ranking (pointwise) approach [90].

We shall not go into details of the implementation of the XGBoost framework, however it is important to note that it exploits parallel learning extensively, depending on the memory capabilities of the system.

The XGBoost framework with a pairwise ranking approach in this project was applied in all of the same learning experiments and objectives, with significant results.

C.1 Single-view experiment

The single view approach with XGBoost had significantly better results compared to the models obtained using Random Forest and the MultinomialNB classifiers. The pairwise ranking loss function optimization along with the gradient boosting trees approach seems to significantly improve predictive performance of the less frequent classes in the dataset. This can be witnessed if we compare the F1 scores for the single-view Random Forest and the XGBoost approaches (Table C.2).

Classifier	XGBoost	
Score type	Test	CV
F1 mirco avg.	0.685	0.683

Table C.1: Single view learning results with the XGBoost framework

Class	Random forest F1-score	XGBoost F1-score
BLOD & IMM	0.31	0.46
CIRK SJD	0.46	0.62
ENDOKRIN	0.40	0.57
GON SJD	0.58	0.64
GRAV SJD	0.48	0.77
HUD SJD	0.66	0.74
INFEKT SJD	0.53	0.64
KONT M HSV	0.54	0.65
MAG-TARM	0.38	0.54
MEDF MISSB	0.24	0.39
MUSK-SKEL	0.70	0.79
NEUR SJD	0.30	0.47
PERIN SJD	0.00	0.00
PSYK SJD	0.59	0.70
RESP SJD	0.62	0.71
RON SJD	0.70	0.76
SKAD & FRG	0.64	0.72
SYMT & OKL	0.56	0.68
TUMR SJD	0.40	0.50
URIN & KN	0.57	0.67
Y ORS T SJD	0.00	0.00
Micro-avg/total	0.57	0.68

Table C.2: Single-view Random Forest vs. XGBoost per class classification scores

C.2 Multi-view experiments

Canonical Correlation Analysis

In the case of dimensionality reduction, it seems that the gradient boosting approach with pairwise ranking does not seem to make a lot difference when it comes to the final performance scores. In particular, we can notice an increase of approximately 0.05 F1 score increase on average of the final models (CCASU) compared to the structured (CCAS) models (table C.3).

Components	2		4		6		8		10	
Score	Test	CV	Test	CV	Test	CV	Test	CV	Test	CV
CCAS	0.358	0.360	0.506	0.502	0.557	0.554	0.591	0.586	0.588	0.589
CCAU	0.298	0.297	0.383	0.382	0.399	0.400	0.447	0.444	0.443	0.445
CCASU	0.369	0.366	0.505	0.507	0.558	0.556	**0.594**	0.591	0.590	0.592

Table C.3: CCA multi-view dimensionality reduction classification results with XGBoost

Multi-view stacked ensemble per data type

It seems that the pairwise ranking approach does improve the performance of the per-data type multi-view stacked ensemble experiment compared to the pointwise ranking. The best performance overall (a micro-F1 score of **0.684**) is achieved using a combination of the pairwise XGBoost library along with the MultinomialNB classifier and the XGBoost again as a final classifier. However, it is important to note that this improvement is marginal compared to the structured data model (PTS) trained (0.682 F1 for the PTS vs. 0.684 F1 for the PTSU using XGBoost). The different F1 scores obtained for the experiment can be observed in table C.4.

Multi-view stacked ensemble per data view

Same as we observed in the case of Random Forest and Multinomial Naive Bayes, here we can also see a maximum performance achievement of 0.686 only on the diagnosis view. However, in contrast to the pointwise approach, here we can notice that the final classifier (PVF)

Model				PTSU					
	Classifier			MultinomialNB		Random forest		XGBoost	
		Test	CV	Test	CV	Test	CV	Test	CV
Structured data (PTS)	MultinomialNB	0.484	0.485						
Unstructured data (PTU)	XGBoost	0.517	0.515	0.531	0.541	0.547	0.584	0.626	0.655
Structured data (PTS)	Random forest	0.573	0.588						
Unstructured data (PTU)	XGBoost	0.517	0.515	0.602	0.726	0.565	0.680	0.566	0.764
Structured data (PTS)	XGBoost	0.682	0.679						
Unstructured data (PTU)	XGBoost	0.517	0.515	0.675	0.695	0.593	0.622	0.680	0.706
Structured data (PTS)	XGBoost	0.682	0.679						
Unstructured data (PTU)	MultinomialNB	0.508	0.503	0.674	0.685	0.593	0.617	**0.684**	0.702
Structured data (PTS)	XGBoost	0.682	0.679						
Unstructured data (PTU)	Random forest	0.462	0.466	0.617	0.757	0.550	0.683	0.574	0.765

Table C.4: Multi-view stacked ensemble per data type results

F1 scores do not differ to a large extent from the diagnosis view classifier (e.g. 0.684 for the PVF model vs 0.686 for the PVD model). The different performance scores obtained are presented in table C.5.

Classifier	XGBoost	
Score	Test	CV
Date trends (PVDT)	0.233	0.233
Diagnosis view (PVD)	**0.686**	0.682
Analysis view (PVA)	0.297	0.296
Prescriptions view (PVP)	0.295	0.293
Sick leave view (PVS)	0.207	0.207
Products view (PVPR)	0.210	0.208
Notes view (PVN)	0.522	0.513

Classifier	MultinomialNB		Random forest		XGBoost	
Score	Test	CV	Test	CV	Test	CV
Final classifier (PVF)	0.677	0.692	0.595	0.616	0.684	0.701

Table C.5: Multi-view stacked ensemble per view results

TRITA EECS-EX-2018:587