THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *10/11/2017* **par :**

Ghazar CHAHBANDARIAN

Elicitation of Relevant Information From Medical Databases: Application to the Encoding of Secondary Diagnoses

---

JURY

Christine VERDIER	Professeur des Universités	Rapporteur
Sandra BRINGAY	Professeur des Universités	Rapporteur
Rémi BASTIDE	Professeur des Universités	Directeur de thèse
Nathalie BRICON-SOUF	Maître de Conférences	Co-encadrant
Olivier TESTE	Professeur des Universités	Invité
Jean-Christophe STEINBACH	Maître de Conférences	Invité

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle

Unité de Recherche :
Institut de Recherche en Informatique (UMR 5505)

Directeur(s) de Thèse :
Rémi BASTIDE et *Nathalie BRICON-SOUF*

Rapporteurs :
Christine VERDIER et *Sandra BRINGAY*

Acknowledgements

First, I would like to thank Midi-Pyrénées region, Castres-Mazamet Technopole, INU Champollion and Paul Sabatier University for funding my PhD thesis. Without such a grant this work would not have been realized.

I would like to express my gratitude to my thesis directors Prof. Rémi Bastide for his guidance and recommendations in the course of the thesis. My deepest gratitude to my co-directer Dr. Nathalie Bricon-Souf for her patience, advices, enthusiasm, comments, follow-ups and attentive corrections throughout my thesis. I have been extremely lucky to have such supervisors, they are great source of inspiration.

I would also thank the hospital of "*Centre Hospitalier Intercommunal de Castres Mazamet*" for accepting the collaboration with their professionals. I would particularly like to acknowledge all the members of the DIM "Department of Medical Information" Isabelle, Véronique, Jérôme, Pauline, Virginie, Karine, Hélène, Laurence, Béatrix and Valérie for the effort and the time devoted explaining all the details of their job. I am particularly thankful to Dr. Jean-Christophe Steinbach, the head of DIM, for for his collaboration and for all the time devoted to achieve my work.

My deep gratitude is also addressed to the members of the jury Prof. Chrsitine Verdier, Prof. Sandra Bringay for reviewing my thesis and providing valuable comments; Prof. Olivier Teste for accepting being part of the PhD committee members.

I am thankful to the engineering school "ISIS" for having me in their offices. I would like to thank all those with whom I have exchanged and discussed about my thesis and about computer science in general including Dr. Nicolas Singer, Dr. Elyes Lamine, Dr. Imen Megdiche, Dr. Adrien Deffosez, Prof. Hérvé Pingaud, Dr. Emmanuel Conchon, Dr. Rejane Dalce and Prof. Bernard Rigaud. I would also like to thank all my colleagues Samuel, Florence, Eric, Véronique, Laurent, Lauriane, Alexandre and Emmanuelle for the enjoyable moments spent in the engineering school. Special thanks go to Bruno Roussel and Francis Faux for the enjoyable discussions and ping-pong pauses we have made during my stay in the connected health lab.

I want to thank all my friends in Castres Amal, Fiona, Anne, Marine, Kévin and Jing. My freinds from the "Amicale des Arméniens" Gérard, Nicolas, Lussiné, Stephane, André, Lussiné, Jeanna, Armen, Avedis, Aroutine, Guevorg, Maria, Tatev, Romain, Sasha,

Anna, Ani, Armine and Lilith. My friends from the "Amis du Levant" Anthony, Laura, Rana, Farid, Mouin and Dani. Thank you all, you are my family in France. I would also thank my friends from all over the world Jocelyn, Sami, Houssam, Alek, Sevag, Houssam, Rami and Ammar for keeping contact and encouraging me all the time. Special thanks go to Rita Zgheib for being a great friend and an excellent colleague, for sharing the office, for the support and for the fruitful discussions all along the PhD.

It is with great emotion that I dedicate the entire work to my lovely parents, my brother and my sister. You are my source of strength that keeps me up. Without your advices and support, I would never reach this grade and get the diploma.

Abstract

In the thesis we focus on encoding inpatient episode into standard codes, a highly sensitive medical task in French hospitals, requiring minute detail and accuracy, since the hospital's income directly depends on it. Encoding inpatient episode includes encoding the primary diagnosis that motivates the hospitalisation stay and other secondary diagnoses that occur during the stay. Unlike primary diagnosis, encoding secondary diagnoses is prone to human error, due to the difficulty of collecting relevant data from different medical sources, or to the outright absence of relevant data that helps encoding the diagnosis.

We propose a retrospective analysis on the encoding task of some selected secondary diagnoses. Hence, the PMSI[1] database is analysed in order to extract, from previously encoded inpatient episodes, the decisive features to encode a difficult secondary diagnosis occurred with frequent primary diagnosis. Consequently, at the end of an encoding session, once all the features are available, we propose to help the coders by proposing a list of relevant encodings as well as the features used to predict these encodings. Nonetheless, a set of challenges need to be addressed for the development of an efficient encoding help system. The challenges include, an expert knowledge in the medical domain and an efficient exploitation methodology of the medical database by Machine Learning methods.

With respect to the medical domain knowledge challenge, we collaborate with expert coders in a local hospital in order to provide expert insight on some difficult secondary diagnoses to encode and in order to evaluate the results of the proposed methodology.

With respect to the medical databases exploitation challenge, we use ML methods such as Feature Selection (FS), focusing on resolving several issues such as the incompatible format of the medical databases, the excessive number features of the medical databases in addition to the unstable features extracted from the medical databases.

Regarding to issue of the incompatible format of the medical databases caused by relational databases, we propose a series of transformation in order to make the database and its features more exploitable by any FS methods.

To limit the effect of the excessive number of features in the medical database, usually motivated by the amount of the diagnoses and the medical procedures, we propose to group the excessive number features into a proper representation level and to study the best representation level.

Regarding to issue of unstable features extracted from medical databases, as the dataset linked with diagnoses are highly imbalanced due to classification categories that are unequally represented, most existing FS methods tend not to perform well on them even if sampling strategies

[1] *Programme de Médicalisation des Systèmes d'Information*, a huge database that documents all the inpatient episodes information across France.

are used. We propose a methodology to extract stable features by sampling the dataset multiple times and extracting the relevant features from each sampled dataset.

Thus, we propose a methodology that resolves these issues and extracts stable set of features from medical database regardless to the sampling method and the FS method used in the methodology.

Lastly, we evaluate the methodology by building a classification model that predicts the studied diagnoses out of the extracted features. The performance of the classification model indicates the quality of the extracted features, since good quality features produces good classification model. Two scales of PMSI database are used: local and regional scales. The classification model is built using the local scale of PMSI and tested out using both local and regional scales.

Hence, we propose applying our methodology to increase the integrity of the encoded diagnoses and to prevent missing important encodings. We propose modifying the encoding process and providing the coders with the potential encodings of the secondary diagnoses as well as the features that lead to this encoding.

Résumé

Dans cette thèse, nous nous concentrons sur le codage du séjour d'hospitalisation en codes standards. Ce codage est une tâche médicale hautement sensible dans les hôpitaux français, nécessitant des détails minutieux et une haute précision, car le revenu de l'hôpital en dépend directement. L'encodage du séjour d'hospitalisation comprend l'encodage du diagnostic principal qui motive le séjour d'hospitalisation et d'autres diagnostics secondaires qui surviennent pendant le séjour.

Nous proposons une analyse rétrospective mettant en oeuvre des méthodes d'apprentissage, sur la tâche d'encodage de certains diagnostics secondaires sélectionnés. Par conséquent, la base de données PMSI[2] est analysée afin d'extraire à partir de séjours de patients hospitalisés antérieurement, des variables décisives (Features). Identifier ces variables permet de pronostiquer le codage d'un diagnostic secondaire difficile qui a eu lieu avec un diagnostic principal fréquent. Ainsi, à la fin d'une session de codage, nous proposons une aide pour les codeurs en proposant une liste des encodages pertinents ainsi que des variables utilisées pour prédire ces encodages. Les défis nécessitent une connaissance métier dans le domaine médical et une méthodologie d'exploitation efficace de la base de données médicales par les méthodes d'apprentissage automatique.

En ce qui concerne le défi lié à la connaissance du domaine médical, nous collaborons avec des codeurs experts dans un hôpital local afin de fournir un aperçu expert sur certains diagnostics secondaires difficiles à coder et afin d'évaluer les résultats de la méthodologie proposée.

En ce qui concerne le défi lié à l'exploitation des bases de données médicales par des méthodes d'apprentissage automatique, plus spécifiquement par des méthodes de "Feature Selection" (FS), nous nous concentrons sur la résolution de certains points : le format des bases de données médicales, le nombre de variables dans les bases de données médicales et les variables instables extraites des bases de données médicales.

Nous proposons une série de transformations afin de rendre le format de la base de données médicales, en général sous forme de bases de données relationnelles, exploitable par toutes les méthodes de type FS.

Pour limiter l'explosion du nombre de variables représentées dans la base de données médicales, généralement motivée par la quantité de diagnostics et d'actes médicaux, nous analysons l'impact d'un regroupement de ces variables dans un niveau de représentation approprié et nous choisissons le meilleur niveau de représentation.

Enfin, les bases de données médicales sont souvent déséquilibrées à cause de la répartition inégale des exemples positifs et négatifs. Cette répartition inégale cause des instabilités de variables extraites par des méthodes de FS. Pour résoudre ce problème, nous proposons une

[2] *Programme de Médicalisation des Systèmes d'Information*, une grande base de données médicales qui documente toutes les informations sur les séjours d'hospitalisation en France.

méthodologie d'extraction des variables stables en échantillonnant plusieurs fois l'ensemble de données et en extrayant les variables pertinentes de chaque ensemble de données échantillonné.

Nous évaluons la méthodologie en établissant un modèle de classification qui prédit les diagnostics étudiés à partir des variables extraites. La performance du modèle de classification indique la qualité des variables extraites, car les variables de bonne qualité produisent un bon modèle de classification. Deux échelles de base de données PMSI sont utilisées: échelle locale et régionale. Le modèle de classification est construit en utilisant l'échelle locale de PMSI et testé en utilisant des échelles locales et régionales.

Les évaluations ont montré que les variables extraites sont de bonnes variables pour coder des diagnostics secondaires. Par conséquent, nous proposons d'appliquer notre méthodologie pour éviter de manquer des encodages importants qui affectent le budget de l'hôpital en fournissant aux codeurs les encodages potentiels des diagnostics secondaires ainsi que les variables qui conduisent à ce codage.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

"There is no greater harm than that of time wasted."

-Michel Angelo

Contents

1.1 Background and motivation

We are living in an age where computer applications are easy to use, intuitive and provide useful information. However, some medical domains suffer from specific challenges that do not exist in other areas. For example, in some areas a lot of medical sources are available and it is difficult to decide which piece of medical information is useful and which piece of information leads to an efficient decision without specialist experience. Moreover, medical data is unique, especially when Machine Learning algorithms are applied on it (J.Cios and Moore, 2002). Therefore, it is not easy to provide an application based on medical data that facilitates the work of the specialist.

This thesis focuses on a difficult medical task of encoding diagnoses in the inpatient episodes of the hospitals. A lot of hospitals hire specialist and trained coders in order to encode properly all the diagnoses. Coding diagnosis involves reading and analysing all the information in the medical record such as discharge letters, reports and radio images of an inpatient episode and extract all the diagnosis, classify them into primary, secondary and related diagnoses. Then, coders look up diagnoses codes in the dictionary and finally register them in the hospital's local database. Finally, each month, the hospitals send all the codes to the national health agencies to provide the hospitals with fair payment according to their encoded activity.

The national health agency stores all the received inpatient information in a database called PMSI (*Programme de Médicalisation des Systèmes d'Information*) database in France. Over years, billions of medical records have been recorded in this national databases.

A lot of scientific challenges exist to analyse the PMSI database in order to support the task of encoding diagnoses. Most of the medical databases suffer mainly from imbalanced distribution of examples. One of the core technical issues we treat in the dissertation is related to applying Feature Selection methods on imbalanced medical databases such as PMSI. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly influence the performance of standard learning algorithms because most of the standard algorithms assume or expect balanced class distributions.

We propose on one hand a solution to enhance the encoding of secondary diagnoses on the other hand we address the challenges related to the exploitation of imbalanced medical databases using ML methods.

1.2 Overview of the research

In the dissertation we are interested in the medical domain, more particularly we focus on the encoding of secondary diagnoses in the context of PMSI. Encoding diagnoses depends heavily on human effort. Moreover, encoding all the diagnoses of a patient's medical record is a difficult task.

Actually, there are few methods to increase the quality and the integrity of the encoded diagnoses. Moreover, there are few methods that benefit from the immense information available in the PMSI database using previously encoded diagnoses in order to increase the quality of the encodings.

The current research of this thesis primarily addresses the encoding secondary diagnoses and tackles the drawback in the existing studies of diagnoses predication approaches and the related technical challenges.

1.3 Research objectives

Our proposed study addresses challenges related to the Machine Learning methods, more specifically on the feature selection methods in order to explore medical databases and provide adapted help to the specialist working in a specific task, such as encoding diagnoses.

In order to reach the aim of this research, the following objectives are addressed:

- Explore medical databases and prepare them for Machine Learning methods.
- Improve the sampling methods for imbalanced databases.
- Provide an efficient approach to select features from imbalanced medical databases.
- Insure the quality and stability of the selected features from imbalanced databases.

- Determine what, when and how to help the coders encode diagnoses.

1.4 Contribution of the thesis

The major contributions of this thesis are stated as follows:

- We presented a comprehensive study of the approaches to Feature Selection and evaluation methods.
- We provided a better understanding of how to select features from imbalanced medical databases.
- The study clearly showed how the existing and most currently used models are inadequate and not sufficient to give stable features from imbalanced datasets.
- We proposed a generic method to prepare medical databases to be used by Machine Learning methods.
- We developed an approach to select stable features from imbalanced dataset.
- We proposed a method to provide values to the selected stable features.
- We proposed a use case to integrate our approach in the thought process of the coders to spot missing diagnoses encoding of inpatient episodes.

1.5 Thesis roadmap

The reminder of the thesis is organised in two parts as follows.

The first part presents the state of the art and it consists of three chapters which present the work context, the used technologies and the application domain.

- The second chapter presents the medical databases in general and the PMSI database in particular. It presents encoding medical information in the medical database PMSI as well as the challenges related to it. We emphasise especially on the procedure followed in the hospitals to encode diagnoses which is the main application domain of the dissertation.

- The third chapter explores the Machine Learning methods more particularly in the Feature Selection methods and the evaluation approaches of such methods. This chapter introduces the technical challenges related to the usage of the Machine Learning methods with medical databases as well as the latest researches to overcome this challenges.
- The fourth chapter introduces the main studies in the scientific literature to encode diagnoses in the hospital. This chapter is particularly interested in providing details on different approaches followed to encode diagnoses.

The second part of the thesis consists of three chapters dedicated to our contribution to overcome the technical challenges as well as the contributions to provide adapted use case to encode secondary diagnoses.

- The fifth chapter focuses on our experience in the real observation sessions on the encoding diagnoses and propose a use case to our contribution in the hospital.
- The sixth chapter aims to present our approach to prepare the PMSI database to the usage of learning algorithms more specifically to the usage of Feature Selection methods. The proposed approach addresses the challenges related to the data selection, data transformation, feature preprocessing and finally the imbalanced datasets. We explain our approach to evaluate the preparation phase. Finally, we implement and evaluate our approach on the PMSI database issued from a local hospital.
- The seventh chapter is dedicated to study the influence of imbalanced dataset on the stability of the selected features and present our approach to select stable feature from imbalanced datasets. Additionally, we discuss the application of the approach in providing aid in the encoding of diagnoses.

Finally, we end the dissertation with a conclusion and future research perspectives to our contribution.

Part I

State of the art

Chapter 2

PMSI, the French national medical database

"Genius is patience."

-Isaac Newton

Contents

2.1 Introduction

Thanks to the digitalization technologies in the healthcare domain and their rapid evolution, a major part of the paper medical records are transformed into electronic ones. Moreover, the new medical records are digital by default. Therefore, a lot of medical databases have been developed. There are a lot of benefits of using medical databases from which we mention the easy access to medical information, the security, since they can be backed up in a secure place and of course the ability to capture and store a big amount of data in order to use it to provide better care. However, the main drawback of the medical databases is that the stored medical data is sensitive and private, therefore anonymisation processes are necessary to insure the confidentiality of the patients private information. Other complications are the heterogeneity of data from different sources and other ethical issues which make medical data unique in these respects. (J.Cios and Moore, 2002).

There are a lot of databases in the medical domain, the most notable ones being Electronic Medical Record (EMR) databases, lab results, financial and administrative databases. Some of them are of small scale, mostly used in a local hospital. Others can get into very huge scale, such as a national or even international scale.

In the dissertation we use the PMSI (*Programme de Médicalisation des Systèmes d'Information*) database. The PMSI is a French medical database that is available in different scales. We detail in this chapter all the related information of the PMSI database, such as the history, the data generation and the data storage of the PMSI.

2.2 The PMSI history

The PMSI (*Programme de Médicalisation des Systèmes d'Information*) project was inspired from a model used in the United States called DRG (*Diagnosis Related Groups*), this project was an empirical construction of hospitalization costs based on several million hospital inpatient episodes. The collected data were classified into groups with similar medical cases and similar cost. Around 500 DRG groups were created with medical and cost homogeneity. The United States had set up the financing of certain inpatient episodes as early as 1983, with the DRG system (Fetter, 1991).

The beginning of PMSI in France dated back to 1985 when a project called "*Médicalisation des Systèmes d'Information*" took place. In 1989, the first Medical Information Department or the so-called DIM was created (Kohler, 2006). The period between 1989 and 1994 was generalisation period of PMSI to be used in the public hospital and some private hospitals. In 1995, the PMSI was used by all private hospitals. In the period between 1995 and 2004, the financial usage of PMSI took place with a global budget modulation aimed essentially to reduce the inequalities between private and public hospitals.

Finally, starting from 2004, France used the information of PMSI to introduce an activity based payment called in French T2A (*la tarification à l'activité*) to finance all acute hospital cares. The T2A activity based payment is a method of financing French healthcare institutions, it aims balancing the allocation of financial resources and aims to empower health actors. Each month public health institutions send their information to ARS *Agence Régionale de la Santé* which triggers the payment. The private health institutions send their information to the medical insurance to get paid accordingly (ATIH, 2016a).

Since the PMSI creation, millions of inpatient episodes have been registered in national database, which makes it an attractive target for data analysis to solve different problems using data mining techniques (Busse et al., 2011).

2.3 The PMSI versions

The PMSI (Programme de Médicalisation des Systèmes d'Information) has multiple versions to describe different activities.

- **MCO**: (*Médecine Chirurgie Obstétrique*) which stands for medicine, surgery and obstetrics. It is based on systematic collection each month of administrative and medical information, which is standardised form of inpatient episode. It has T2A implemented for activity based funding. (ATIH, 2016a)
- **HAD**: (*Hospitalisation à domicile*) which is based on systematic collection of administrative and medical information of home hospitalizations. It has T2A implemented for activity based funding. (ATIH, 2016b)
- **SSR**: (*Soins de Suite et de Réadaptation*) which is based on systematic collection of aftercare and rehabilitation information. Unlike MCO whose information concerns

an inpatient episode, in SSR version of PMSI, it concerns one week of patient care and rehabilitation. It does not have T2A system yet. (ATIH, 2016d)

- **PSY**: (*Psychiatrie*) which concerns systematic collecting of psychiatry episodes information. It does not have T2A system yet. (ATIH, 2016c)

The PMSI database can exist on three scales, all the scales have similar structures.

1. **Local scale**: each hospital has its own PMSI database.
2. **Regional scale**: this scale contains PMSI information from all hospitals of a particular region.
3. **National scale**: this scale contains PMSI information from all hospitals of France.

The ATIH organisation can distribute regional and national versions of PMSI after an agreement from the National Commission of Informatics and Liberties CNIL[1] organisation.

In the dissertation we are interested in PMSI-MCO because it describes the inpatient episodes, we refer to it as PMSI. We use local and regional scales of the PMSI databases, since they are available through collaboration of the "*Centre Hospitalier Intercommunal de Castres Mazamet*" hospital.

2.4 The PMSI content

The PMSI-MCO contains essentially inpatient episode information (ATIH, 2016a).

[1] https://www.cnil.fr/

Three steps are followed to produce PMSI information as show in the Figure 2.1.

1. Each patient in inpatient episode receives care from one or several medical units. These medical units produce reports that describe patient's state and describe all the diagnoses and the received care. The first step consists in encoding all the healthcare that the patient received in each medical unit. The encoded information is called RUM (Résumé d'Unité Médicale) medical unit summary. One RUM is made for each medical unit. The RUM contains information, such as gender, age and length of stay, diagnoses and medical procedures performed during the medical unit care.

   For example, three medical reports are produced when a patient enters a hospital's emergency medical unit and receives care from the intensive medical care unit and then receives care from the cardiology medical unit and finally is discharged to home. The three medical units are (Emergency - intensive care - cardiology) medical units. The three reports describe all the given care to the patient in each medical unit. Afterwards, these reports are encoded to produce three RUMs (Figure 2.1).

2. The second step consists in combining all the RUM reports into one report called standard episode summary or RSS (Résumé de Sortie Standardisé). The contents of RUM reports are combined into one RSS using special algorithms. The RSS is eventually classified within one of the existing GHM (Groupes Homogènes de Malades) groupings. A GHM grouping contains similar diagnoses and facilitates the management of the inpatient episode.

3. The third step, an anonymisation process is applied, thus producing a so-called anonymised episode summary RSA (Résumé de Sortie Anonymisé). In the anonymisation process, information, such as name and identification are removed. Birthdate information is processed to become age; entry date and discharge dates are processed to become length of stay, etc. Finally, the RSA reports are sent to the Regional Health Agencies ARS (Agences Régionales de Santé) where they are stored in the national PMSI database. Each hospital is eventually refunded according to the GHS (Groupe Homogène de Séjours) grouping of the RSA. The GHS grouping contains similar inpatient episodes with similar costs.

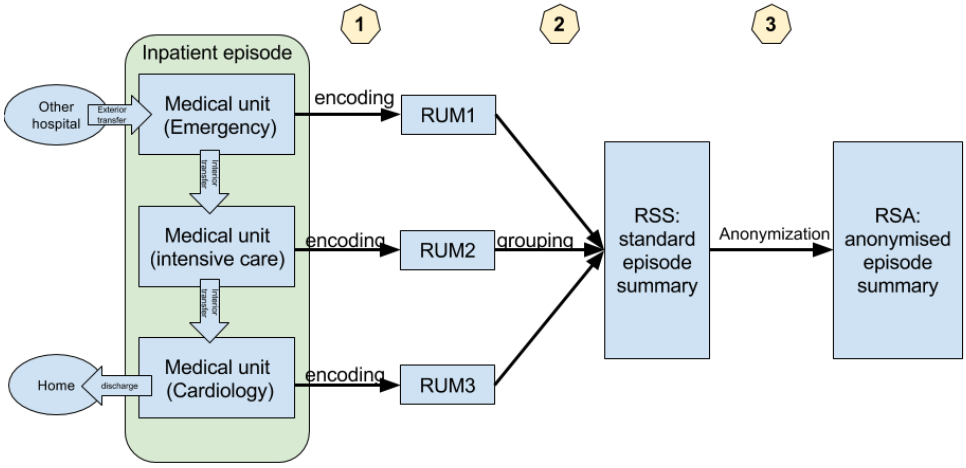


Fig. 2.1 The PMSI information workflow

The RUM is the core component of the PMSI, containing two categories of information as classified by ATIH[2].

1. *Administrative information*
    - Identification of the inpatient episode.
    - Identification of the establishment.
    - Gender.
    - Birth date.
    - Residency zip code .
    - Admission.
        - Admission date.
        - Admission mode (home, emergency, etc.,)
        - Provenance (from psychiatry or from home hospitalisation etc.,).
    - Discharge.
        - Discharge date.
        - Discharge mode (transfer, death, etc.,).
        - Destination (to aftercare and rehabilitation or to psychiatry).
    - Length of stay.
    - Session's count.

[2] http://www.atih.sante.fr/mco/presentation

2. *Medical information*
    - Diagnoses.
        - Main Morbidity.
            - Primary Diagnosis (DP).
            - Related diagnosis (DR).
        - Secondary Diagnoses (DS).
            - Significant secondary diagnoses.
            - Secondary diagnoses by convention.
    - Medical procedures.
    - Other information, such as weight at entry into the medical unit for the newborn.
    - Documentary information.

The administrative information consists of information that describes the inpatient episode. The identification numbers are encoded with special format to identify the inpatient episode, personal information, such as birthdate is encoded as dd-mm-yyyy and gender is encoded as (1 for male and 2 for female). Admission and discharge information is encoded using standard numbers. For example, an admission code of (6-1) indicates that the patient is transferred from another MCO medical unit. The first part of the code "6" indicates the admission mode, in the example "6" indicates "transfer". The second part of the code indicates where the patient is coming from, in the example "1" indicates from another MCO medical unit (ATIH, 2016a).

The administrative information is easy to encode, whereas the medical information uses standard codes that are difficult to choose, especially when encoding diagnoses. Therefore, most of the hospitals hire special coders in order to encode them properly.

2.5 Encoding medical information

2.5.1 Diagnoses encoding

To encode diagnoses, the *International Classification of Disease*[3] (ICD-10) is used. In France, the French version of ICD-10 is named CIM-10[4] (Classification Internationale des Maladies). There are different versions of CIM, France started to use the 10th version of CIM in 2006, but CIM-10 was designed in 1992. Some countries still use the 9th version of CIM. The CIM-10 has hierarchical classification: the first levels of hierarchy consist in chapters gathering same characteristic diseases (such as chapter II dedicated to tumoral diseases), categories help refining this classification. Currently, about 2,049 categories are commonly used for coding. The last level precisely describes each disease and the CIM-10 contains 33,816 codes in which the first three characters stand for code categories.

For example, "J96.101" is a CIM-10 code which is composed of two parts separated by a point. The first part consists of three characters and it designates the category of the diagnosis and can stand alone as a code. The second part is an option to add more specificities to the diagnosis. In the example, the letter "J" designates that the diagnosis is related to the respiratory system diseases, "J" used in conjunction with the numerals "9" and "6" indicates that the diagnosis falls into the category of "Respiratory failure diseases". The second part, the characters after the decimal point, is used to add more precision on the diagnosis. For example the "101" indicates that the respiratory failure is chronic type 1 restrictive (Hypoxia).

There are two kinds of diagnoses:

1. **Main morbidity** it consists of the primary diagnosis, supplemented if necessary by the related diagnoses.

   (a) *Primary diagnosis (DP)* is the main diagnosis of the RUM, it is the health problem that motivated the patient's admission to the medical unit, it is confirmed when the patient is discharged.

[3] http://www.who.int/classifications/icd/
[4] http://www.atih.sante.fr/mco/presentation

(b) *Related diagnoses (DR)* are complementary diagnoses used when DP is not sufficient to admit the patient. These diagnoses can also represent the chronic diseases that affect the DP.

2. **Secondary diagnoses (DS)** there are two kinds of secondary diagnoses: significant secondary diagnoses and secondary diagnoses by convention.

   (a) *Significant secondary diagnoses* are ailments, symptoms or any other reason that requires a healthcare alongside with the DP, such as additional health problems or complications of the DP, or complications of the treatment of the main morbidity. These diagnoses are coded at the end of the inpatient episode using all the knowledge acquired during the inpatient episode including the reports arrived after the discharge of the patient. A secondary diagnosis is significant when it requires additional care or management, such as (medical procedure, diagnostic consultations, etc.).

   However, there is a big difference between significant DS and DR, the former corresponds to additional health problem or an ailment in addition to the DP or complication to it or complication of a treatment to DP, whereas DR is a precision and an essential part of the main morbidity.

   (b) *Secondary diagnoses by convention* are all other diagnoses that do not satisfy the previous definition. For example, diagnoses that have external causes of morbidity, or infection, or complications of medical procedures.

It is essential that the RUM describes the inpatient episode as accurately as possible, without forgetting any diagnosis specially DS, because each inpatient episode is classified into a severity level. Each level of severity has different range of funding, it is important to classify the inpatient episodes into the right class in order to get fair payment. The ensemble of the diagnoses plays big role to define which class of severity is the inpatient episode in, forgetting or adding one DS can change the severity level of the inpatient episode. It is also important to have good quality PMSI database in order to be analysed properly.

In the dissertation we focus on the **P**rimary **D**iagnoses and they are referred as **DP**, we also focus on **S**econdary **D**iagnoses they are referred as **DS**.

2.5.2 Medical procedures encoding

Medical procedures are encoded in France using the Common Classification of Medical Procedures CCAM (*Classification Commune des Actes Médicaux*)[5]. The CCAM was developed between 1996 and 2001. The first version was published in 2002. It is updated frequently, the 39th version was released in 2015. The CCAM has a hierarchical classification: the first level of hierarchy consists of 19 general chapters which organise the medical procedures according to anatomical or functional structure. For instance, chapter 1 is for nervous system procedures and chapter 2 is for eye procedures. The second level of hierarchy separates the diagnoses and therapeutic medical procedures, it is possibly followed by one or more sub-levels.

The CCAM codes are defined with seven characters (four letters and three numbers). There are around 7,583 standard medical procedure codes. The first letter indicates the system or the anatomical device. The second letter provides additional information of the organ or the function of the first letter. The third letter designates the performed action. The fourth letter assigns the access mode. Finally the three numbers are used to differentiate the procedures that have the same four letters.

For example, HEQE003 is CCAM code, the first letter H alone indicates digestive system procedures, the first two letters are HE, they relate to more specific organ procedure: oesophagus. The third letter Q indicates the performed action which is examination. The fourth letter E indicates the access mode which is *Transorifice endoscopic access*. Its location in the hierarchical tree is 7.1.9.1. and it costs 100,45€.

In the dissertation we concentrate more on the first level of the tree which contains 19 general chapters structured according to anatomical or functional structure. The chapters are considered enough to help encoding diagnoses. Table 2.1 shows the 19 chapters of CCAM.

2.5.3 Documentary information

Finally, the RUM contains optional documentary information that could be anything, such as digit, code of procedure or diagnosis, or any free text. Encoding documentary

[5] http://www.atih.sante.fr/version-39-de-la-ccam

Table 2.1 CCAM chapters

Chapter	Label
01	Central nervous system, device and independent
02	Eye and notes
03	Ear
04	Circulatory
05	Immune system and hematopoietic
06	Respiratory
07	Digestive
08	Urinary and genital
09	Acts on the reproductive, pregnancy and the newborn
10	Endocrine and metabolic
11	Osteoarticular apparatus and muscle of the head
12	Osteoarticular apparatus and muscle neck and trunk
13	Osteoarticular apparatus and muscle of the upper limb
14	Osteoarticular apparatus and muscle of lower limb
15	Osteoarticular apparatus and muscle without precision surveying
16	Integumentary system - mammary glands
17	Acts without precision surveying
18	Anesthetic actions and additional statements
19	Transitional adjustments to the acpc

information is optional and it does not change the severity classification of the inpatient episode. It is encoded according to the information type, such as in CIM-10 if it is a diagnosis or in CCAM if it is a medical procedure. For instance, "*Prostate hyperplasia*" encoded in CIM10 as "**N40**" is a documentary diagnosis because it does not cost anything. In the dissertation documentary information is not treated.

2.6 How to encode diagnoses

Hospitals try to document their activities as accurately as possible to get fair payment. Inaccurate encodings of inpatient episode information could cause inaccurate refundings. Consequently, a lot of effort is made by hospitals to increase encoding accuracy of the diagnoses and medical procedures.

Within each hospital the Medical Information Department DIM (*Département d'Information Médicale*) is responsible for the encoding process which is very sensible as explained by (Busse et al., 2011) "*If up-coding or incorrect coding is detected, hospitals must reimburse payments received. In addition, hospitals may have to pay high financial*

penalties of up to 5 per cent of their annual budgets". Depending on the hospital's size, the DIM usually consists of one or more physicians in charge of the department, specialist coders that have strong medical background, nurses that collect important field information necessary to encode diagnoses properly and finally technicians that are familiar to deal with the management of DIM information.

All the diagnoses are important to encode in order to get an exhaustive information of the performed healthcare and classify the inpatient episode into the right level of severity. It is not an easy task to encode the diagnoses without missing any diagnosis. In order to encode the diagnoses properly, medical data is collected from different healthcare sources, such as discharge letters, laboratory reports, radiology images, patient's consultations, observations, interpretations of the physician and nurses collected information. When all the information is available, the specialist coders read all the sources and code all the diagnoses accordingly in a dedicated program. One of the encoding challenges is encoding all the secondary diagnoses. Unlike primary diagnosis, which is unique and not too difficult to detect, some secondary diagnoses require an extra effort to be identified, because sometimes they are not clearly mentioned in the medical reports and cannot be directly implied.

2.7 Conclusion

In this chapter, the PMSI was introduced emphasizing all the relevant aspects to the dissertation topic, more particularly we emphasized the contents that are related to the diagnoses and how it is encoded. In the dissertation we aim to help the coders encode all the secondary diagnoses. In order to propose appropriate solution for this particular problem of encoding all the secondary diagnoses, we investigate this problem even further in the Chapter 5 by observing real encoding sessions in the hospital and study possible solutions to facilitate the tedious task and increase coders awareness to encode all the secondary diagnoses.

Chapter 3

Relevant information elicitation from medical database

"Intelligence without ambition is a bird without wings."

-Salvador Dali

Contents

3.1 Introduction

Artificial Intelligence (AI) is one of the most interesting fields in computer science, it is attracting a great deal of attention in the information industry and in the society thanks to its role of helping people to achieve everyday tasks more easily. There is no specific definition to AI because of its wide range of applications, it is rather defined by the problems that it deals with. These problems have one thing in common, they automate the intelligent behaviour.

Most of the researchers agree that learning is the basic requirement for any intelligent behaviour. Therefore, one of the major branches of artificial intelligence is Machine Learning (ML). The idea of ML is to identify strong patterns in a database and generate a model that can predict or detect similar cases in the future. Other typical branches of AI are reasoning, knowledge representation, planning and natural language processing.

Feature Selection (FS) is the process of selecting a subset of relevant features (variables, predictors) for the use of a ML model construction. Feature selection techniques are used for many reasons:

- To increase the interpretability of a ML model by removing irrelevant features.
- To accelerate the training of ML models.
- To enhance generalization by reducing overfitting (Guyon and Elisseeff, 2003).

In this chapter some of the most popular FS methods in the scientific literature are presented as well as the applications of these methods in real life. A lot of factors influences the quality of the selected features, the main factors are related to the quality of the database. These factors are explained in this chapter as well as the technical challenges that encounter ML methods to exploit medical databases efficiently. Furthermore, some of the works that address these challenges are presented. Moreover, evaluation approaches of the selected features are explained in general context and detailed furthermore in the context of medical data. Additionally, some of the popular performance metrics are presented in both contexts.

3.2 Feature Selection (FS) methods

The instances used to build a learning algorithm consist of attributes. In supervised learning context, the attributes are divided into input attributes and an output attribute. The input attributes are used to build the ML model and they are called features. The output attribute is the attribute to predict and it is called the prediction class.

The FS methods are part of feature reduction methods. Other feature reduction methods exist that transform and combine the input features creating different features, whereas FS methods choose the most relevant features to the prediction class and ignore other features. The former group methods are referred to as feature extraction algorithms. This dissertation is concerned primarily with the latter group.

The main advantage of FS methods is that they provide a better understanding of the underlying process that generates the data. "*They preserve the original semantics of the variables, hence, offering the advantage of interpretability by a domain expert*" (Saeys et al., 2007). Moreover, they contribute to a better learning performance i.e., better accuracy for learning models and lower computational cost.

3.2.1 FS categories

Feature selection methods can be classified into different categories according to different factors, such as supervised for labeled dataset and unsupervised unlabelled dataset. Supervised methods can be further classified into three categories **Filter**, **Wrapper** and **Embedded** according to how the features are chosen and according to how these methods employ learning models in the selection process. In this section, the supervised FS methods are detailed, since our application domain has labeled training set.

3.2.1.1 Filter methods

Filter methods use a measure quality to rank each feature according to its relevance to the prediction class without the intervention of the learning algorithm. Usually low scoring features are removed and the remaining features are presented as input to the learning algorithm.

Furthermore, filter methods are often classified into **Univariate** and **Multivariate** categories. The former category evaluates only single feature at a time and the later category evaluates and compares whole set of features.

Filter methods are considered *lightweight* because they do not require much time nor resources and they are *independent* of the employed data modeling algorithm, therefore they work equally well with all the algorithms. Filter methods are *scalable* since they can evaluate unlimited number of features in linear time. Filter methods ignore the interactions with learning methods which in some cases could lead to better results if they would have been considered.

Famous filter methods are *Chi-square* (Magidson, 1994), *Fisher score* (Richard et al., 2001), *ReliefF* (Robnik-Šikonja and Kononenko, 2003), *ReliefC* (Dash and Ong, 2011), *Information Gain* (Han et al., 2012), *Gain Ratio* (Witten et al., 2016), *Gini index* (Han et al., 2012) and *Multi-cluster* feature selection (Alelyani et al., 2013).

We provide some details on the most popular filter methods *Information Gain* (IG), *Gain ratio* (GR) and *Correlation-based feature selection* (CFS) methods.

Information gain (IG) This measure is based on information theory, which studies the information content of messages. The IG is used in ID3 decision trees (Quinlan, 1986) in order to choose best feature to split in each node. The IG is *univariate* FS method, it orders the features according to the IG equation 3.3. The information gain for a feature F having v distinct values, $f_1, f_2, ..., f_v$ in a dataset D is given by the (equation 3.1).

$$Gain(F) = Info(D) - Info_F(D) \tag{3.1}$$

Where $Info(D)$ is just the average amount of information required to identify the class label of a tuple in the dataset D (equation 3.2), and $Info_F(D)$ is the expected information required to classify a tuple from D based on the partitioning by F (equation 3.3).

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i) \tag{3.2}$$

$$Info_F(D) = -\sum_{j=1}^{v} \frac{|Dj|}{|D|} \times Info(D_j) \tag{3.3}$$

Where p_i is the probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. D_j contains those tuples in D that have outcome f_j of F.

The IG of a feature F is the difference of information in the original dataset D and the information in the datasets divided according to the values of the feature F.

Gain ratio (GainR) The information gain measure is biased toward features having a large number of values. The gain ratio is an extension to the information gain to alleviate this bias. GainR applies a kind of normalization to IG using a "split information" value defined analogously with $Info(D)$. It is used in C4.5 decision tree (Salzberg, 1993) a successor to ID3. The GainR is a *univariate* FS method, it orders the features according to the GainR given by the equation 3.4.

$$GainRatio(F) = \frac{Gain(F)}{SplitInfo(F)} \tag{3.4}$$

Where the $SplitInfo(F)$ is given by the equation 3.5.

$$SplitInfo_F(D) = -\sum_{j=1}^{v} \times log_2(\frac{|Dj|}{|D|}) \tag{3.5}$$

Correlation-based feature selection (CFS) The CFS is a *multivariate* FS method used to rank subset of features and choose the best subset. Unlike univariate filters, multivariate filters aim to find a set of features that is highly correlated with the prediction class, and the features are not correlated with other features in the set. Therefore, the CFS evaluates the subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them (Hall, 1999).

Correlation coefficients are used to estimate correlation between subset of features and class, as well as inter-correlations between the features.

Relevance of a group of features grows with the correlation between features and classes, and decreases with growing inter-correlation.

CFS is used to determine the best feature subset and is usually combined with search strategies, such as forward selection, backward elimination, bi-directional search, best-first search and genetic search.

The correlation of CFS is given by the equation 3.6

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k(k-1)\overline{r_{ii}}}} \tag{3.6}$$

Where r_{zc} is the correlation between the feature subsets and the class variable, k is the number of subset features, r_{zi} is the average of the correlations between the subset features and the class variable, and r_{ii} is the average inter-correlation between subset features (Hall, 1999).

3.2.1.2 Wrapper methods

Wrapper methods use a previously chosen learning algorithm to select and to assess a subset of features. Exhaustive search can be done in order to assess all the possible subsets. However, subsets count grows exponentially with the features count, and evaluating all the subsets is not always possible, therefore heuristic methods are used in order to find and evaluate the optimal subsets. After defining the strategy to select the subsets, they are evaluated using any desired ML algorithm, then the best evaluated subset is retained.

The advantage of wrapper methods over filter methods is their simplicity, the learning algorithm can be considered as a black box used to assess the usefulness of the features. The main disadvantages of wrapper methods are the high risk of overfitting to the learning algorithm used to assess the features set as well as the high computational cost implicated from assessing each features set (Han et al., 2012).

Famous methods are Sequential Forward Selection (SFS) (Kittler, 1978), Sequential Backward Elimination (SBE) (Kittler, 1978) and Beam search (Siedelecky and Sklansky, 1998).

3.2.1.3 Embedded

Embedded methods incorporate the search for the best features into the process of training a classification model, and are thus specific to one learning algorithm.

This category is less complicated in terms of computations compared to wrapper methods and it is more efficient in terms of considering feature dependencies. The

main drawback of these methods is that they are not highly dependent to the learning algorithms. Embedded methods are not new, such as CART that has a built-in mechanism to perform feature selection (Breiman et al., 1984), weighted Naïve Bayes (Saeys et al., 2007) and Recursive Feature Elimination (RFE-SVM) (Guyon et al., 2002).

3.2.1.4 FS methods comparison

In the dissertation we use filter methods because they are independent from any ML algorithm, scalable and lightweight methods. On the other hand, wrapper and embedded methods are ML algorithm dependent. Some comparison is proposed in the Table 3.1. We intend to use Feature Selection methods independently from any ML algorithm therefore, filter method category is the best candidate to explore its methods.

Table 3.1 FS methods comparison

FS method	**Advantages**	**Drawbacks**
Filter	Lightweight; Scalable; Independent from learning models	Ignores interaction with the model
Wrapper	Simple to implement; Modeling feature dependencies.	Computationally intensive (compared to filter methods); Risk to overfitting to a learning model
Embedded	Less computational complexity (compared to wrapper methods); Feature dependencies	Computationally intensive (compared to filter methods); Dependant to a learning model; Complicated to implement.

In the Table 3.2 the class of some known filter methods is presented including the application task if it is supervised or unsupervised learning. In the dissertation we use labeled dataset, therefore we are interested in supervised methods. Concerning which filter class to use in the dissertation experiments, we decided to use one method from each class (univariate/multivariate). We use *Gain ratio* method as a representative to univariate methods and *CFS* as a representative to multivariate.

Table 3.2 Common filter FS methods

Filter methods	Class	Application task	Reference
Information gain	Univariate	Supervised	(Han et al., 2012)
Gain ratio	Univariate	Supervised	(Witten et al., 2016)
CFS	Multivariate	Supervised	(Hall, 1999)
Chi-square	Univariate	Supervised	(Magidson, 1994)
ReliefF	Univariate	Supervised	(Robnik-Šikonja and Kononenko, 2003)
Fisher score	Univariate	Supervised	(Richard et al., 2001)
ReliefC	Univariate	Unsupervised	(Dash and Ong, 2011)
Multi-cluster feature selection	Multivariate	Unsupervised	(Alelyani et al., 2013)

3.2.2 Applications of FS methods

There are various application domains for the FS methods, we review some well known application domains with additional emphasis to the healthcare domain which is related to our application domain.

3.2.2.1 Text analytics

Text analytics also referred to as text mining. It is the process of deriving quality information from text (Liu et al., 2003). It involves the process of structuring the input text by extracting features, such as word counts, word presence or absence. Text analytics usually produce high dimensionality of feature space. This occurs because usually text analytics consist of using all the words from all the documents in the dataset to build their features which is not necessarily important to their final application. Most of the text analytics applications are text clustering (Liu et al., 2003) and text classification (Forman, 2003). Feature selection methods are applied to select only the important features and accelerate the learning period of the intended model.

3.2.2.2 Image processing

Similarly to text analytics the number of features extracted from an image can be limitless (Bins and Draper, 2001). One of the application domain in image processing is image classification, such as in (Bins and Draper, 2001), they used multiple filter methods (Relief, K-means clustering and sequential floating forward/backward feature selection (SFFS/SFBS)) to rank the important features and exclude the irrelevant and the redundant ones. Other application is detecting breast cancer in x-ray images (Mustra et al., 2012) using different categories of FS methods.

3.2.2.3 Industrial application

Fault diagnosis (Forman, 2003) are the most important examples in the domain of industrial applications. Among fault diagnosis application is fraud detection (Lima and Pereira, 2017) where the best set of features is an essential task to build classification methods that identify frauds. For example, (Lima and Pereira, 2017) used CFS, Gain ratio, Relief filter methods to achieve better classification model in fraud detection application. Industrial applications are proved to be enhanced by using feature selection methods.

3.2.2.4 Healthcare

In the healthcare domain FS methods are used to improve the accuracy of disease prediction, such as (Akay, 2009; Chen et al., 2011) who used filter FS techniques to detect breast cancer and (Su and Yang, 2008) who used FS methods to detect hypertension. (Abeel et al., 2010) used multiple FS methods to detect four cancers: Colon, Leukaemia, Lymphoma and Prostate Cancers.

Other applications of FS, particularly in medical diagnosis, are to identify all the elements that have influenced or triggered a particular symptom or events that chained together to cause a disease. There are three application domains in bioinformatics according to (Saeys et al., 2007): feature selection for **sequence analysis**, **microarray analysis** and **mass spectra analysis**.

The FS for **sequence analysis** is the focus of many researches, since the early days of bioinformatics, one of the applications is to predict the sub-sequences that code

for proteins (Al-Shahib et al., 2005; Chuzhanova et al., 1998), other applications involve recognizing less conserved signals in the sequence representing mainly binding sites for various complex proteins, such as (Keles et al., 2002; Tadesse et al., 2004).

Concerning **microarray analysis**, its objective is to identify the responsible genes that cause certain disease, the primary source of data is microarray database which consists of several hundred of thousands of gene features against small sample sizes. Univariate filter methods category is the most popular category used in microarray analysis thanks to its simplicity and efficacy with huge number of features (Dudoit et al., 2002; Lee et al., 2005; Statnikov et al., 2005). Other FS categories have been also used, such as multivariate filter in (Ding and Peng, 2005; Wang et al., 2005) to discover the responsible genes for NCI, Lymphoma, Lung, Child Leukemia and Colon cancers. Embedded methods are used in (Guyon et al., 2002) to detect a distinctive pattern of DNA or the responsible genes for cancer disease using SVM methods.

Finally, **Mass spectro analysis** is a framework for disease diagnosis and protein-based biomarker profiling. A mass spectrum is the distribution of ions represented by thousand of mass to charge ratios against their intensities. It is used to discover the patterns in complex mixtures of proteins derived from tissue samples or fluids. For example, (Petricoin et al., 2002) analysed the mass spectrum and clustered the patterns that cause ovarian cancer (Coombes et al., 2007). (Ressom et al., 2005) processed mass spectral data to achieve high prediction accuracy in distinguishing liver cancer patients from healthy individuals.

3.2.3 Evaluation approaches

The common practice in the literature to evaluate the quality of the features is to build a ML model out of the features and to measure its performance. The proper features produce good quality classification model and vice versa. Features selected using filter and wrapper methods can use any ML model. The main objective of FS methods is to produce better classification models.

Since wrapper methods depend on a specific learning method to select the features, by default these methods use the performance of the same learning method to evaluate their features (Chrysostomou, 2009).

For pattern recognition and image processing neural networks are generally used (Egmont-Petersen et al., 2002). For other applications SVM, Naive Bayes and Decision Trees are often used, such as in (Alonso-González et al., 2010; Cavallaro et al., 2015; Lima and Pereira, 2017; Maldonado et al., 2014; Popescu and Khalilia, 2011)

In this section, we detail some of the most used classifiers to evaluate feature selection methods in the literature. We emphasize some points that are important in the context of the dissertation, such as accuracy, training speed, scalability, interpretability, clarity and simplicity. In the dissertation, we use medical databases and we collaborate with medical experts, therefore we are particularly interested in two issues (*scalability*, *interpretability*). First, medical databases are huge in volume and require scalable approach. Second, we collaborate with medical experts and we are interested in showing and discussing the results with them therefore interpretability is a very important area to consider.

The first issue we emphasize is the *scalability*, traditional methods assume that all training examples can be stored in main memory (Han et al., 2012), therefore they are limited to an equal or smaller memory size of training sets. Nowadays, the data generation becomes very fast and the traditional classification methods are not adapted anymore to process the enormous volume of data.

Actually, two methods are proposed in the literature to handle scalability issue, either by using *MapReduce* techniques or by *adapting* the existing methods to process huge volumes of data.

The first method uses MapReduce technique (Dean and Ghemawat, 2004), a simple but powerful programming technique, large number of computer clusters to process and to distribute data automatically. The computing model consists of two functions, Map and Reduce. The Map function processes a part of the input and transforms it into key-value pairs. Then, the Reduce function combines all the intermediate values related to the same key. The MapReduce takes care of all the complicated steps for developing parallel applications so the user only needs to program two functions to develop a scalable application.

The second method modifies the existing model to a streaming model (Han et al., 2012) that processes learning instances one by one i.e. the adapted version can learn sequentially when the examples arrive.

The second issue we emphasize is the *interpretability* and the clarity of the built model. Each ML algorithm produces a model that predicts and assigns the class of the future instances, some of the produced models are easy to interpret and are self-explanatory, it is easy to understand how the prediction has being made and it is easy to diagnose the error by tracking each decision, whereas other ML algorithms produce models that are very difficult to interpret and explain, and there is no way to understand how the prediction has been made, these kind of models are called black box models (Witten and Frank, 2005).

We present in the following subsections four of the most used classification algorithms to evaluate the FS methods. (Artificial Neural Network (ANN) - Decision Trees (DS) - Naive Bayes (NB) - Support Vector Machines (SVM))

3.2.3.1 Artificial Neural Network (ANN)

A neural network is a complex adaptive system that can change its structure based on the input used in the training set. The basic component of the ANN model is a node also named an artificial neuron, since it has similar functioning to an actual neuron in the human brain. The model can have numerous nodes build up on many levels where each node builds connection with other nodes depending on the desired output (Han et al., 2012), as shown in Figure 3.1.

The advantage of this method is that it can handle large amount of input features without knowing their nature. The ANN is best used in pattern recognition field which is considered as one of the difficult tasks for a computer to perform. The pattern recognition applications range from character recognition to facial recognition. The ANN is also used in simple classification problems where the output is 0 or 1 i.e. two class classifiers (Witten and Frank, 2005).

To explain how a neural network works we explain the perceptron invented in 1957 by Frank Rosenblatt which represents the simplest neural network possible, a network with only one neuron Figure 3.2. A perceptron has 3 inputs, a weight for each of the inputs and a processor. Depending on the input and the weight for each input the processor decides what is the output. The Equation (3.7) represents the output of the perceptron where the b is the bias. The bias shifts the decision boundary away from the origin and does not depend on any input value (Shiffman, 2012).

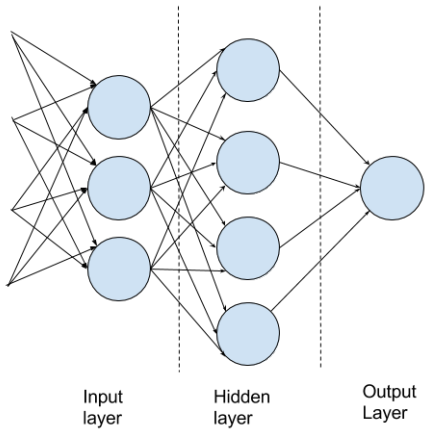


Fig. 3.1 Neural Network structure

$$Output = f(\Sigma_{k=1}^{n} i_k.W_k + b) \tag{3.7}$$

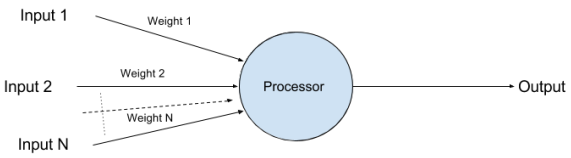


Fig. 3.2 A perceptron

The simple form of the perceptron can decide whether an input belongs to one class or another, it is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

A group of perceptrons forms a network that can solve nonlinear classification problems, usually the perceptrons are grouped in three layers, the first layer is called input, the second layer is called hidden layer, the third layer is the output layer Figure 3.1. Deep learning is a new type of neural networks where more than one hidden layers is used in order to solve even more complicated problems, such as speech recognition and visual object recognition. (LeCun et al., 2015)

Scientist used the ANN in the medical field in order to predict medical diagnosis. On one hand, the ANN offers a number of advantages, such as requiring less training to develop a good model, the ability to implicitly detect complex nonlinear relationships between independent and dependent variables, a good performance if it is configured

properly. On the other hand, there are some disadvantages related to this model, such as the black box nature of the model and the proneness to overfitting compared to other ML algorithms (Tu, 1996). In addition, it requires a lot of experimentations in order to choose the right parameters that lead to the best results. i.e. the number of neurons, the size of the hidden layers and the value of the learning rate. The most important disadvantage of the ANN is the significant amount of time required in order to build a prediction model, it is considered slower than other ML methods (Witten et al., 2016).

As one of our objectives is to identify the best features used in a model, we have disconsidered the use of ANN, since they use a black box model, difficult to determine which elements contributed most to predict the output.

An important aspect we are interested in is the scalability, since we plan to use large databases that cannot fit necessarily in the RAM memory of a computer, the ANN with backpropagation learning method has straightforward ability to learn from streamed dataset (Bifet and Kirby, 2009), which can be useful to avoid the problem of not being able to fit the entire training set in the RAM memory of a certain computer. In addition, it is possible to use the ANN under MapReduce algorithm which divides the training sets into small chunks and distribute them into parallel machines (Wu et al., 2014).

3.2.3.2 Decision Trees (DT)

One of the famous ML classification methods is Decision Tree (DT). A DT is a flowchart tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label, the topmost node in a tree is the root node (Han et al., 2012). A typical DT is shown in Figure 3.3, representing the concept buys_computer and predicting whether a customer is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals. Some DT algorithms produce only binary trees where each internal node branches to exactly two other nodes, whereas others can produce n-ary trees (Han et al., 2012).

DT can make a prediction by testing the attribute values of a tuple against the nodes of the DT. A path is traced from the root to a leaf node, which holds the class prediction for that tuple.

DTs are considered popular because they offer structural description of what has been learnt. The description can be understood by people and can be used later to explain on what basis the prediction has been made.

The applications of ML in the medical domain and some other domains prefer understanding the explicit knowledge structure that are acquired, because gaining knowledge from data can be much more important than the ability to perform well on new examples (Magoulas and Prentza, 2001).

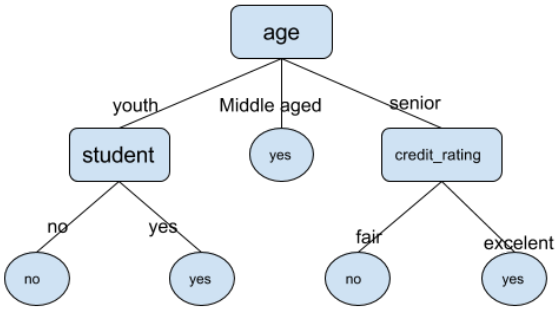


Fig. 3.3 AN example of a Decision Tree

DTs are widely used because they generate simple models, easy to interpret. Any DT can be converted into a set of rules explaining each prediction. Therefore, DTs can be validated by physicians who are not necessarily specialists in ML. DTs are scalable and produce efficient models even when using large amounts of data (Magoulas and Prentza, 2001).

There are many algorithms that implemented DTs, the very beginning was in the late 1980s with the *ID3* algorithm (Iterative Dichotomiser) (Quinlan, 1986) later on the *C4.5* algorithm (Salzberg, 1993) was implemented the DTs which is considered a successor of ID3. Another group of statisticians published *Classification and Regression Tree (CART)* (Breiman et al., 1984) algorithms as induction algorithm as it allows to build a binary DT. Later, other types of DTs are designed, such as *Random forest* (Breiman, 2001) which consists of multiple trees, *NBtree* (Kohavi, 2011) which combines Naïve Bayes algorithm in it and *Ensemble methods* (Polikar, 2006) which consists of combining multiple DTs in order to have better classification model.

A common problem in the DT algorithms is overfitting. The errors made by a learning model are divided into types training set errors and testing set errors. The training set is the data used to build the model, whereas the testing set is unseen and

not used to build the learning model and it is usually used to measure the prediction power of the learning model. Usually a learning algorithm should perform well on both training and testing sets. Overfitting problem occurs when a learning algorithm is trained to perform well only on the training data. This occurs often in the DTs because they fully grow branches to consider all the outliers in the training set at the expense of poor performance on the testing set. Tree pruning methods address overfitting problem in DTs by removing the least reliable branches. Pruned trees tend to be smaller and less complex therefore faster, easier to understand and classifies better on the testing set.

There are two kinds of tree pruning: prepruning and postpruning (Han et al., 2012). In the prepruning approach, a tree is pruned by halting its construction early e.g., by deciding not to further split or partition the subset of training tuples at a given node.

The second and more common approach is postpruning, which removes subtrees from a "fully grown" tree. A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labeled with the most frequent class among the subtree being replaced.

Concerning scalability issues, the standard version of DT cannot handle large training sets that cannot fit into the memory, therefore new DT algorithms are proposed to address the scalability issues, such as "Adaptive Hoeffding trees" proposed by (Bifet and Gavaldà, 2009) (Han et al., 2012). The idea behind an adaptive DT is that it is built incrementally from data streams. The main advantage of Hoeffding adaptive trees over other adaptive trees is that it does not require user defined parameter to guess how fast the stream flows.

Concerning the second method to scale the DT algorithm, most of the researches programmed MapReduce functions to cover most of the DT algorithms, such as CART (Chrysos et al., 2013), ID3 (Wang-Wei, 2012), C4.5 (Dai and Ji, 2014) regression trees (Yin et al., 2012), random forest (Li et al., 2012) and ensemble trees (Panda et al., 2009). MapReduce functions permit the algorithms to run over many computers in parallel, consequently speeding them up and eliminating the dataset size problem.

3.2.3.3 Naive Bayes (NB)

Bayesian classifiers are statistical classifiers based on Bayes theory (Domingos and Pazzani, 1997). They can predict class membership probabilities by measuring the probability that a given tuple belongs to a particular class.

The classifier is called "Naïve" because it assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. Thanks to this assumption, the calculations are simplified and the learning is faster. It is one of simplest approach yet very powerful compared to other ML classifiers.

The idea behind Bayes theorem is that it relates conditional or posterior probability with the marginal probabilities. For example, for two events A and B, the Bayes rule can be written as:

$$P(A|B) = \frac{p(A)P(B|A)}{P(B)} \tag{3.8}$$

where $p(A)$ is the prior probability of A. $P(A|B)$ is the conditional/posterior probability of A given B. $P(B|A)$ is the conditional/posterior probability of B given A. $P(B)$ is the prior probability of B.

Based on the Bayes theorem and the naïve assumption of class conditional independence, the attributes can be predicted $X = (x_1, x_2...x_n)$ to which class they belong $C = (C_1, C_2...C_n)$ by the equation below

$$P(X|C_i) = \prod_{k=1}^{n} (x_k|C_i) = P(x_1|C_i)P(x_2|C_i)...P(x_n|C_i) \tag{3.9}$$

These probabilities can be easily measured from the training set and assign the highest X with the highest class probability. For the full details of how NB classifier works check the reference (Han et al., 2012).

NB model is stable and robust, not too difficult to interpret (Murali et al., 2016) thanks to its independent attributes assumption. The main advantage of NB classifiers is the high accuracy and the high speed when applied to large databases. It can learn naturally without any adaptation from unlimited dataset thanks to its sequential algorithm. The model can learn incrementally example by example without the need to scan all the training set, therefore the memory usage is small and bounded. (Bifet and Kirby, 2009)

NB can be used on parallel computers thanks to MapReduce implementations, such as Liu (Liu et al., 2013) and it can be used freely using "Mahout[1]" an open source implemented by "Apache[2]". Therefore, NB is one of the best scalable algorithms that can learn both by streaming the dataset or by parallel programming using MapReduce method.

3.2.3.4 Support Vector Machines (SVM)

A Support Vector Machine SVM is an algorithm that uses a nonlinear mapper to transform the original training data into a higher dimension (Cortes and Vapnik, 1995). Within this new dimension, the algorithm searches for an optimal linear separating hyperplane i.e. separating the tuples of one class from another. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. The SVM finds this hyperplane using support vectors essential training tuples and using margins defined by the support vectors.

The first paper on SVM was presented in 1992 by Vladimir Vapnik and his colleagues Bernhard Boser and Isabelle Guyon, although the groundwork for SVMs has been around, since the 1960s (including early work by Vapnik and Alexei Chervonenkis on statistical learning theory). Although the training time of even the fastest SVMs can be extremely slow, they are highly accurate, thanks to their ability to model complex nonlinear decision boundaries. They are much less prone to overfitting than other methods (Suykens et al., 2015). Other methods are proposed and explored to improve the performance and accelerate the calculations (Do and Poulet, 2006; Graf et al., 2005), SVMs can provide a compact description of the learned model. SVMs can be used in prediction as well as in classification. They have been applied to a number of areas, including handwritten digit recognition, object recognition, and speaker identification, as well as benchmark time series prediction tests (Cavallaro et al., 2015).

The SVM model is considered as one of the models that is difficult to understand. Some papers tried to solve the problem by using fuzzy logic to make the model more interpretable (Nguyen and Le, 2014). A complete introduction to SVMs and for more technical details can be found in (Cortes and Vapnik, 1995).

[1] http://mahout.apache.org
[2] http://apache.org

A traditional SVM algorithm is not scalable, because it is formulated in terms of quadratic program that requires a lot of resources and time, for a training set of N records, the storage requirement is O(N2) and time complexity is about O(N3). Therefore, a major research goal regarding SVMs is to improve the speed in training and testing so that SVMs may become a more feasible option for very large data sets. Some papers solved the scalability problem by adapting the algorithm to build the model in iterative steps, some of them require one pass (Rai et al., 2009) others require multiple passes on the training set (Domeniconi and Gunopulos, 2001) in order to build a model with better performance.

Other papers solved the scalability problem by creating parallel algorithm (Do and Poulet, 2006; Graf et al., 2005) and more recently using MapReduce method (Caruana et al., 2011; Sun and Fox, 2012).

3.2.3.5 Classification methods summary

There are a lot of other ML algorithms in the literature, such as logistic regression, KNN K-Nearest Neighbor, Adaboost and other adaptations depending on the application needs, we tried to cover the most known algorithms and the ones we use in the thesis.

A brief summary of the reviewed classification methods are presented in the table 3.3 highlighting the important points that we are interested in.

- **Scalability**: the ability of the method to function well when the dataset contains millions or billions of data objects.
- **Interpretability, comprehensibility, understandability**: the ability of the model to express the behaviour of the learnt model in an understandable way.
- **Training speed**: the time required to build a model.

We work with sensitive medical data that cannot be moved outside the hospital. In addition, we are allowed to work only on one computer, therefore, *streaming* data has big advantage in our case over *MapReduce* technique that uses more than one computer.

In the dissertation we use DT mainly for the interpretability reasons, since the evaluations are discussed with medical experts as well as for the high training speed and

Table 3.3 Classification methods comparison

Method	Interpretability	Scalability		Training speed	Implementation difficulty
		Streaming	MapReduce		
Naïve Bayes	Medium	High	High	High	Easy
Decision Trees	High	Medium	High	High	Easy
SVM	Low	Medium	High	Low	Difficult
Artificial Neural Networks	Low	High	High	Low	Difficult

acceptable streaming scalability compared to other algorithms. We use also NB algorithm for the sake of performance comparison. Furthermore, both NB and DT are easy to implement, since they have either few or no parameters to be tuned.

3.2.4 Performance metrics

Evaluation is the key to measure the performance of Machine Learning methods and choose the best one.

For a classification model, its performance is measured in term of error rate. The classifier predicts the class of each instance, if it is correct, it is counted as a success; if it is not correct, it is counted as an error. Therefore, the error rate represents the proportion of the errors made of the all predicted instances. It is not interesting to measure the performance using the same instances used to train the model, the main objective of training a model predicts future cases which are not seen by the model. Therefore, the performance of a model is measured on new dataset that played no part in the formation of the model.

This independent dataset is called testing set used only to measure the model performance, and the dataset used to train the model is called training set, with the assumption that both training and testing set represent most of the cases of the problem. If there is enough of instances, the dataset can be divided into training set and testing set, otherwise if the dataset is limited other adapted methods are required to alleviate the dataset shortage. One of the general methods is called “Cross-validation” used mainly when there is not enough data or to have more reliable evaluations. The idea behind cross-validation is to repeat the dataset dividing process several times with different random training and testing sets. In each iteration a certain proportion of the data is randomly

selected for the training set and the rest is used for testing set. The error rate at the end of the iteration is calculated to estimate an overall error rate. In the k-fold cross-evaluation, the k is the number of the iteration. 10-fold cross-validation is the most standard way used to measure the error rate of Machine Learning model, however other dataset dividing methods exist used in particular scenarios, such as "The bootstrap","Leave-one-out" (Witten and Frank, 2005).

In most applications simple error rate is not enough to evaluate the ML performance, specially if the classes distribution is imbalanced. For example, if there were 90% of the examples from the first class and 10% from the second class the model could be easily biased towards classifying all the examples as the first class. The overall error rate is 10%, but in practice the classifier has 0% error rate for the first class and 100% error for the second class. Therefore, other methods exist to alleviate this problem.

Table 3.4 Different outcomes of a two-class prediction "Confusion-matrix"

		Predicted class	
		Yes	No
Actual class	Yes	True Positive (TP)	False Negative (FN)
	No	False Positive (FP)	True Negative (TN)

Table 3.4 has the basic 4 units that most of the evaluation methods are based on to build their equations.

- **TP** is the number of True Positive instances, which represent instances that are correctly assigned to positive examples.
- **TN** is the number of True Negative instances, which represent instances that are correctly assigned to negative examples,
- **FP** is the number of False Positive instances, which represent instances that are incorrectly assigned to positive examples,
- **FN** is the number of False Negative instances, which represent instances that are incorrectly assigned to negative examples.

The standard metrics are used to evaluate classification Accuracy, Precision, Recall and F1-measure. The measurements are defined based on the following sets according to (Tuffery, 2007).

Accuracy is the ratio of correctly assigned negative and positive examples to the total number of examples.

$$A = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{3.10}$$

Precision is the ratio of correctly assigned examples to the total number of examples produced by the classifier. A precision score of 1.0 for a Class C means that every item labeled as class C indeed belongs to Class C but it says nothing about the number of items from class C that were not labeled correctly.

$$P = \frac{TP}{(TP + FP)} \tag{3.11}$$

Recall or **Sensitivity** or **True Positive Rate** is the ratio of correctly assigned examples to the number of target examples in the test set. A perfect recall score of 1.0 means that every item from Class C was labeled as belonging to class C but it says nothing about how many other items were incorrectly also labeled as belonging to class C.

$$R = \frac{TP}{(TP + FN)} \tag{3.12}$$

False Positive Rate is the rate of the negative examples predicted as positive.

$$FPR = \frac{FP}{(FP + TN)} \tag{3.13}$$

Specificity is the proportion of negatives examples that are correctly identified as such.

$$S = \frac{TN}{(TN + FN)} \tag{3.14}$$

F-measure represents the harmonic mean of precision and recall according to the Equation 3.15 with the possibility to give more weight to either the precision by choosing a value for β smaller that 1 or to give more weight to Recall by choosing a value greater than 1, the most common value of β is 1, Equation 3.16.

$$F_\beta = (1 + \beta^2) * \frac{P * R}{(\beta^2 * P) + R} \tag{3.15}$$

$$F_1 = \frac{2 * P * R}{(P + R)} \tag{3.16}$$

G-mean evaluates the degree of inductive bias in terms of a ratio of positive accuracy and negative accuracy, Equation 3.17.

$$G-mean = \sqrt{\frac{TP}{(TP+FN)} \times \frac{TN}{(TN+FP)}} \tag{3.17}$$

ROC and AUC: Receiver Operating Characteristic (ROC) is a curve created by plotting the true positive rate against false positive rate. Area Under Curve (AUC) it is used to have normalised value in order to compare different curves. The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). AUC has a range of [0.5,1] 0.5 rate indicates that the model classifies the examples in complete random manner while the 1 rate indicates that the model classifies them perfectly.

In the context of medical diagnoses, the specificity, sensitivity and AUC are used more often to indicated how well the classifier is performing detecting the positive and negative examples. For example, test sensitivity is the ability of a test to correctly identify those with the disease (true positive rate), whereas test specificity is the ability of the test to correctly identify those without the disease (true negative rate) and AUC combines the two measures in a single normalised unit.

3.3 Technical challenges of using FS with medical databases

Learning from real data, especially when dealing with medical data sets, raises several challenges.

One common challenge is the **imbalanced datasets**, where the targeted observation is usually under-represented among other representations of the data sets. The unfair repartition of the examples can reach up to 10,000:1 ratio in some medical databases (PMSI). For example, *Respiratory failure*, a common pneumonia disease concern at best 1% of the inpatient episodes. Most learning algorithms, including classification and FS methods, build their models using the majority examples and ignore the few "important" examples. Even the common evaluation methods, such as accuracy fail to measure the performance of classification algorithms.

Another issue is the **simplicity** to understand and use the algorithm. The learnt model will be used as medical decision aiding tool. Therefore, the proposed solution should provide a good interpretability, allowing the users assessing the validity of the proposed aid.

In ML domain, in order to effectively learn from a dataset it should be in **a proper format**, i.e. all the information of the studied subject should exist in a flat database. (Han et al., 2012). Generally, most of the medical databases are relational databases, where an information concerning a particular subject exists in multiple tables with different relationships "one to one", "one to many" and "many to many".

Finally, the last issue treated in this section is the **excessive number of features** in the medical databases that can limit the performance of the ML methods.

3.3.1 Imbalanced database

In recent years, the imbalanced learning problem has drawn a significant amount of interest from academia, industry, and government funding agencies. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly compromise the performance of most standard learning algorithms. Most standard algorithms assume or expect balanced class distributions. Therefore, when presented with complex imbalanced data sets, these algorithms fail to properly represent the distributive characteristics of the data and resultantly provide unfavourable accuracies across the classes of the data. This challenge is relatively new and it receives an increased rate of attention over years. The rapid expansion and the consistent assessments of past and current works in this field contribute to the received attention. Moreover, the possible projections for future research are essential for long-term development (Haibo He and Garcia, 2009).

3.3.1.1 Resampling methods

These methods are used to rebalance the class inequality in the imbalanced dataset to alleviate the risk of the dominance of the majority class at the expense of the minority

class. Resampling methods are the most used methods so far, since they are independent from learning algorithms.

Three main sampling methods that tackle this problem are presented:

- **Undersampling**: It creates new dataset of the original dataset by eliminating instances from the majority class.
- **Oversampling**: It creates new dataset of the original dataset by adding instances from the minority class.
- **Hybrid**: A combination of the undersampling and oversampling methods

Some of the most famous methods are presented in the following section:

- ***Random sampling***: or non informed methods, these methods are the simplest techniques since they use non heuristic methods. The most famous methods are random undersampling and random oversampling. As the name suggest, it is about adding and removing instances randomly. The major draw back of the random undersampling is that it can discard potentially useful data to the learning process, whereas the drawback for the random oversampling is the increased likelihood of overfitting, since it duplicates the existing instances. The random sampling methods are simple to both understand and visualize, thus we refrain from providing any specific examples of its functionality. Random sampling is considered the baseline method in order to compare other new more complicated sampling methods.
- ***Informed sampling***: these methods use heuristics in order to alleviate the deficiency of information loss introduced in the traditional random undersampling method.
  - Examples of informed undersampling are: *NearMiss-1* which removes negative examples whose average distances to three closest positive examples are the smallest, *NearMiss-2* which removes negative examples based on their average distances to three farthest positive examples in addition to *NearMiss-3* which removes negative examples to guarantee every positive example is surrounded by some negative examples. These algorithms are proposed by (Zhang and Mani, 2003) that use the K-nearest neighbor (KNN) classifier to achieve undersampling. Similar algorithm Condensed Nearest Neighbor is proposed by (Angiulli, 2005).

- An example of informed oversampling is "Synthetic Minority Oversampling TEchnique" SMOTE (Chawla et al., 2002). The main idea of SMOTE is to add new artificial minority examples by interpolating between pre-existing minority instances.
- An example of hybrid informed sampling is (*Outlier sampling*) proposed by (Lima and Pereira, 2017). The main idea of this method is to remove the rare instances of negative class and replicate the rare instances of positive class, using the SMOTE method.

Although sampling methods increase the performance of the learning methods in general, not enough research is done to assess the quality of the features extracted using feature selection methods in particular.

Among the related works that targeted the feature selection is an embedded feature selection approach proposed by (Maldonado et al., 2014) using backward elimination approach based on successive holdout steps of features. Starting from S full set of features, they search for a subset K ($K \subseteq S$) features so that the performance of the SVM classifier using this subset of features is maximized. The contribution of (Maldonado et al., 2014) is to add hold out set of data in order to evaluate the SVM classifier. The main drawback of the proposed approach is the dependency to the classification algorithm (SVM algorithm).

Another related work is proposed by (Yin et al., 2013), clustering the dataset with the majority class into numerous clusters, applying a FS method on each cluster and finally retaining the common features among all the clusters. The evaluation of the paper showed that the method enhanced the classifiers performance in comparison with the performance when only one FS method is applied. Another similar work is proposed by (Martín-Félez and Mollineda, 2010) in the context of identifying the melodic track given by a MIDI file they proposed a new sampling method based on clustering the training set. Moreover, they studied the effect of combining sampling methods with feature selection methods and the order in which they are applied. The paper concludes that applying sampling methods first has better effect on the features.

3.3.1.2 Cost-sensitive learning

Contrary to the sampling methods that create balanced data distributions through different sampling strategies, cost-sensitive learning addresses the imbalanced learning problem by adding the notion of the cost associated with misclassifying examples. In other words, it assigns cost matrices to describe the cost for each classified example where C_{ij} represents the misclassification cost of assigning examples of the class i to class j (Elkan, 2001).

The costs can be determined by experts in the domain, or by other approaches (Sun et al., 2007). In the case of imbalanced learning, when recognizing that the minority class is more important than recognizing the majority class, the cost of misclassifying the minority class is higher than the majority class. In this way, the classifier gives more importance to the minority class.

Two main methods exist to incorporate the cost sensitivity into the learning process:

- **Direct methods**: The direct methods modify the behaviour of the learning algorithm to consider the misclassification costs while building the model. For example, in the decision trees, the split criteria is adapted to consider the misclassification costs (Witten et al., 2016), or the pruning methods of the tree can be adapted to use the matrix in order to determine if a subtree can be pruned or not (Bradford et al., 1998).
- **Indirect methods**: The indirect methods do not modify the main learning algorithm, they just integrate in the preprocessing or the postprocessing of the data. For example, in the postprocessing method, a tradition decision tree assigns the node with the majority class, in the cost-sensitive learning, the decision tree assigns the node with the class that minimizes the classification cost (Domingos, 1999). Another example to preprocessing case, is to resample the training set according to the matrix cost. In this case, cost-sensitive method is equivalent to the sampling method (Zadrozny et al., 2003).

Compared to the sampling method, cost-sensitive learning is computationally more efficient, however it is less popular due to two reasons according to (Haixiang et al., 2017). The first reason is the difficulty to assign cost values to the matrix even if the expert are available, the second reason is the difficulty to integrate cost matrices in the learning process compared to use sampling methods.

A lot of works used cost-sensitive learning with imbalanced dataset which indicates it is a good alternative to overcome the imbalanced learning problem. In some cases, it surpasses the sampling methods (McCarthy et al., 2005; Zhou and Liu, 2006). Consequently, cost-sensitive learning is a practical alternative to the sampling methods.

3.3.1.3 Ensemble methods

A good technique to overcome imbalanced learning is to combine multiple learning methods and use them as one. Ensemble methods also known as multiple classifier systems (Polikar, 2006) are used in order to produce a new learning method that outperforms every independent method.

One of the most famous methods is Adaboost (Schapire, 1999); the outputs of the used methods in the Adaboost are combined into a weighted sum that represents the final output of the boosted learner.

Ensemble learner have become a popular solution for class imbalance problems (Haixiang et al., 2017). The ensemble learners for imbalance problem are classified into two main categories according to (López et al., 2013): *Cost sensitive ensembles* and *Data preprocessing + Ensemble learning*.

Examples of the first category "*Cost sensitive ensembles*" include: AdaCost (Fan et al., 1999), CSB1,CSB2 (Ting, 2000), RareBoost (Joshi et al., 2001) and AdaC1, AdaC2, AdaC3 (Sun et al., 2007).

Examples of the second category "*Data preprocessing + Ensemble learning*" include: SMOTEBoost (Chawla et al., 2003), MSMOTEBoost (Holte et al., 1989) RUSBoost (Seiffert et al., 2010), OverBagging (Wang and Yao, 2009), UnderBagging (Barandela et al., 2003), EasyEnsemble (Liu et al., 2009) and BalanceCascade (Liu et al., 2009).

This taxonomy identifies cost-sensitive boosting methods, which differ from cost-sensitive approaches by the use of a boosting algorithm that guides the minimization costs procedure. Furthermore, the second category distinguishes three families of ensemble methods (boosting, bagging and hybrid) that apply data preprocessing techniques before applying the ensemble method.

The main disadvantage of the ensemble methods is the high correlation of their performance with the base classifier used in them. The user should carefully choose the base classifier according to the application domain. For example, SVM classifier is good at handling missing values but has difficulty with large scale data, whereas decision trees are good at handling missing values but fail to model small size data (Li et al., 2016).

3.3.1.4 Adapted learning algorithm

An alternative strategy to deal with the imbalanced datasets is altering the base model to adapt and to improve the classification performance for imbalanced data. There are a lot of works in the scientific literature that modified the algorithms. The most used base classifiers are SVM, decision tree, Neural networks, K-nearest neighbour, rule-based classifiers in order, based on a review performed by (Haixiang et al., 2017).

For example, (Cieslak and Chawla, 2008) proposed a decision tree with a new splitting criteria "*Hellinger distance*" less sensitive to the class imbalance which showed better performance than traditional DTs when no sampling methods are applied.

One of the interesting works in the medical domain is done by (Jacques et al., 2015) who proposed an algorithm called MOCA (*Multi-Objective Classification Algorithm for Imbalanced data*), it deals with the uncertainty of the negative examples justified by the absence of real negation in the medical files, i.e. the absence of medical diagnosis in the medical file does not mean necessarily the patient does not suffer from the disease.

The modified learning algorithms proved to enhance the base classifiers, however they are designed to work well only in specific application domain. Moreover, the users are restricted in the choice of the learning algorithms that have been modified in order to meet their goals.

3.3.1.5 One-class learning

Traditional classifiers use two or more classes in order to train a model, one used as positive examples and other classes as negative examples. In one-class learning, only a single class is used to train the classifier. The classifier is usually built by estimating the density of the target class (Tao et al., 2004).

For example, the traditional SVM strategy is to map the examples into a higher dimensional feature space corresponding to a kernel and separate the classes using a hyperplane. On the other hand, the one-class SVM strategy is to use a hypersphere is used instead of hyperplane to surround the single class examples. This technique is called Support Vector Data Description (SVDD) (Schölkopf et al., 2001). One-class SVM is proposed by (Schölkopf et al., 2001). The downsides of one-class SVM are that it requires high computation power, has low accuracy compared to the traditional classifier and more crucially that the user is required to supply suitable values for crucial parameters in order to maximize the performance of the classifier, which is not an easy task.

Some enhancement are proposed to the one-class SVM, such as in (Zhang et al., 2015) who proposed Least square fuzzy approach to increase accuracy and to decrease the complexity by solving linear equation instead of complex equation. (Theissler et al., 2015; Wang et al., 2010; Zhuang and Dai, 2006) proposed automatic assignment of the one-class SVM by using the negative examples if they are available. (Hovelynck and Chidlovskii, 2010) propose an extension to the one-class evaluation framework when only some positive training examples are available.

Using one-class learning in the context of imbalanced datasets has shown a good potential in achieving good performance classifiers (Chawla et al., 2004). However, one-class learning is still not competitive enough to the sampling method to overcome the imbalanced dataset challenge.

3.3.1.6 Evaluation methods

Model evaluation is a very crucial process in machine learning. There are a lot of performance measures to evaluate the effectiveness of the learner. However, in the context of imbalanced learning, not all of the measures are adapted to evaluate the learners. For example, the accuracy is the most commonly used measure in traditional learning, but it is not adapted for the imbalanced dataset because of the bias toward the majority class. Some of the performance metrics used with imbalanced learning are F-measure, precision, recall, ROC, AUC and G-Mean (explained in the Section 3.2.4). These metrics are less likely to suffer from imbalanced distributions as they take class distribution into account.

There are some works focused on proposing novel evaluation metrics for imbalanced data, such as (Maratea et al., 2014) who proposed adapted version of F-measure,

called *Adjusted F-measure*, (Batuwita and Palade, 2012) who proposed *adjusted geometric mean*, and (Weng and Poon, 2008) who proposed *weighted AUC*, they divided the area under ROC curve into sections and assigned each section with different weight in order to overcome the imbalance problem.

3.3.1.7 Summary

In the dissertation we focus on the sampling methods to solve the imbalanced problem, since it is the only method independent of the learning algorithm. All other methods provide better results with specific learning algorithms, whereas we search for a general method that can work with any learning algorithms in general, and can work with feature selection method in particular.

Therefore, we propose an approach based on sampling methods, independent from the learning algorithm.

3.3.2 Interpretability

One of the important challenges related to the usage of medical databases with learning methods in general and feature selection methods in particular is the interpretability of the extracted results. The understandability of the generated results allows the users assessing their validity, since the selected features will be part of a medical decision aid tool.

Some of the classification methods are interpretable by nature, such as Decision Trees (DT) since its model can be decomposed easily into simple if-then rules which enable a transparent understanding of model behaviour and validation by practitioners. Other classification methods are considered as a black boxes by practitioners who therefore suffer to interpret the results, such as Artificial Neural Networks (ANN). Other ML methods are somewhat interpretable, such as SVM but it has been proposed to increase the interpretability of these methods by decreasing the dimensional space of the feature representation by using feature selection methods. The reduced feature makes the model more interpretable (Maldonado et al., 2014).

Concerning feature selection methods, they all select the important features. Their main role is to enhance and accelerate the ML model. However, these features lack the interpretability, and are useless outside the learning process context. They are not understandable by humans because their values are unknown and the relationships between each other are undiscovered. For example, it is impossible to diagnose a disease if only important features are presented to a physician without being aware of the features value and the relationship between each other.

The interpretability of the features selection methods are rarely addressed in the scientific literature. We are interested in the dissertation not only in selecting stable and robust features but also in the interpretability of these features. Therefore, we propose methods to understand the features and use them in order to achieve important tasks more easily. In the context of bio gene selection (Haury et al., 2011) investigated which FS method provide better understandability to the physician. Moreover, the interpretability of the features are pointed out in more general context in future perspectives in the (Maldonado et al., 2014).

3.3.3 Database format

Generally, most of the medical information are stored in relational databases, where a relational database decomposes data in multiple tables (relations). These tables are related to one another according to relational model. In the ML domain, in order to effectively analyse a relational database two options are available.

1. Transform the relational database into flat dataset, i.e. all the information of the studied subject should exist in single table.
2. Use special relational data mining algorithms.

Each option has its own advantages and disadvantages. Transforming relational database into a flat dataset makes the data exploitable by all the traditional learning algorithms. The transforming process is usually done through a series of joining tables and aggregation functions. This process is known as the construction of the Universal Relationship (Codd, 1990). However, the transformation process could lead to information redundancy if not carefully designed and chosen.

The second option uses special relational data mining algorithms. "A relational data mining algorithm searches a language of relational patterns to find patterns that are valid in a given relational database" (Džeroski, 2010). Most of these algorithms come from the field of inductive logic programming (ILP) (Lavrac and Dzeroski, 1994; Muggleton et al., 1992): ILP has been concerned with finding patterns expressed as logic programs. The advantage of the relational data mining is that it can be applied directly to a relational database without transforming the relational database into a flat one. However, relational data mining methods are mostly dependent to the ILP algorithms which is not suitable to all kinds of problems. Moreover, most of the relational data mining algorithms are not highly scalable due to the computational expense of repeated joins (Han et al., 2012).

There are few works in the scientific literature that support feature selection methods in the context of relational database. Therefore, in the dissertation we have chosen to transform the relational database into a flat dataset so it can be compatible with the existing feature selection methods. We provide an efficient transformation approach that avoids information redundancy and information loss.

3.3.4 Data preprocessing

The appropriate format and accuracy of the data is an important issue for learning algorithms, as real-world data tends to be incomplete, noisy, and inconsistent. Data preprocessing includes data cleaning, data integration, data transformation (Han et al., 2012).

Although numerous methods of data preprocessing have been developed, data preprocessing remains an active area of research, due to the huge amount of inconsistent or dirty data and the complexity of the problem. Specially in the context of medical data, the problem of missing information is a very active and sensitive research area, because of the specificity of medical data (J.Cios and Moore, 2002). Data integration of heterogeneous medical data is another active research area. However, the nature of the medical data we treat in the dissertation does not suffer from these issues. The PMSI database does not contain much missing or inconsistent data, but it does contain a lot of numerical attributes in addition to the possibility to represent the attributes on different hierarchy levels. Therefore, in the dissertation we focus on data transformation methods before using them in any learning algorithm including feature selection methods. Data

transformation includes data discretization and creating hierarchy concepts to represent the attributes on the appropriate level of granularity.

Concerning data discretization "*Many Machine Learning (ML) algorithms are known to produce better models by discretizing continuous attributes*" (Kotsiantis and Kanellopoulos, 2006). In general, there are two kinds of discretization methods, supervised and unsupervised (Dougherty et al., 1995). Famous discretization methods are Entropy based methods (Kohavi and Sahami, 1996). Cluster analysis based methods are also popular (Chmielewski and Grzymala-Busse, 1996). Some other works in the domain include novel methods, such as the one proposed by (Rahman and Islam, 2016) who used a discretization technique called low frequency discretizer (LFD) that does not require any user input. However, to our knowledge, meaningful discretization is missing in the scientific literature where the meaning of the data is taken into consideration specially in the unsupervised discretization, except for (Vannucci and Colla, 2004).

Concerning the creation of hierarchy concept to represent the attributes, it is important to represent necessary background knowledge which controls the generalization process (Han et al., 1992). Using a concept hierarchy, the rules learned can be represented in terms of generalized concepts and stated in a simple and explicit form, which is desirable to most users. This is specially true in the case of categorical attributes, which have a finite number of distinct values, without any type of ordering among the values. Examples include geographic location, job category, and diagnoses type. There are several methods for the generation of concept hierarchies for categorical data, supervised and unsupervised described in (Han et al., 2012).

In the dissertation we investigate the possibilities to represent attributes, such as diagnoses and medical procedures on different level of hierarchies in order to increase the readability as well as to avoid the curse of dimensionality by decreasing the number of features.

3.3.5 Stability and robustness of feature selection methods

An important factor to consider when medical databases are used with feature selection methods is the stability and robustness of the features. Two types of features stability can be distinguished:

1. Stability of the features over feature selection method.

2. Stability of the features over training sets.

The first kind of stability is when obtaining different features over with different FS methods applied on the same dataset. This kind of instability is studied in the context of biomarker discovery from genomics data (Haury et al., 2011; Jovic et al., 2015). The paper concluded that the outputs of univariate methods seem to be more stable than to the multivariate methods. In addition, filter methods are more accurate compared to wrapper and embedded methods.

The second type of feature instability is obtaining different features over slightly different training sets when the same FS method is used. *"Stability of a feature selection algorithm can be viewed as the consistency of an algorithm to produce a consistent feature subset when new training samples are added or when some training samples are removed"* (Chandrashekar and Sahin, 2014). This problem is pointed out by (Dunne et al., 2002; Yang and Mao, 2011). They demonstrate examples of instabilities by running the feature selection algorithm multiple times and changing the training set by adding or removing some examples. If the algorithm produces a different subset in each run then the algorithm might not be reliable for feature selection.

For example, in (Dunne et al., 2002) the authors studied the instability of the wrapper methods, and proposed to use "*Hamming distance*" to measure the instability of features. They proposed to alleviate the instability by repeating the wrapper method multiple times and by retaining the frequent ones. (Somol and Novovi, 2010) investigated in a various feature stability measures, using these measures, a more robust features can be selected from different datasets. Another method used in the scientific literature is combining the results of different FS methods instead of choosing one particular FS method and accepting its outcome, such as (Yang and Mao, 2011; Yang et al., 2004). Bayesian averaging approaches are proposed in (Yeung et al., 2005). Boosting has been adapted to improve the robustness and stability of the final features (Ben-Dor et al., 2000; Dudoit et al., 2002). Overall, ensemble methods produce robust and stable features, the main disadvantage of these methods being the additional computational resources which in some case is not tolerable when huge datasets are used.

In the dissertation we focus on the second type of stability of feature selection over training set, since we plan to use feature selection after applying sampling method that can change the training set which can lead to instability to the features.

3.4 Conclusion

In this chapter, we presented the background of learning algorithms in general and the background of feature selection methods in particular. This study of related work helped us formulate the rationale for the technical choices made in the remainder of the thesis. The evaluation of feature selection methods is discussed, as well as the performance metrics used for this purpose. We presented some of the technical challenges related to the usage of feature selection method with medical databases and the related works to overcome these challenges. Our contribution addresses some of these challenges, namely the imbalance of datasets by providing a new stable and robust approach to select features. Moreover, the approach is interpretable so any non expert can understand the how the features are selected and the relation between the features are extracted.

Chapter 4

Encoding diagnoses: a state of the art

"The true sign of intelligence is not knowledge but imagination."

-Albert Einstein

Contents

4.1 Introduction

In this chapter we review different researches that propose aid in encoding diagnoses. Most of the researches concentrate on proposing automatic diagnosis coding. This problem falls under data prediction category and researchers address this problem in a variety of applications, such as marketing, e-business and other industrial sectors, but in in the medical domain, data prediction has specific constraints, since it deals with sensitive data, which is considered unique in terms of heterogeneity, privacy-sensitive, ethical, legal, and social issues (J.Cios and Moore, 2002). Therefore, various methods are proposed to overcome these constraints and to use medical data properly.

4.2 Encoding diagnoses data sources

In the scientific literature, encoding diagnoses is performed through different techniques according to the different types of sources used. We clearly distinguish two types of data sources used to predict diagnoses:

1. **Non-structured data** where the main sources are clinical reports, physician's interpretations, discharge letters and other medical documents that are usually written in free text and that are frequently used by coders to determine the medical code.
2. **Structured data** one of the important sources is PMSI database, which contains well formatted data concerning inpatient episodes.

4.2.1 Non-structured data

Automatic encoding of diagnosis based on non-structured data can be further classified into two types of methods:

1. Methods using previously coded examples.
2. Methods not using any previously encoded examples.

The first type of methods uses a database of previously encoded diagnoses in order to construct rules or learning model that predict the new diagnoses code. These methods are usually text classification methods. These methods consider each code as class label and use machine learning methods to build models that assign the diagnosis code.

Usually text classification methods use *Natural Language Processing* (NLP) methods as a first step of data analysis. NLP methods consist of translating free text into formal representation of features so that machines can understand the text and manipulate it. Afterwards, learning algorithms are often applied to extract coding knowledge.

Machine Learning techniques study features to produce an intelligent model that interprets these features and finds a logical relation in them in order to assign the prediction class (Collobert and Weston, 2008), consequently one of the problems is to determine which features could be extracted from the data to perform efficient learning. In the medical area, researchers extract feature matrices from medical reports and other non-structured medical sources from patient episodes. Next, machine learning methods are applied on these matrices in order to generate models that can predict a diagnosis code.

Different learning algorithms are used to tackle this prediction problem, such as the one proposed by (Farkas and Szarvas, 2008) which used Decision Trees to acquire rules and synonyms to assign codes. Other supervised learning methods are proposed, such as Naïve Bayes Classifiers (Okamoto et al., 2012; Pakhomov et al., 2006), Support Vector Machine (SVM) (Yan et al., 2010) and Scikit-Learn (Kavuluru et al., 2015). Likewise, regression methods are proposed by (Lita et al., 2008; Xu et al., 2007). Similarly, (Aronson et al., 2007; Erraguntla et al., 2012; Ruch et al., 2007) used unsupervised learning methods, such as K-Nearest Neighbours (KNN) to acquire the prediction rules.

The second type of methods do not use any previously encoded examples of coding, they directly map the sources into the corresponding diagnoses codes through knowledge base or word comparison.

Some methods use expert rules to assign encodings to the diagnoses. Researchers transform experts' coding knowledge into rules directly applied on the medical reports. An example for expert rules is proposed by (Goldstein et al., 2007). The authors used hand crafted rules applied directly on radiology reports. The rules aim to extract lexical

elements from radiology reports written in free text, lexical elements can be generated using semantic features to include negations, synonyms and uncertainty.

Semantic similarity between words or terms are estimated through knowledge-based methods and corpus-based methods. In the medical domain, "*SNOMED-CT*" and "*UMLS*" are the most famous knowledge bases used for semantics implementation. A semantic based work is proposed by (Pereira et al., 2006), the UMLS thesaurus is used to make automated MeSH-based indexing system that maps between prescription drug and the relevant ICD-10 codes. However, semantic similarity methods are highly dependent to the knowledge base, i.e. what works in the USA does not work in China and vice versa. For example, a Chinese study is published by (Ning et al., 2016). The authors used "*HowNet*" knowledge base which can only encode Chinese discharge letters to ICD-10 codes.

The results of both types of methods (semantic and expert rules) reach interesting prediction performances, for instance 88% F1 measure score in (Farkas and Szarvas, 2008). Nonetheless, the limit of these methods is the dependence to the quality of the text, the text language, the knowledge base used and other factors, which make these methods relevant only in their designed context. We search for more independent methods even if the quality of the results is not as much as the methods that use non-structured data.

4.2.2 Structured data

Few works in the literature used structured patient data for diagnosis prediction. Unlike non-structured methods, these methods are based only on previously encoded cases. In such cases, the data is mostly extracted from medical records, such as patient information (i.e. age, sex, length of stay), clinical information (i.e. prescription, medications) and other related medical data, such as medical procedures and diagnoses. The interesting study of (Lecornu et al., 2009) is based on statistical methods and probabilities. The authors focus on three types of medical data in order to estimate the probability of a diagnosis code. The first type is patient information (age, sex, length of stay), the second type is medical unit information and the third type is medical procedures. According to their study, diagnosis prediction is considered valid if it falls within the first 10 diagnoses ordered by probability score. The results of (Lecornu et al., 2009) show that medical procedures were the most informative input, whereas the patient information was the

least informative input. The authors report that better results could be achieved using all the inputs together by defining the right coefficient for each input.

The limit of probabilistic and statistical approaches is the sensibility of these methods with respect to the quality of the used data. In particular, these methods generate imperfect results when they are applied on imperfect data, missing data or erroneous codes. Data mining approaches are good alternative, since data preprocessing techniques can help reducing the impact of imperfect data (Han et al., 2012).

The authors of (Ferrao et al., 2013) propose to use well structured data extracted from electronic medical records and convert them to around 5000 features. They use different data mining algorithms in several steps including feature selection methods and various learning algorithms, such as Naïve Bayes and Decision Trees algorithms in (Ferrao et al., 2012), SVM in (Ferrao et al., 2013) and finally regression algorithms in (Ferrao et al., 2015), trying to assign codes during different periods of the inpatient episode. All the proposed algorithms gave about similar evaluation in terms of F1-measure but the results are still less effective than the F1-measure results reached by NLP techniques on radiology reports (Farkas and Szarvas, 2008; Goldstein et al., 2007).

In France, two studies used data mining techniques to tackle the problem of assigning medical codes to inpatient episodes (Djennaoui et al., 2015; Pinaire et al., 2015). These approaches used the diagnoses occurred in the previous inpatient episodes and constructed sequential patterns rules to predict a diagnosis code in the current patient episode. Two out of three diagnoses were successfully predicted using sequential patterns in (Djennaoui et al., 2015).

Although, these methods achieve acceptable rates of accuracy, they mainly avoided addressing some technical challenges related to medical databases, one of the important challenge being imbalanced datasets, which simply avoided addressing by choosing the few diagnoses that have balanced datasets. Another difficulty not sufficiently addressed or explained is the database transformation to a ML usable format. The use of feature selection methods is modest, mainly used to enhance the performance of the learning algorithms without involving them to increase the interpretability of the extracted results.

4.3 Conclusion

In the state of the art of encoding secondary diagnoses the emphasis is on providing automatic encoding of a few carefully selected diagnoses, very few of them proposing a general approach that encodes well on a wide range of diagnoses. Since concentrating only on one diagnosis is easier and it produces higher performance prediction models. Additionally, the reviewed methods did not consider providing the coder with useful information that helps to facilitate the encoding. In the dissertation, we emphasize on the interpretability of the provided predictions as well as independent method that can be generalised into large number of diagnosis encodings, in the maximum variation of contexts.

The application domain of the dissertation mandates the use of a structured medical database (PMSI), consequently we are interested in finding a general method that uses structured data input, taking into consideration all the technical challenges related to the medical database. Therefore, we used data structure similar to the data used in (Ferrao et al., 2012; Lecornu et al., 2009) specially when patient information is used. Concerning the data structure to represent diagnoses and medical procedures, we explore different level of hierarchies inspired from (Ning et al., 2016), more details on our approach is to be found in the contribution part of the dissertation.

Usually, non-structured data produces excessive amount of features. Therefore, feature selection methods are more common in the approaches that uses non-structured data compared to approaches use structured data. In the case of non-structured data, the features are extracted mostly from text. Features extracted from text usually describe the properties of the text, such as word count or word average. Consequently, these features are irrelevant to be presented to the coders in order to have coding assistance or to support an encoding decision made by a learning model. However, in the case of structured data, the features carry semantic meaning to the encoding process, such as diagnoses features or medical procedure features. Therefore, these features can be useful to be presented to the coders when a predication is made. In the dissertation the features are extracted from structured data and they have useful meaning. Consequently, we plan to show the features to the coders in order to justify and support the decision of a diagnosis prediction algorithm.

Part II

Contribution

Chapter 5

Application domain and field observation

"Attitude is a little thing that makes a big difference."

-Winston Churchill

Contents

5.1 Introduction

In this chapter, we observe the process of encoding diagnoses and we describe the thinking process the coders follow in order to encode diagnoses properly. Moreover, we observe the difficulties coders encounter in order to encode all the diagnoses occurring in each inpatient episode. Therefore, we identify at which part in the encoding process it is the best to intervene using intelligent informatics tools and to provide encoding help to the coders.

The main problem in encoding diagnoses is encoding all of them without missing any code. Unlike encoding the primary diagnosis which is not difficult to detect, encoding secondary diagnoses needs a lot of effort investigating all the possible signs that exist in all the possible sources of encodings.

5.2 Encoding observation

This section presents the preparation of the observation sessions organised in the hospital.

5.2.1 Observation preparation

In order to understand the ongoing thought process of the coders, we defined some elements before starting the observation. We defined the objectives and the characteristics of the observation based on the recommendation of (Taylor-Powell and Steele, 1996). The following characteristics define our observation sessions:

1. **Objective**: description of the thinking process of the coders while they encode inpatient episodes as well as observation of all the sources coders use to achieve efficient encodings.
2. **Who to observe**: specialist CIM-10 coders of diagnoses.
3. **What to observe**: the encoding of the diagnoses in the inpatient episode.

4. **What to note**: information concerning the inpatient episode as well as the observation session:
   (a) Inpatient episodes information:
      i. Personal information (age - gender).
      ii. Administrative information (length of stay - entry mode - discharge mode - diagnoses count - procedures count ...).
      iii. Medical information (diagnoses - medical procedures).
   (b) Observation session information:
      i. The used sources, such as discharge letters and reports.
      ii. Session's period.
      iii. Difficulties.
      iv. Impressions.
      v. Other.
5. **How to observe**:
   (a) Recording the session.
   (b) Taking notes using printed forms and checklists.
6. **Who observes**: By myself, a PhD student and the observations are validated by the supervisors of the thesis and the head of the medical information department.

The sessions are optimised to be efficient and to not waste the coders time. In order to achieve high efficiency of observation, the list below is prepared.

- Observation sheets and checklist are prepared in advance with the all the observation elements. An example of blank observation sheet is presented in the Appendix Figure B.1
- Coders consent is taken before recording the observation session.
- The sessions are recorded.
- The coders are asked to think aloud and to explain on what basis each diagnosis is encoded.

- The observer checks the checklist and takes notes of all the observed elements.
- The observation notes are verified later by listening the recorded session.

5.2.2 Observations summary

We organised observation sessions in the hospital of "*Centre Hospitalier Intercommunal de Castres Mazamet*" a local hospital in the town of Castres. We observed two specialist coders. There was 30 pre-classified inpatient episodes according to two primary diagnoses, the first primary diagnosis is related to lung disease which can be coded by different variations in CIM-10 **J15-J18-J69** codes, the second observed primary diagnosis is related to delirium disease and all its variations which can be coded in CIM-10 by **F05** code.

Some statistics on the observation sessions are given in the following:

- The observations count is 30 observations.
- The mean observation duration is 10 minutes.
- The mean count of diagnoses is 10 diagnoses.
- The mean count of medical procedures is 3.
- The mean age of the patients 86 years old.
- The male count is 18, the female count is 12.
- The lung disease count is 18.
- The delirium disease count is 12.

The detailed observation notes are presented in the Appendix Table B.1.

5.2.3 Encoding description

The main sources of diagnosis encoding during a coding session are the discharge letter, the anesthesia sheet and the operative reports. These sources represent reports written in free text that summaries the patient status and summaries the details of a surgery etc., each care service includes the patient's received care during the inpatient episode. These

sources usually do not follow a standard form, depending on the service care, they could be well or less organized making coding diagnoses vary in difficulty.

Essentially, these main sources contain the physician's remarks. Moreover, they might contain antecedent diseases, abnormal behaviour, nutrition habit abnormalities, special equipment used as well as any other remarks made by nurses.

First step the coders follow is analysing the main encoding sources. The obvious diagnoses are then encoded, which are usually extracted from the physician's remarks and some antecedent diseases. Afterwards, coders look into signs for potential encodings. If some signs are detected, coders register the potential diagnoses in a list of codes. These potential diagnoses are verified and confirmed later by further research in other secondary sources, such as lab reports, digital medical records and radio images. In some cases, coders search for the right specification of the diagnosis and in other cases, they search for criteria to confirm certain diagnosis.

For example, during an encoding session of an 86 years old female patient, entered by the urgency service of the hospital for abnormal coughing suspecting *Inhalational pneumonia* problem, the coders follow these steps:

First, the coder looks at the antecedents where some diseases are clearly mentioned, such as "*Hemiplegia stroke*=G811", "*Bedridden*=R26" and others. These diagnoses are encoded directly without any further verifications. Moreover, the coder notices that a team of palliative care has visited the patient and that the idea of life ending has been mentioned in the discharge letter therefore the coder encodes "*palliative care*=Z515" diagnosis, later some other signs confirmed this diagnoses, such as the visit of the psychologist doctor.

In the next step, the coder prepares a list of potential diagnoses to encode based on signs extracted from the discharge letter. For instance, the coder said "*We remark here the usage of a bedsore mattress we will verify later from the medical record if the patient has a bedsore disease*" therefore the coder adds bedsore diagnosis in the list of potential diagnoses. Another example of a sign that makes the coder suspect of potential diagnosis is the presence of "*Leukocytosis*" which is a sign of "*Inflammatory syndrome*". This diagnosis needs to satisfy some criteria in order to be encoded, such as temperature or the level of white blood cells. Furthermore, the coder finds a sign of *Malnutrition* through eating

disorder. *Malnutrition* has different severity levels that should be determined through some criteria, such as height, weight and "*Albumine*" level.

Finally, the coder finds six potential diagnoses (*Bedsore - Multi-resistant germ - Inflammatory syndrome - Respiratory failure - Malnutrition - Cognitive disorder*). To confirm or to exclude a diagnosis, the coder needs to go further and search for evidences and criteria. The main verification sources are medical records, lab tests and radio images. After the verification and the research the coder excludes *Bedsore* and *Respiratory failure* due to lack of evidence in the sources. The *Malnutrition* is confirmed through weight and height. Moreover, the mild severity level is specified through *Albumine* level test. Similarly, the *Inflammatory syndrome* is confirmed through lab tests.

The procedure is summarised into two principal steps:

1. The direct coding when there is no need to search for further evidence.
2. The potential coding when further evidence are required and criteria to be satisfied.

The encoding procedure is presented in the Figure 5.1.

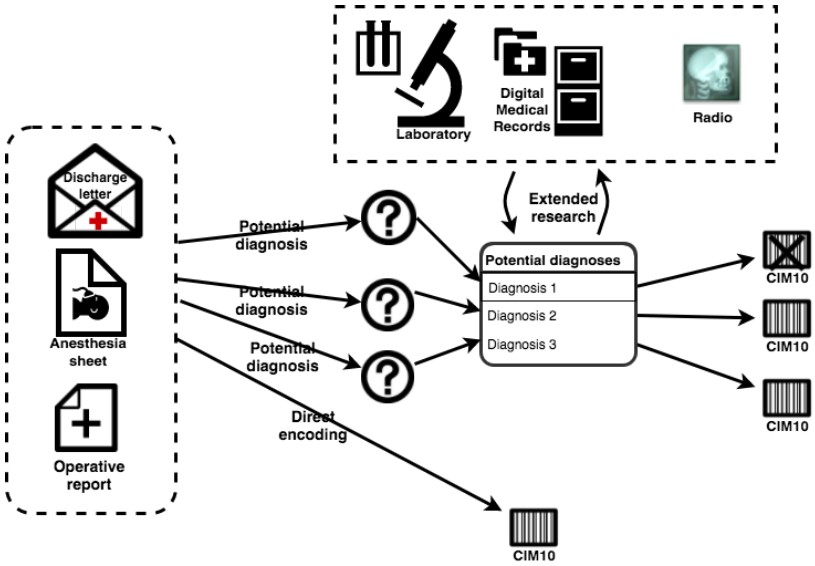


Fig. 5.1 The procedure followed to encode diagnoses

5.3 Proposition to enhance the procedure of encoding diagnoses

The main concern of encoding diagnoses is to avoid missing any diagnosis. The discharge letter is not perfect and each care service provides the coders with different versions of discharge letters. The objective we are looking for is to prevent missing diagnoses codes. For that, we propose to modify the procedure and add an intelligent tool that completes the list of potential diagnoses. The tool uses previously encoded inpatient episodes stored in the PMSI database in order to build a model that suggests missing diagnoses and completes the list of potential diagnoses. The modified procedure is shown in the Figure 5.2 Each suggested diagnosis is supported by providing the information pieces used to make the decision as well as the relation between these pieces.

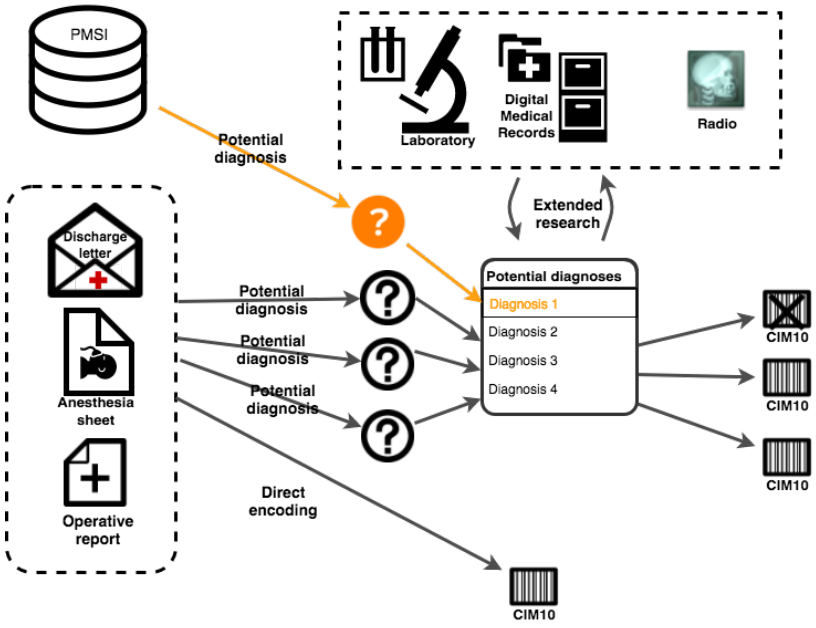


Fig. 5.2 The proposed contribution to encode diagnoses

We propose completing the potential encoding at two different stages:

- At the beginning of an encoding session, some information is already coded prior to diagnosis encoding session, such as age, gender, medical procedure, entry and exit mode. We explore the possibility to suggest diagnoses based on the available information prior to diagnoses encoding.

- At the end of encoding session, we explore the possibility to complete diagnoses that have high probability rates of association with the current inpatient episode based on all the available information posterior to encoding.

5.4 Conclusion

In this chapter, the coding environment is described, the thinking process of the coders is modelled. The difficulties of encoding secondary diagnoses from discharge letter are highlighted, emphasising the fact that encoding sources are not perfect and these sources do not have all the signs that permits to encode all the diagnoses properly. We presented our proposition to modify the diagnoses encoding process in order to guide the coders verify all the possible encodings of secondary diagnoses.

Our proposition does not replace the coders job but it supports them to inspect further some secondary diagnoses that are likely to be present in the inpatient episode.

Chapter 6

Medical database (PMSI) preparation for Feature Selection

"Science is not only a disciple of reason but, also, one of romance and passion."

-**Stephen Hawking**

Contents

6.1 Introduction

In this chapter we explain how to prepare a medical database, such as PMSI database for the usage of Machine Learning methods.

A lot of preparation is required to extract knowledge from medical databases using ML methods. Medical databases are usually imbalanced which means they contain unequally distributed classification examples (positive and negative examples). Moreover, medical databases are usually in relational database format and each instance has many records. Furthermore, medical databases contain attributes format difficult to manage using ML methods. Consequently, we propose an approach and a sequence of steps to prepare a medical database properly, transforming the database into single flat table with a single record for each instance. Moreover, the database attributes are prepared by discretizing the continuous attributes and by choosing the best hierarchy level to represent attributes. The preparation also includes balancing the database to address the distribution of the classification examples. These preparations play big role to make the medical databases more exploitable and effectively analysed by ML methods.

In order to evaluate the database preparation we selected some representative diagnoses under a physician's supervision and used the prepared database to predict these representative diagnoses.

The rest of this chapter is organised as follows: the second section shows the steps followed to prepare the database. The third section presents the evaluations of the proposed approach. The fourth section provides a discussion of the results and the seventh section concludes the chapter.

6.2 The PMSI database preparation

The PMSI database is one of the important sources of medical data, since it documents all the inpatient episodes across the country of France. It provides a detailed information of the inpatient episodes through standard codes. It provides useful information to encode diagnoses if it is exploited retrospectively. The detailed description of PMSI is provided in the state of the art Chapter 2. Therefore, the objective of this section is to make the

PMSI database exploitable by ML methods, especially by feature selection methods. The proposed approach is shown in the Figure 6.1, it is inspired from a procedure to extract knowledge from databases presented by Fayad (Fayyad and Uthurusamy, 1996).

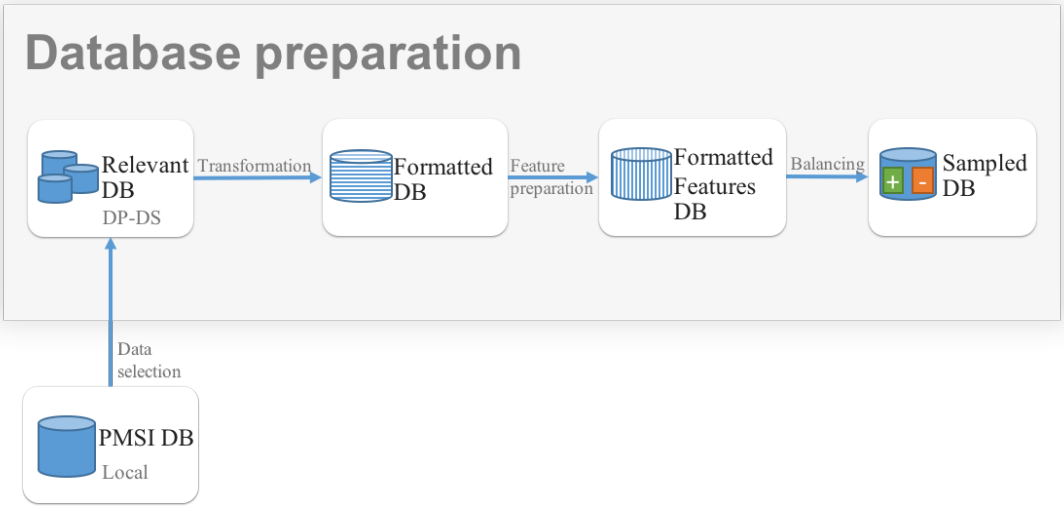


Fig. 6.1 The PMSI database preparation for Machine Learning analysis

1. **Data selection**: this step develops an understanding of the application domain by capturing relevant prior knowledge from the end users perspective (in our case the coders) and by identifying the application objectives i.e., selecting proper datasets representative of the targeted problem based on the domain knowledge and based on the help of the experts. For example, in our case we identified difficult secondary diagnoses to encode, in addition to original combination of secondary diagnoses and primary diagnoses that do not occur usually together.

2. **Dataset transformation**: this step projects the selected data into an appropriate format that ML algorithms work on. The ideal format that most of the ML algorithms prefer is (single record case) where each instance is represented by a feature vector. The representation level of the features is inspected whenever multi level representation is possible. For example, the diagnoses can be represented with different hierarchy levels therefore we inspected the best hierarchy level to represent in order to produce the best results.

3. **Feature preparation**: this step analyses the numeric features and discretizes them.

4. **Dataset balancing**: the last step balances the dataset in order to have fair representation of classification classes.

The used PMSI database is extracted from "*Centre Hospitalier Intercommunal de Castres Mazamet*" hospital. It contains around 90,000 inpatient episodes between 2011 and 2014. In order to facilitate the management of the PMSI content we use the classification of the ATIH[1] as reported in the state of art (Figure 6.2).

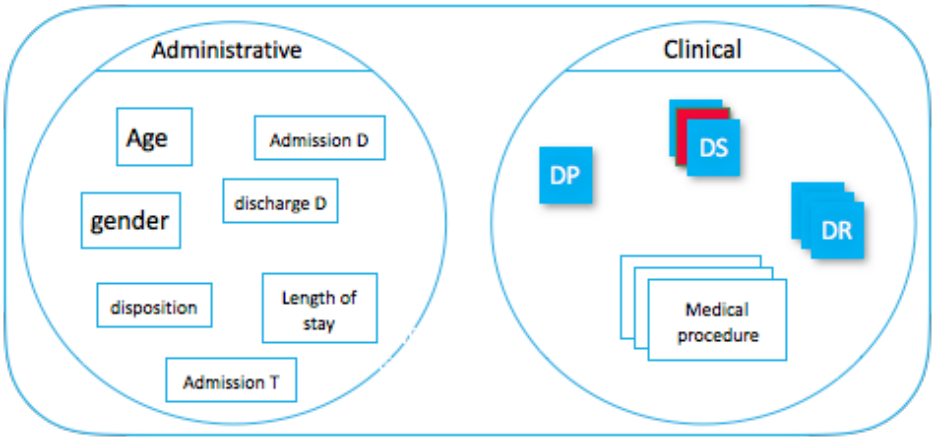


Fig. 6.2 The PMSI information classification by ATIH

6.2.1 Data selection

The first step of our approach identifies the relevant data that well expresses the studied problem with the help of the domain expert. In the case of encoding secondary diagnoses within PMSI domain, we propose to select interesting diagnoses to study in two stages. In the first stage we identify the interesting **S**econdary **D**iagnoses (**DS**) that are difficult to encode, whereas in the second stage we identify interesting **P**rimary **D**iagnoses (**DP**) occurred with the DSs in order to choose more targetted datasets that eliminates irrelevant instances.

6.2.1.1 Interesting Secondary Diagnoses (DS)

With the help of the physician in charge of the **M**edical **I**nformation **D**epartment (DIM) in the '*Centre Hospitalier Intercommunal de Castres Mazamet*' hospital, we identified the interesting and the frequent secondary diagnoses that are difficult to detect as they are usually not well described across the medical sources. Eight DS are retained as listed in Table 6.1.

[1] http://www.atih.sante.fr/mco/presentation

Table 6.1 The studied secondary diagnoses.

ICD-10 codes	**Label**	Count in PMSI DB	**Included specific diagnoses CIM10 codes**
L89	Pressure ulcer	1131	L89.0(stage I Pressure ulcer) - L89.1(stage II Pressure ulcer) - L89.2(stage II Pressure ulcer) - L89.3(stage IV Pressure ulcer) - L89.9(Pressure ulcer without precision)
J96	Respiratory failure	4166	J96.0(Diagnosis Acute respiratory failure) - J96.1(Diagnosis Chronic respiratory failure) -J96.9(Respiratory Failure, Unspecified)
B96	Bacterial agents, such as Mycoplasma and pneumoniae	6514	B96.0(Mycoplasma Pneumoniae) - B96.1(Klebsiella Pneumoniae) - B96.2(Escherichia Coli) - B96.3(Hemophilus Influenzae) - B96.4(Proteus) - B96.5(Pseudomonas) - B96.6(Bacteroides Fragilis) - B96.7(Clostridium Perfringens) - B96.8(Other Bacterial Agents)
T81	Complications of procedures	1150	T81.0(Hemorrhage and hematoma complication procedure) - T81.1(Postprocedural Shock) - T81.2(Accidental perforation and tearing during a procedure)- T81.3(Disruption Of Wound) - T81.4(Infection Following A Procedure) - T81.5(Comp Of Foreign Body Acc Left In Body Following Procedure) - T81.6(Comp Of Foreign Body Acc Left In Body Following Procedure) - T81.7(Vascular Complications Following A Procedure) - T81.8(Other Complications Of Procedures) - T81.9(Unspecified Complication Of Procedure)
R29	Nervous and musculoskeletal systems, such as (Neonatal tetany)	1596	R29.0(Tetany) - R29.1(Meningismus) - R29.2(Abnormal Reflex) - R29.3(Abnormal Posture) - R29.4(Clicking Hip) - R29.6(Repeated Falls) - R298(Other Symptoms And Signs Involving The Nervous And Musculoskeletal Systems)
R26	Abnormalities of gait and mobility	2378	R26.0(Ataxic Gait) - R26.1(Paralytic Gait) - R26.2(Difficulty In Walking) - R26.3(Immobility) - R26.8(Other Abnormalities Of Gait And Mobility)
E66	Overweight and obesity	5453	E66.0 (Obesity due to excess calories) - E66.1(Diagnosis Drug-induced obesity) - E66.2(obesity with alveolar hypoventilation) - E66.8 (other obesities) - E66.9 (Diagnosis Obesity, Unspecified)
E44	Malnutrition	2144	E44.0 (Moderate malnutrition)-E44.1 (Mild malnutrition)

For example, malnutrition E44 and obesity E66 are frequently not well encoded in a lot of inpatient episodes due to different reasons which can impact the hospital's budget. (Potignon et al., 2010)

The diagnoses are encoded using CIM10 which consists of two parts "category" and "precision". In this dissertation we used the category i.e. the first three characters to indicate the diagnosis. The category is sufficient to identify the diagnosis, the second part of the encoding, the precision is complementary, it can be identified easily if the category is known. Table 6.1 presents all the chosen diagnoses CIM10 codes in addition to their labels and all the specifications of each diagnosis.

6.2.1.2 Interesting DP-DS couples

In the second stage of selecting data we propose to use more targeted datasets as the whole database holds a lot of non useful cases that cause irrelevant information to the studied case. Therefore, we propose to work on targeted PMSI dataset where one DP appears with the selected interesting DSs. It is realistic, since the DP is supposed to be easily known by the coders in the most of the cases. In other words with the hypothesis of fixing the DP, more focused dataset can be obtained that helps to facilitate the analysis. Hence, for each selected DS, the most frequent DPs are queried and a dataset is built for each DP. In this dissertation, we had 8 interesting DS, therefore 80 datasets are extracted and analysed i.e. ten for each DS.

In the Tables 6.2 6.4 6.3 6.5 6.6 6.7 6.8 6.9 ten most frequent DPs for each DS are presented with some details, the first column contains the label of the DP, the second column shows the CIM10 code, the third column presents the count of the DP in the database, the fourth column shows the number of the studied DS occurred during the same inpatient episode (positive examples' count), the fifth column shows the number of the episodes that contain the DP, but does not contain the studied DS (negative examples' count) and finally the last column shows the ratio of the positive examples to the negative examples of the studied DS in the episodes that contains the DP.

After the first analysis of the datasets we have consulted the physician to classify the extracted DP-DS couples and identify the interesting ones. The physician identified three categories of DP-DS couples:

1. **Trivial couples**: These DS-DP couples are usually linked together and they do not need any effort from the coders to identify the DS when the DP occurs.

For example, in the case of a patient who entered an inpatient episode with abnormality of breathing (R06) such as sneezing, it is easy to conclude that the patient has respiratory failure (J96). Therefore, the couple DP-DS (**R06-J96**) is trivial to encode. Other obvious (DP-DS) couples are (Medical care encounter **Z51-L89** Pressure Ulcer; respiratory failure **J96-J96** respiratory failure; Acute pyelonephritis **N10-B96** Bacterial agents; Fever B96 Bacterial agents; Chronic obstructive pulmonary disease **J44-B96** Bacterial agents; Pneumonia **J18-B96** Bacterial agents; Inflammatory diseases of prostate **N41-B96** Bacterial agents).
The Trivial couples are represented in red color background in the tables.

2. **Original couples**: These DS-DP couples represent interesting arrangements, they are unusual to occur together. Therefore, more attention is needed by the coders in order to encode the DS.

   For example, it not usual for a patient that have pneumonia (**J18**) as primary diagnosis to suffer from malnutrition (**E44**), which makes the **J18-E44** original combination that does not occur very often. Therefore, encoding E44 requires more attention from the coders in order to detect E44 and encode it properly. Other original couples are (Abdominal and pelvic pain **R10-E66** Obesity; Cholelithiasis **K80-E66** Obesity; Atrial fibrillation and flutter **I48-E66** Obesity; Heart failure **I50-R26** Abnormalities of gait and mobility; Bacterial pneumonia **J15-R26** Abnormalities of gait and mobility; Acute bronchitis **J20-R26** Abnormalities of gait and mobility; Fever **R50-R26** Abnormalities of gait and mobility; Pneumonitis **J69-R26** Abnormalities of gait and mobility; Delirium **F05-R26** Abnormalities of gait and mobility; Fracture de femur **S72-L89** Pressure Ulcer; Bacterial pneumonia **J15-E44** Malnutrition; Pneumonia **J18-E44** Malnutrition; Delirium **F05-E44** Malnutrition; Nervous and musculoskeletal system symptoms **R29-E44** Malnutrition; cognitive function and awareness symptoms **R41-E44** Malnutrition).
   The original couples are represented in green color background in the tables.

3. **Frequent couples**: These DS-DP couples occur often together but they are not as easy as the obvious couples to encode.

   For example, it is not surprising to have Pressure ulcer L89 as DS in a patient who has Pneumonia as DP. The remaining diagnoses couples of the 80 couples are regular couples.
   The frequent couples have no color background in the tables.

Table 6.2 PMSI information about L89 diagnosis

L89: Pressure Ulcer

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Malaise and fatigue	R53	3262	117	3145	4%
Abnormalities of breathing	R06	2398	58	2340	2%
Heart failure	I50	2428	56	2372	2%
Encounter of medical care	Z51	3430	55	3375	2%
Bacterial pneumonia	J15	694	35	659	5%
Pneumonia	J18	1201	34	1167	3%
Fever	R50	1184	34	1150	3%
Respiratory failure	J96	877	34	843	4%
Acute bronchitis	J20	879	31	848	4%
Fracture of femur	S72	1110	25	1085	2%

Table 6.3 PMSI information about J96 diagnosis

J96: Respiratory failure

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Abnormalities of breathing	R06	2398	557	1841	23%
Respiratory failure	J96	877	395	482	45%
Heart failure	I50	2428	385	2043	16%
chronic obstructive pulmonary disease	J44	759	333	426	44%
Pneumonia	J18	1201	272	929	23%
Malaise and fatigue	R53	3262	173	3089	5%
Acute bronchitis	J20	879	148	731	17%
Bacterial pneumonia	J15	694	124	570	18%
Encounter of medical care	Z51	3430	90	3340	3%
Pneumonitis	J69	272	72	200	26%

Table 6.4 PMSI information about B96 diagnosis

B96: Bacterial agents

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Malaise and fatigue	R53	3262	377	2885	12%
Acute pyelonephritis	N10	435	351	84	81%
Abnormalities of breathing	R06	2398	251	2147	10%
Heart failure	I50	2428	231	2197	10%
Fever	R50	1184	230	954	19%
Abdominal and pelvic pain	R10	2650	200	2450	8%
chronic obstructive pulmonary disease	J44	759	149	610	20%
Respiratory failure	J96	877	138	739	16%
Pneumonia	J18	1201	137	1064	11%
Inflammatory diseases of prostate	N41	220	102	118	46%

Table 6.5 PMSI information about T81 diagnosis

T81: Complications of procedures

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Encounter for other post-procedural aftercare	Z48	771	76	695	10%
Malignant neoplasm of colon	C18	261	25	236	10%
Abdominal and pelvic pain	R10	2650	24	2626	1%
Cutaneous abscess, furuncle and carbuncle	L02	209	23	186	11%
Fracture of femur	S72	1110	22	1088	2%
Cholelithiasis	K80	1114	21	1093	2%
Paralytic ileus and intestinal obstruction without hernia	K56	632	20	612	3%
Peritonitis	K65	60	19	41	32%
Encounter for attention to artificial openings	Z43	204	15	189	7%
Intestine	K63	159	10	149	6%

In the dissertation most of the results are shown on the original DP-DS couples (R10-E66; K80-E66; I48-E66; I50-R26; J15-R26; J20-R26; R50-R26; J69-R26; F05-R26; S72-L89; J15-E44; J18-E44; F05-E44; R29-E44; R41-E44) and it can be generalised on any diagnoses couple DP-DS.

Table 6.6 PMSI information about R26 diagnosis

R26: Abnormalities of gait and mobility

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Malaise and fatigue	R53	3262	276	2986	8%
Abnormalities of breathing	R06	2398	151	2247	6%
Heart failure	I50	2428	104	2324	4%
Acute bronchitis	J20	879	88	791	10%
Pneumonitis	J69	272	80	192	29%
Delirium	F05	540	73	467	14%
Bacterial pneumonia	J15	694	70	624	10%
Pneumonia	J18	1201	70	1131	6%
Fever	R50	1184	68	1116	6%
Encounter of medical care	Z51	3430	62	3368	2%

Table 6.7 PMSI information about R29 diagnosis

R29: symptoms and signs involving the nervous and musculoskeletal systems

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Malaise and fatigue	R53	3262	135	3127	4%
Delirium	F05	540	89	451	16%
Intracranial injury	S06	1954	88	1866	5%
Symptoms and signs involving the nervous and musculoskeletal systems	R29	446	87	359	20%
Symptoms and signs involving cognitive functions and awareness	R41	669	61	608	9%
Hypotension	I95	730	33	697	5%
Pain, unspecified	R52	768	33	735	4%
Heart failure	I50	2428	32	2396	1%
Fracture of femur	S72	1110	24	1086	2%
Cerebral infarction	I63	895	23	872	3%

Table 6.8 PMSI information about E44 diagnosis

E44: Protein-calorie malnutrition

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Malaise and fatigue	R53	3262	272	2990	8%
Delirium	F05	540	137	403	25%
Heart failure	I50	2428	96	2332	4%
Abnormalities of breathing	R06	2398	93	2305	4%
Symptoms and signs involving the nervous and musculoskeletal systems	R29	446	76	370	17%
Pneumonia	J18	1201	72	1129	6%
Symptoms and signs involving cognitive functions and awareness	R41	669	67	602	10%
Abdominal and pelvic pain	R10	2650	57	2593	2%
chronic obstructive pulmonary disease	J44	759	45	714	6%
Bacterial pneumonia	J15	694	35	659	5%

Table 6.9 PMSI information about E66 diagnosis

E66: Overweight and obesity the nervous and musculoskeletal systems

DP Label	DP code	DP count	Pos.	Neg.	Perc.
Heart failure	I50	2428	229	2199	9%
Abnormalities of breathing	R06	2398	148	2250	6%
Pain in throat and chest	R07	1739	135	1604	8%
Malaise and fatigue	R53	3262	110	3152	3%
Encounter for other post-procedural aftercare	Z48	771	98	673	13%
Abdominal and pelvic pain	R10	2650	95	2555	4%
Type 2 diabetes mellitus	E11	336	95	241	28%
Respiratory failure	J96	877	93	784	11%
Atrial fibrillation and flutter	I48	961	91	870	9%
Cholelithiasis	K80	1114	75	1039	7%

6.2.2 Dataset transformation

Generally, most of the medical databases are relational databases, whereas in ML domain, in order to effectively analyse a dataset it should be in one table, i.e. all the information of the studied subject should exist in a flat database (Han et al., 2012). Another issue of a relational database is that it suffers from redundancy of information i.e. *multiple record* to describe an instance. Finally, the last issue treated in this section is the excessive number of features in the medical databases that limits the performance of the ML methods.

Therefore, in this step, the selected datasets are transformed to a format that most of the ML algorithms work on. The transformation consists of converting the dataset into flat table with *single record* instance representation where each instance is represented with a vector of features. Finally, the transformation decreases the excessive number of features by representing them on different hierarchies and choosing the best granularity level whenever possible.

6.2.2.1 Feature construction

Feature construction aims to project a relational databases with *multiple record* instances into a flat format with a single record instance.

Relational model databases are organized in one or more tables of columns and rows, with a unique key identifying each row. Rows are also called records or tuples. Each table represents one entity, such as "diagnosis" or "medical procedure". The columns represent attributes or features to the entity, such as "CIM-10 code" or "level" are attributes to the "diagnosis" entity.

In order to have flat dataset, *the first stage* of the feature construction step is to query the intended data by joining two or more tables.

However, a flat table can suffer from redundancy of information i.e. *multiple record* instances to describe a subject. For example, if we have an inpatient episode with two diagnoses stored in a relational database, the episode is represented with two instances in a joined table (Table 6.10), whereas, in a *single record* instance dataset, an instance is represented in one record using a vector of features, such as in (Table 6.11). Table 6.11 represents a vector with the same two records in the (Table 6.10).

In ML very few algorithms are adapted to work properly using *multiple record* instances datasets. Most of the ML algorithms prefer to work with a *single record* instance in order to avoid bias to the instances that have multiple records (Han et al., 2012). *Single record* instance is usually represented with a vector of features $(f_1, f_2, \ldots, f_n)$, where the features are either binary (i.e., $f_i \in$ true, false), numerical (i.e., $f_i \in$ R), or nominal (i.e., $f_i \in$ S, where S is a finite set of symbols).

Therefore, *the second stage* in feature construction is transforming multiple instance dataset into *single record* instance dataset where each instance is represented with vector of features.

We propose to transform the dataset by targeting the features that cause multiple records for the same instance. These features can be broken down into dummy features. For example, if a feature has ten values, it can be broken down into ten features each feature represents a boolean value indicating the presence or the absence of the value.

We distinguish two types of features based on the need of transformation:

- **Simple record**: These features occur once per inpatient episode instance. Therefore, these features can be used directly in the dataset without any transformation. In the case of the PMSI database, all the administrative information is considered as simple features as it occurs once per inpatient episode. These features include the age, the gender, the length of stay, the patient admission type, the patient discharge status, the time interval between the admission date and the date of the first medical procedure, the transfer count between medical units during the inpatient stay, the medical procedures count, the season of the admission and the previous hospitalisations count.
- **Multiple record**: These features occur multiple times per inpatient episode instance therefore they need to be broken down into multiple features with the values of the original feature. These features can be represented on different level of granularity. More details on this are presented in the Section 6.2.2.2. In the case of the PMSI database, the medical information including all the diagnoses and medical procedures are multiple record features and they need to be broken down into multiple features.

In summary, the feature contraction in the PMSI database, whose structure is shown in the (Appendix C), has two stages.

Table 6.10 Example of an inpatient episode that has two rows in a relational database

Inpatient episode id	Age	Diagnosis CIM-10 code
1	80	J96
1	80	B96

Table 6.11 Example of an inpatient episode that has two diagnoses expressed in a record

Inpatient episode id	Age	J96	B96	L89	..
1	80	1	1	0	..

- The first stage is transforming the relational database into flat datasets by joining all the tables that contain selected (DP-DS) couples.
- The second stage is transforming the multiple record instance features into single record instance with a vector of features.

However, in the PMSI domain the features can occur more than once for each inpatient episode. Therefore, in order to avoid multiple records for the same inpatient episode we broke down the features into their categorical values. In other words, a feature that had 100 values has been transformed into 100 features, where the value of each transformed feature is a boolean, indicating the presence or the absence of the feature in the inpatient episode. For example, in medical information we have two main features, the diagnosis and the medical procedure. The diagnosis has 2,049 possible values, since it is encoded using CIM-10 codes. The medical procedure has around 7,583 codes classified under 19 chapters. Therefore, if we break down these two categorical features into boolean features we will have 2,049+7,583 = 9,632 features which are difficult to manage. It can cause high memory consumption and it takes a lot of time to be analysed. In addition, (Sebban et al., 2000) confirms that excessive number of features does not yield necessarily to good results for ML algorithms.

6.2.2.2 Features representation

To manage and to limit the number of features we propose reducing the dimension of the features by representing the values of the features in this category using different

granularity and hierarchy levels. Two main features are concerned: medical procedures and diagnoses.

- Concerning medical procedure features, we used the highest level of hierarchy chapters defined in the CCAM to represent 19 features, each feature indicating whether one or many medical procedures in the corresponding chapter have occurred during the inpatient episode. The choice of this level of granularity is motivated by coders recommendations. The coders do not look at the exact medical procedure but they consider the nature of the procedure which can be obtained by the chapter of the medical procedure.
- Concerning diagnosis features, 145 diagnoses categories are available to represent these features. These categories can be organised into two hierarchical granularity levels. **(1) Coarse level** granularity which contains 19 chapters of diagnoses classification and **(2) Fine level** granularity which contains 126 specific chapters of diagnoses classification. These two levels of diagnosis granularity were recommended by the physician in charge of Medical Information Department (DIM), since the coders understand these classification categories and can relate them to find the right encoding of a diagnosis.

6.2.3 Dataset feature processing

"Many Machine Learning (ML) algorithms are known to produce better models by discretizing continuous attributes" (Kotsiantis and Kanellopoulos, 2006). In the dissertation we use ML algorithms either to select feature or to build a classification model in order to evaluate the extracted features, more details are provided on the evaluation in the Section 7.2.1.

In the PMSI database, there are two kinds of data:

1. Numeric (continuous or discrete) such as age, length of stay.
2. Categorical, such as gender, admission type, discharge type.

Some ML methods are more efficient using discrete values, for instance Decision Trees (DT), Naive Bayes (NB) algorithms are more efficient with categorical values, if

the values are numeric then they are discretized prior to build the model (Kotsiantis and Kanellopoulos, 2006). In the scientific literature several methods are proposed to discretize numeric values into categorical. For instance, "binning" is an unsupervised method which discretizes the numerical values either into equal-interval binnings or into equal-frequency binnings. Supervised discretization methods, such as "entropy-based", measure the information gain to the class and split the intervals recursively (Witten and Frank, 2005). Although these methods could generate a model with good performance, the proposed intervals lack clarity in terms of interpretability. In a first test we performed without treating this problem, ages for example were splitted into the following intervals (>6),([7-12]),([13-30]),([31-40]),(<40) such intervals could not really make sense for medical interpretation, especially when it is conducted on the entire database not on the diagnoses couple DP-DS. It is important in the medical domain to help the physicians to interpret the results. Therefore, with the help of the experts in the domain we retained three intervals that could be helpful in the encoding process applied. One interval contains the average population. The second interval contains below average population. The third interval contains over average population. Therefore, the second and the third intervals contain the extreme population.
These intervals are calculated on the filtered datasets, i.e. on the diagnosis couple (DP-DS) in order to target specific problem and to avoid generalised intervals calculated on the entire database. Therefore, these intervals help the experts work better by better identifying the extreme cases compared to other discretization methods that choose intervals in order to maximise the performance of the ML algorithm.

In order to verify the feasibility of feature discretization into the proposed intervals ("**Below**", "**Mean**", "**Over**"), the distribution of the continuous features are studied in order to find out the best values to cut the intervals. The studied statistic measures are the min, the max, the average and the median.
Therefore, studying the features consists of two parts:

- The distribution of the continuous features.
- The statistical measures of the continuous features.

In the case of the PMSI database, some new aggregated features are added to the study, these features do not exist by default in the PMSI database but they are helpful for the future analysis and they are often used in the hospital for activity reporting. The new

aggregated features are diagnoses count, medical procedures count, previous inpatient episodes count. Therefore, we study these new aggregated features in addition to the existing numeric features: age, length of stay and sessions count.

In this section, we study the datasets that correspond to each diagnoses couple (DS-DP).

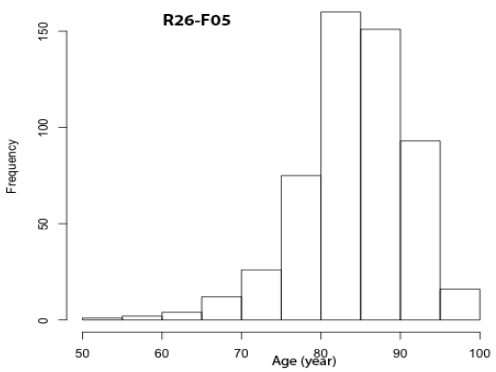
6.2.3.1 The distribution of the features

In the first part of the feature analysis, in order to find out the distribution and the population of the numeric features, the histograms of the attributes are plotted and the distribution of the features are stated by the form of the histogram. The results show that the distributions of a feature across different diagnoses have similar shapes, therefore we chose one diagnosis couple as illustration example. The DP-DS couple is "*Delirium*" as DP and "*Abnormalities of gait and mobility*" as DS" to present the histograms of the features in the (Figure 6.3).

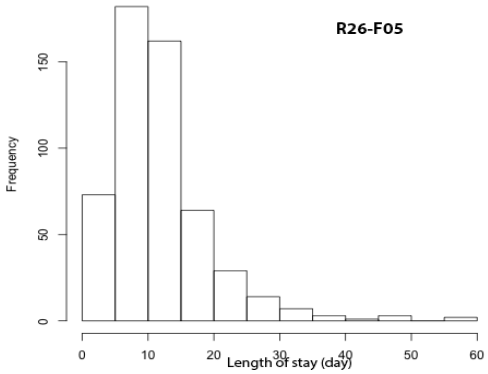
The histograms of the "Diagnoses count", "Medical procedures count" and "Length of stay" have normal distribution with skew to the left, the histogram of "Age" has also normal distribution but skewed to the right. The "Sessions count" and "Previous inpatient count" do not follow a normal distribution they have exponential distributions.

The distributions are skewed to the left or to the right because of the location of the normal values are close to the edge of the interval of possible values. For example, in the length of stay the possible values are in the interval [0-n] and the normal value is having short length of stay, consequently the distribution of the values are skewed to the left. Similarly, medical procedures count, diagnoses count are skewed to the left. The only feature skewed to the right is the age where the normal values are the high values, which correspond to the old people.

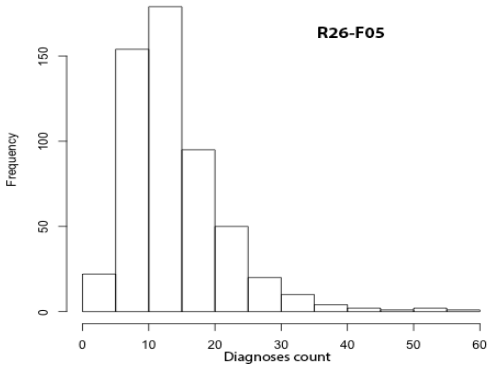
Concerning the "Sessions count", the normal value is zero. The normal value of the "previous inpatient episode count" is one. Therefore most of the values are concentrated on these values producing exponential distribution for these two features.



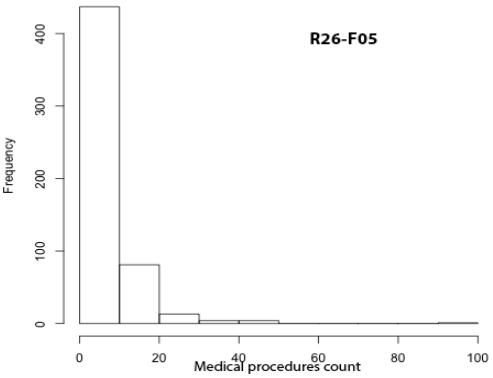
(a) Histogram of age



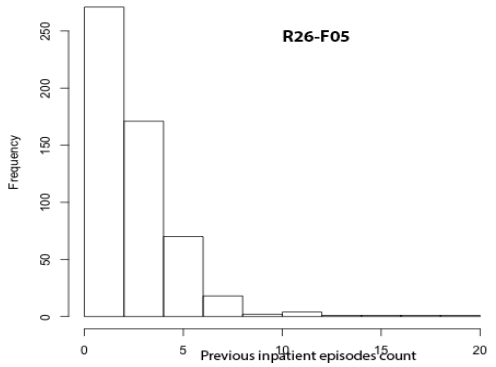
(b) Histogram of length of stay



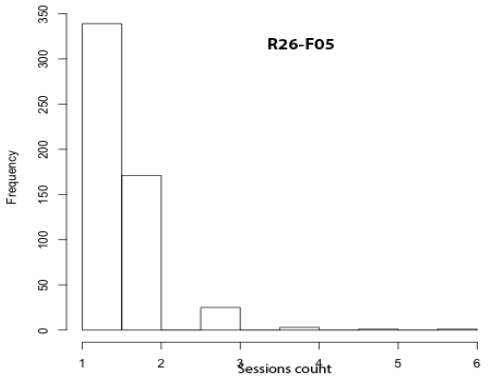
(c) Histogram of diagnoses count



(d) Histogram of medical procedures count



(e) Histogram of previous inpatient episode count



(f) Histogram of sessions count

Fig. 6.3 Histogram of the PMSI features for the diagnoses couple of Delirium F05-R26 Abnormalities of gait and mobility

6.2.3.2 Statistic measures of the features

For the second part of feature study, we illustrate the statistics of the interesting diagnoses couple DP-DS identified by the physician in the Section 6.2.1.2. The distributions of the

studied features follow normal and exponential distributions. Consequently, mean is a valid method to evaluate the average value.

For each interesting diagnoses DP-DS couple, Median and Mean measures are studied and presented in the Tables 6.12 6.13. The studied features are: "Age", "Medical procedures count", "Diagnoses count", "Length of stay", "Previous inpatient episodes count" and "Session count".

- **Min**: The minimum value of all the values.
- **Max**: The maximum value of all the values.
- **Mean**: The sum of the values divided by the number of values–often called the "average."
- **Median**: The value which divides the values into two equal halves, with half of the values being lower than the median and half higher than the median.

Table 6.12 Statistics on Age - Medical procedures Count - Diagnoses count

	Age				**Medical procedures Count**				**Diagnoses count**			
DP-DS	**Min**	**Max**	**Median**	**Mean**	**Min**	**Max**	**Median**	**Mean**	**Min**	**Max**	**Median**	**Mean**
R10-E66	14	92	67	62	1	182	5	12	3	68	11	15
K80-E66	19	87	62	61	1	73	5	9	3	92	7	11
I48-E66	30	96	69	71	2	126	9	16	4	69	13	16
I50-R26	72	102	90	88	1	119	13	20	5	91	20	24
J15-R26	21	103	88	85	2	297	9	20	6	55	14	16
J20-R26	3	102	84	80	2	115	9	14	4	73	13	16
R50-R26	5	95	81	77	1	148	7	13	5	50	15	18
J69-R26	28	95	83	80	2	148	9.5	16	4	55	13	15
F05-R26	68	99	88	87	2	47	7	9	6	54	16	18
S72-L89	69	96	88	86	3	79	7	13	6	88	13	20
J15-E44	53	97	86	84	2	365	11	28	8	92	15	19
J18-E44	35	98	83.5	81	1	204	7	13	4	42	14	15
F05-E44	65	99	86	85	3	47	6	8	5	43	14	15
R29-E44	74	99	86	86	1	26	6	8	6	42	15	15
R41-E44	53	99	87	86	1	35	6	8	7	36	13	15

For example, we notice that for the R10-E66 couple, the minimum age is 14 while the maximum is 92 the median is 67 and the mean age is 62 years old. The minimum medical procedure count is 1 the maximum is 182 the median is 5 and the mean is 12. The minimum diagnoses count is 3 the maximum is 68, the median is 11 and the mean is 15.

Table 6.13 Statistics on Length of stay - Previous inpatient stays - Sessions count

	Length of stay				Previous inpatient stays count				Sessions count			
DP-DS	Min	Max	Median	Mean	Min	Max	Median	Mean	Min	Max	Median	Mean
R10-E66	0	38	6	8	1	39	2	4	1	5	1	2
K80-E66	0	31	5	6	1	14	2	3	1	5	1	1
I48-E66	0	29	5	8	1	33	3	5	1	5	1	2
I50-R26	1	54	13	14	1	14	3	4	1	6	2	2
J15-R26	1	98	12	14	1	12	3	3	1	5	1	2
J20-R26	0	52	8	11	1	13	3	4	1	5	1	2
R50-R26	0	32	10	11	1	50	3	5	1	4	2	2
J69-R26	1	89	10	14	1	12	3	4	1	5	1	2
F05-R26	1	60	13	16	1	7	3	3	1	3	1	2
S72-L89	3	45	10	12	1	7	2	3	1	5	1	2
J15-E44	3	56	12	14	1	7	2	3	1	7	2	2
J18-E44	2	39	10	11	1	34	2	4	1	3	1	1
F05-E44	2	48	11	13	1	7	3	3	1	4	1	1
R29-E44	0	25	8	9	1	14	2	3	1	4	1	1
R41-E44	0	48	10	11	1	13	2	3	1	4	1	1

6.2.3.3 Conclusion

We retained the cut points of the three intervals applied on each diagnosis couple (DP-DS).

- "**Below**" [Min, Mean-sd].
- "**Mean**"]Mean-sd, Mean+sd[.
- "**Over**" [Mean+sd,Max].

The features, which have been discretized using the three intervals, are: medical procedure count, diagnoses count, age, length of stay and delay, whereas the features that are not normally distributed have been discretized using three equal interval binning, such as previous inpatient episodes count and sessions count. In total, we have used 183 features to build our ML model. A detailed description can be found in Table 6.14.

Table 6.14 The final retained PMSI features to prepare the database

	Feature Name	**Description**	**Valid values**
Personal	Gender	Patient's gender	F=Female, M=Male
	Age	Patient's age at admission	Below;Mean;Over
Inpatient episode	Length of stay	Time interval between admission date and discharge date	Below;Mean;Over
	Admission type	Patient's admission type	1= Emergency 2=Urgent 3=Elective 4=Newborn 5=Trauma 9=Information not available
	Provenance	The place where the patient is coming from	1=Acute care unit 2= Rehabilitation unit 3=Long-term care unit 4=Psychiatric unit 5=Passing through the institution's emergency facility 6=Hospitalized at home
	Disposition	Patient's discharge status	1=Discharge to home 2=Transferred to short-term facility 3=Transferred to skilled nursing facility 4=Transferred to intermediate care facility 5=Transferred to other healthcare facility 6=Transferred to home health care 7=Left AMA(Against Medical Advice) 20=Expired/Mortality
	Destination	The place where the patient is going after the discharge	1=Acute Care Unit 2=Rehabilitation unit 3=Long Term Care Unit 4=Psychiatric unit 6=home hospitalization 7=Medico-social housing structure
	Season	The season at the admission	Summer Winter Fall Spring
	Frequency	The count of the inpatient episodes of the patient during his life.	Below;Mean;Over
	Delay	Time interval between admission date and first medical procedure	Below;Mean;Over

Continue: Used features.

	Feature Name	Description	Valid values
Inpatient episode	Inpatient transfer count	The count of the transfers between medical units in the inpatient episode	Below; Mean; Over
	Medical procedures count	The count of the medical procedures during the inpatient episode	Below; Mean; Over
Clinical	Classified	A feature indicating whether the inpatient stay has a classified/important medical procedure or not.	0=No 1=Yes
	Emergency	A feature indicating whether the inpatient stay has an emergency case or not.	0=No 1=Yes
	Medical procedure groupings	19 features, each feature indicates whether the inpatient stay has a diagnosis within the corresponding medical procedure category.	0=No 1=Yes
	Urgent medical procedure grouping	5 features, each feature indicates whether the inpatient stay has a medical procedure within the corresponding urgent medical procedure category.	0=No 1=Yes
	Coarse level diagnoses granularity	19 features, each feature indicates whether the inpatient stay has a diagnosis within the corresponding diagnosis granularity.	0=No 1=Yes
	Fine level diagnoses granularity	126 features, each feature indicates whether the inpatient stay has a diagnosis within the corresponding diagnosis granularity.	0=No 1=Yes
Output	Label	A feature indicating whether the inpatient stay has the studied secondary diagnosis or not.	0=Negative 1=Positive

6.2.4 Imbalanced database

"A dataset is imbalanced if the classification categories are not approximately equally represented" (Chawla, 2005). In other words, a database is imbalanced if the studied subject does not have equal number of positive and negative examples. For example, if the studied subject is headache diagnoses, a dataset is imbalanced if it contains three times more cases of headache than cases without headache examples. Real life datasets are often imbalanced, this is particularly true in the medical databases. In our case, the Tables 6.2 6.4 6.3 6.5 6.6 6.7 6.8 6.9 show that the negative examples are nine times more frequent than the negative examples in most of studied diagnoses therefore our database is heavily imbalanced.

In the scientific literature different methods exist to solve the imbalanced dataset problem, most of these methods are discussed in the state of the art section.

In the dissertation we chose the baseline methods of undersampling and cost-sensitive, namely random undersampling and weighting methods, since they are effective and they do not cost much calculation power compared to oversampling methods. Moreover, oversampling methods tend to add more data that need to be processed. Furthermore, oversampling methods could add bias in the medical data and tend to perform worse than undersampling methods (Drummond and Holte, 2003). However, weighting methods could be considered equivalent to oversampling method if the minority class instances are given more weight than the majority class instances in certain algorithms, such as Decision Trees. For example, the minority class instances are considered twice more often if they have the double weight assigned to them in the building phase of a Decision Tree.

6.2.5 Conclusion

The main database preparation steps were presented as shown in the Figure 6.1. The first step was the relevant datasets selection according to the studied problem, where we used specialist help in order to select 80 relevant datasets to study. The second step was dataset transformation, which consisted in transforming the dataset from relational form (multi instances record) to flat form (single instance record) dataset. The second step also consisted in feature representation according to hierarchical levels. The third step was

feature preparations, it consisted in discretizing the continuous features according to three intervals calculated on the selected datasets. Finally, the last step consisted in balancing the dataset according to one of the balancing techniques (oversampling, undersampling, cost-sensitive).

6.3 Empirical Evaluation

An empirical evaluation of the PMSI database preparations approach is performed in this section. The evaluation is based on Machine Learning methods. In short, we use the datasets prepared according to our approach in order to build a prediction model that predicts secondary diagnoses. Finally, we evaluate the performance of the prediction model in order to determine the quality of the prepared database.

6.3.1 Objectives

The objective of the evaluations is on one hand to evaluate if the prepared datasets are of good quality, on the other hand, to compare different options in each preparation step.

Two main options in the preparation are compared:

- The impact of the diagnoses granularity representation level: which representation level produces a better performing prediction model (fine level when specific diagnoses groupings are considered, coarse level when general diagnoses groupings are considered).
- The impact of the dataset balancing techniques: which balancing technique produces a better prediction model.

6.3.2 Evaluation approach

The general evaluation method is presented in Algorithm 1 in order to build and to evaluate the prediction model. The first step (1->3) allows to choose the right configuration by fixing 3 variables:

- The weight of positive and negative examples (for instance, we decide to give the positive examples twice the weight of the negative examples in order to highlight their importance).
- The sampling option (Whether the dataset is going to be sampled or not).
- The granularity level of diagnosis (for instance, we represent the diagnoses based on the 19 features issued from general diagnoses chapters).

The second step (4) queries the most 10 frequent Primary Diagnoses DPs occurred with the studied secondary diagnosis. (for example, in case of "B96" bacterial agents infection as DS, the most frequent primary diagnoses found in the database are "**Acute tubulointerstitial nephritis**" with the code "**N10**", "**Malaise and fatigue**" with the code "**R53**", "**Fever**" with the code "**R50**", etc...)

Afterwards, (6) for each DP the corresponding dataset that contains the positive and negative examples are queried.

Then, all the pre-processings are performed(7->9), next (10) split the data into K training and testing sets and for each set (12) the training set is used to build a prediction model. We evaluate (14) the model using the testing set. Next (15), we average the evaluations produced by each fold. Finally (17), we average the evaluations of all the performances of the prediction model of the DPs.

In order to evaluate which preparation option is best, we compare the performances of different prediction models, each one being built using different choices according to following points:

- *Granularity level*: as the codification of the diagnoses belongs to a hierarchical classification, it is possible to use different levels of description: either coarse level with 19 features (which correspond to general chapters) or fine level of diagnoses with 126 features (more specific chapters).
- *Imbalanced dataset*: as the PMSI database contains by nature more negative examples than positive ones, we have made the hypothesis that a better performance prediction model can be built by balancing the number of positive and negative examples. To verify the hypothesis three sampling methods are considered.

Algorithm 1 The steps followed to build secondary diagnoses prediction model

```
Set(The weight of positive and negative examples)
Set (Sampling option)
Set (Granularity level of diagnoses)
Query the most 10 frequent DPs occurred with the DS
for each primary diagnosis DP do
    Query the dataset using the chosen granularity level
    Discretize the continuous features (age-length of stay - frequency - medical procedures count...)
    if sampling option is set then undersample the majority class
    Give the positive and negative classes their weights
    Split the data into k folds
    for Each fold do
        Build a prediction model with the training set
        Evaluate the model by measuring (Precision -Recall- F1) on the testing set
    end for
    Calculate the average evaluations of the folds
end for
Calculate the average of the evaluations of DPs
```

- The first method uses the original dataset without any sampling method.
- The second method gives the positive examples in the dataset double weight compared to the negative ones which is equivalent to oversampling the positive examples.
- The third method uses randomly undersampling technique with 1:1 ratio.

Several preliminary tests helped us to choose the weights and the ratio of random undersampling presented in this dissertation.

6.3.3 Implementation and results

Among the ML methods, we have chosen to use Decision Tree. The main reason behind this choice is the interpretability of the model. The extracted model can be easily verified by domain experts, such as physicians (Tuffery, 2007). In terms of performance, Decision Trees can produce better prediction models compared to Naive Bayes or Neural Networks using a similar structured data to predict some diagnoses as outlined in (Soni et al., 2011).

Moreover, Decision Trees are less sensitive to imbalanced datasets i.e. when the dataset contains unequally distributed classes (Cieslak and Chawla, 2008).

Decision Trees use an attribute selection rule at each node of the tree to split the data (split criterion), this rule is important to classify the records correctly. The main split criteria in the literature are Information Gain and Gini Index (Han et al., 2012). The difference in the performance between those two criteria is not huge. The best criterion is debatable and it depends on the used dataset (Raileanu and Stoffel, 2004). Since Gini Index tends to perform slightly faster than Information Gain (Raileanu and Stoffel, 2004), we retained Gini Index. For the Decision Tree, we have chosen the *Classification and Regression Tree* (CART) algorithm (Breiman et al., 1984) that uses Gini Index. The CART is a binary Decision Tree, which is built by recursively splitting each node into two child nodes, until there is no significant decrease in the Gini Index criterion. Overfitting problem occurs when the model is more accurate on the training set than on the testing data. Pruning can be used to avoid the overfitting problem (Han et al., 2012). The minimal cost-complexity pruning is implemented in the CART Decision Tree as described in (Breiman et al., 1984). Default parameters for pruning were used in our case because such overfitting problem could occur.

The performances are evaluated using 5-fold cross validation. In each fold, the dataset is divided into 80% training set and 20% testing set. The standard metrics are used to evaluate classification Precision, Recall and F1-measure.

The proposed evaluation approach is implemented using R-Studio [2] and Weka[3]. R-Studio is used to query the subsets from MySql[4] database where the PMSI is stored, then the preprocessing of the dataset is performed using R-Studio, next a dataset with the ARFF [5] format is produced, An ARFF (Attribute-Relation File Format) file is a text file that contains features description in addition to the dataset instances in a special format mostly used with weka. Finally, weka platform is used to build the CART Decision Tree, as shown in Figure 6.4. The approach is experimented according to three scenarios. In each scenario we represent features as described in the Section 6.2.2.2 which consists of coarse and fine level of diagnoses granularity. Moreover, we have changed the methods for sampling imbalanced data set. Hence, the three scenarios are described as following:

[2]https://www.rstudio.com/products/rstudio/
[3]http://www.cs.waikato.ac.nz/ml/weka/
[4]https://www.mysql.fr/
[5]https://weka.wikispaces.com/ARFF

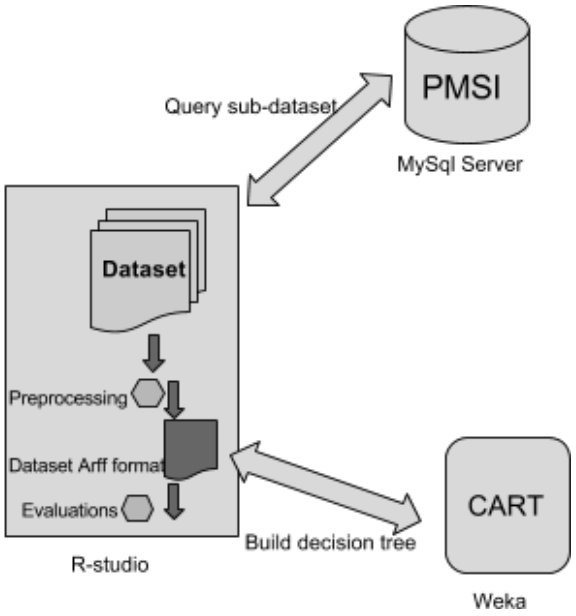


Fig. 6.4 Implementation of the database preparation algorithm

- **Scenario 1** corresponds to using the original dataset without any sampling.
- **Scenario 2** corresponds to cost-sensitive learning method for sampling imbalanced dataset.
- **Scenario 3** corresponds to random undersampling of negative examples to a ratio of (1:1).

Scenario 1

Figure 6.5 shows the results of different measures on the original dataset. First, we observe that even for fine and coarse granularity, using all the dataset is not an interesting strategy as recall and F1-measures results are very low. Except for B96 (bacterial agents) and J96 (Respiratory failure), our results approximate 2%. For B96 and J96, we observe that the results of fine granularity are more interesting than the results of coarse level granularity.

Scenario 2 In the lights of the results of the evaluations shown in Figure 6.6, we observe that the measurement varied between different diagnoses. On one hand, B96 scored the best for F1, precision and recall measurements, around 75%. On the other hand, other diagnoses scored lower percentages using the same measurements. As reported by (Stanfill et al., 2010), a same ML applied on different diagnoses, produces different results.

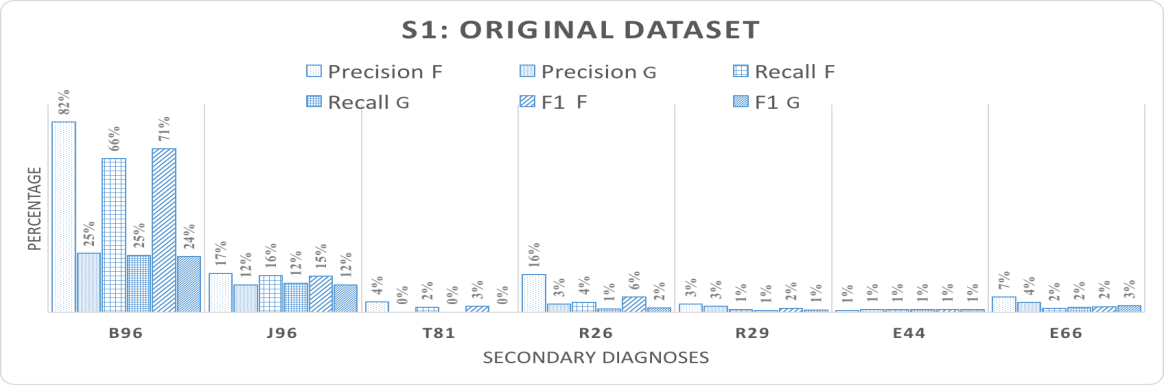


Fig. 6.5 The average measurements of the Decision Tree's performance in the scenario 1 - based on original dataset - using fine and coarse levels of granularity for all the studied diagnoses, F: Fine Level; C: Coarse Level

Our results confirm such a variation of measurements, and the complexity of the problem. Concerning the highlighted issues about the effect of the granularity level, we notice that using fine level granularity gives better measurements compared to using coarse level granularity. We observe that the differences between fine and coarse level of granularity range between 1% and 27% in the results Figure 6.6. In particular, for B96 we notice an important enhancement of results quality using the fine level granularity.

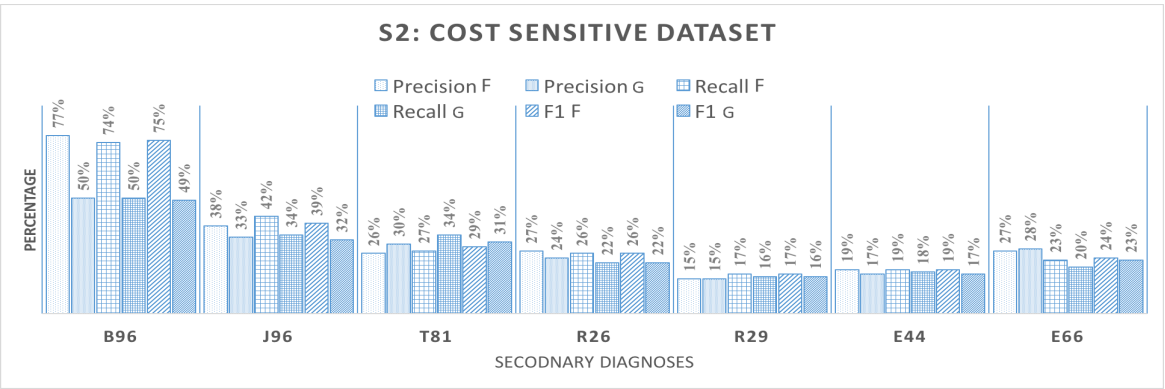


Fig. 6.6 The average measurements of the Decision Tree's performance in the scenario 2 - based on Cost-sensitive/Oversampling learning - using fine and coarse levels of granularity for all the studied diagnoses, F: Fine Level; C: Coarse Level

Scenario 3 Figure 6.7 shows the results of the third scenario. Clear improvement is observed in the quality of detection of all secondary diagnoses. Compared to the results presented in Figure 6.5 and Figure 6.6 in which the used sampling methods privileged B96 and J96 diagnoses, this evaluation substantiates that sampling negative examples according to 1:1 ratio is the best method to predict a right quality over all type of secondary

diagnoses. In fact, the results show that the values of the quality measures range between 55% and 84%, which are very trustworthy to satisfy our main objective. The difference of each sampling methods is observed clearly in Figures 6.8, 6.9 and 6.10, each figure shows the performance and the differences between the sampling methods using the metrics F1, Precision and recall in order.

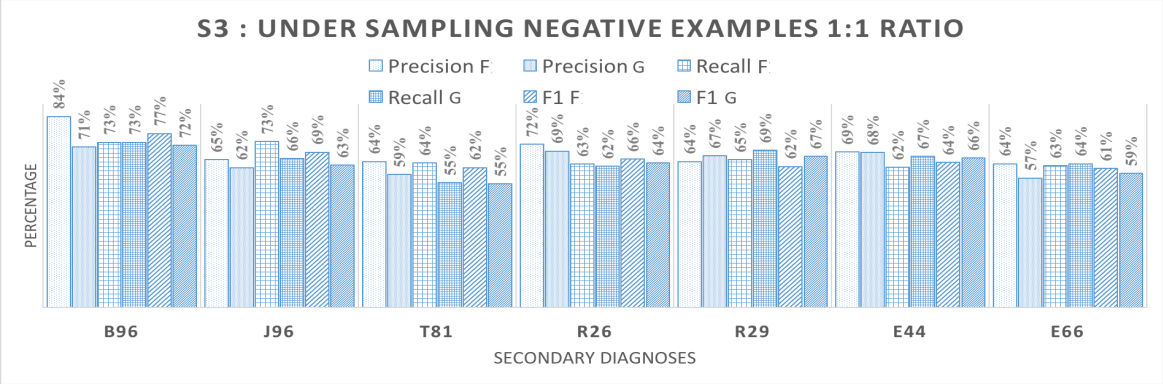


Fig. 6.7 The average measurements of the Decision Tree's performance in the scenario 3 - based on undersampled dataset - using fine and coarse levels of granularity for all the studied diagnoses, F: Fine Level; C: Coarse Level

To sum up the differences between the performed experimentations, we overlap the results of the three scenarios on the three metrics F1-measure, recall and precision respectively in Figures 6.8, 6.9 and 6.10. The most important remarks are:

- Fine level granularity features give better results than coarse level granularity features regardless the type of secondary diagnoses and the type of metric, this seems coherent with the fact that detailed level provide more information and give better prediction power.
- The method of sampling impacts the quality of results. We observe that the undersampling method improves the results significantly compared to the Cost-sensitive/ Oversampling and the original non sampled dataset regardless the type of secondary diagnoses and the type of metric. Intuitively, sampling methods are improving the quality because they make the number of positive examples more representative compared to negative examples.

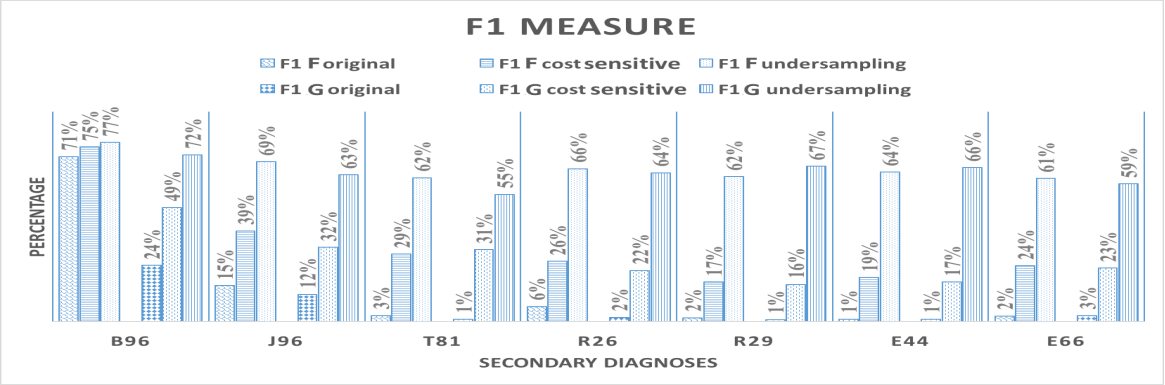


Fig. 6.8 The F1 measurements on the three sampling methods

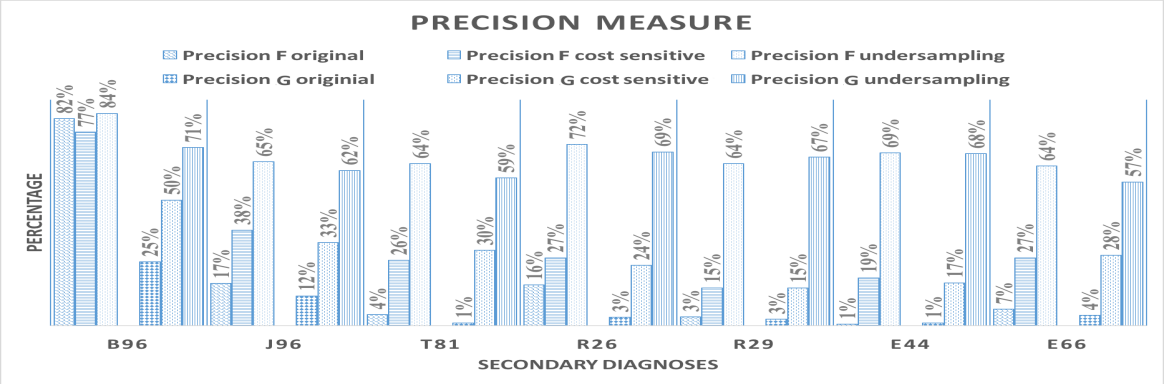


Fig. 6.9 The Precision measurements on the three sampling methods

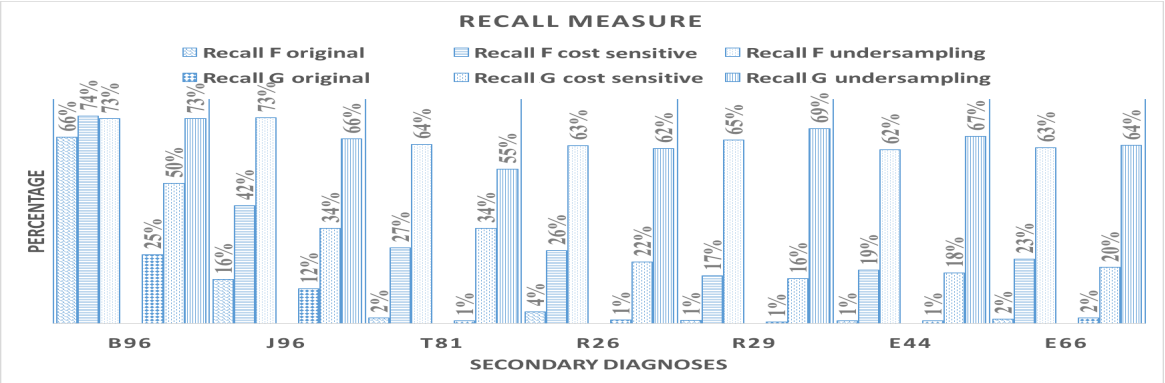


Fig. 6.10 The Recall measurements on the three sampling methods

6.3.4 Discussion

The hypothesis of filtering the datasets according to DP is motivated by observing that, in the hospital, encoding folders are distributed on the specialist coders according to

the main hospitalisation motivation of the inpatient episode, i.e. the DP of the inpatient episode. This hypothesis allowed to increase the quality of the dataset and target more case specific datasets. The results shown in the Section 6.3.3 indicate that the hypothesis is valid.

We proposed different preparation steps that have been evaluated by prediction models, such as Decision Trees. Concerning the best granularity representation level of diagnoses, as the codification of the diagnoses forms a hierarchical classification, it is possible to use different levels of description: either coarse level with 19 features (which correspond to general chapters) or fine level of diagnoses with 126 features (more specific chapters). The performances of two Decision Trees were compared, each tree was built using different level of diagnoses granularity. The results showed that by using the fine level of granularity we enhance on average 5% to 10% all the quality measures regardless of the predicted diagnosis code. The prediction power seems to be related to the preciseness of the medical information.

Some diagnoses had better performance Decision Tree compared to others, such as B96 "bacterial agents". B96 is the most frequent secondary diagnosis. The good prediction performance is explained either by the low imbalance ratio of B96 dataset, or by the medical specificity of bacterial agents. Since the selected datasets of B96 diagnosis contain only frequent and trivial combinations of DP-DS, therefore it is not difficult for a prediction model to discover the relation between DP and DS and predict the DS.

The customized discretization ranges for the continuous features adapted to each dataset provides better interpretability and improves the prediction quality. Moreover, a better understanding of predictive power of each feature could be established with the help of the medical staff in the hospital. The understanding of the features used in the predication models explain the behaviour and the performances of each model.

Concerning the imbalanced dataset, as the PMSI database contains by nature more negative examples than positive ones, the improvement of results in the third scenario when the balanced dataset is used confirms that balancing techniques are useful to produce better performance Decision Trees. In the second scenario Cost-sensitive/Oversampling learning is used by giving the positive examples in the dataset double weight compared to the negative ones, this technique produced 25% better performance model compared to the model based on original dataset. Finally, random undersampling technique were used to reduce the number of negative examples to be

equal with the positive ones, this technique generated the best performance model regardless to the predicted diagnosis which is a good step towards better database quality.

6.4 Conclusion

This chapter outlined the results of our approach to prepare the PMSI medical database to be used by Machine Learning methods. The proposed approach allowed us to determine each step, and to make efficient choices for implementing the preparation as mentioned in the figure 6.11

The strength of the proposed approach is to provide a generic structured dataset that can be populated with any database, while allowing personalized data preprocessing for each studied dataset.

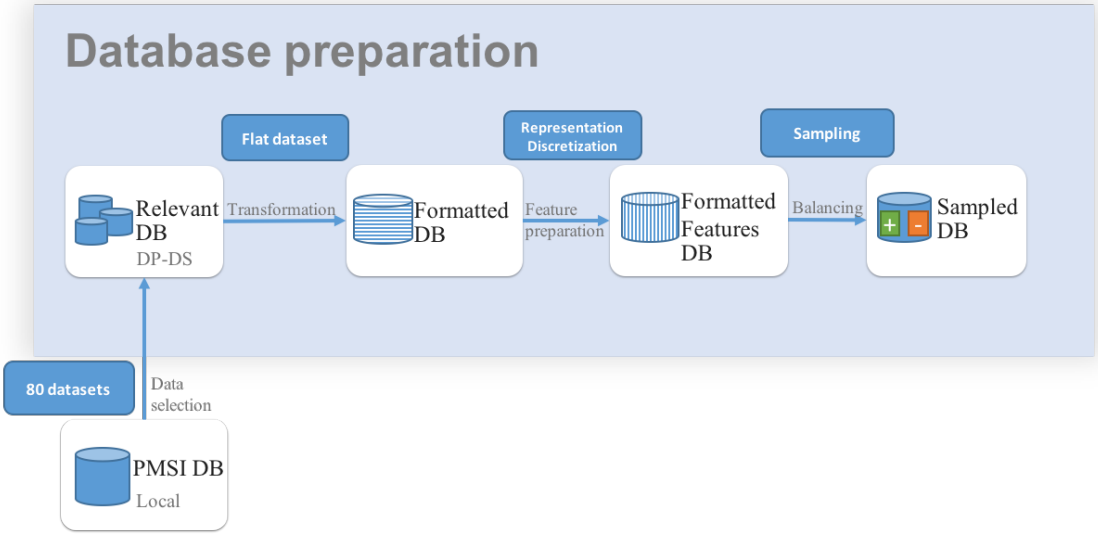


Fig. 6.11 The PMSI database preparation for Machine Learning analysis: implementation choices

The approach was evaluated by measuring the performance of CART prediction model, which was built using prepared PMSI datasets. The good performance of the CART showed the usefulness of the approach. Moreover, two options in the preparation steps were evaluated. The first option is the balancing method and the second option is diagnoses feature representation level. These options were mainly meant to address the imbalanced datasets and excessive number of features problems. In the first

scenario the original dataset was used without any sampling, in the second scenario Cost-sensitive/Oversampling learning was applied, and in the third scenario random Undersampling was applied. In each scenario we used two diagnoses representations: coarse and fine level of diagnoses granularity. The best performance model was achieved by using the third scenario i.e. random Undersampling and by using the fine level granularity of diagnoses representation. Therefore, we adopt these two options: random Undersampling and fine level granularity in the following chapters.

Chapter 7

Stable feature selection from medical databases

"Success is a science; if you have the conditions, you get the result."

-Oscar Wilde

Contents

7.1 Introduction

7.1.1 Objective

The main issue we address in this chapter is the **stability** and the **quality** of the features selected from Feature Selection (FS) methods.

- The features are **stable** when a FS method selects the same features from multiple datasets that represent the same subject. For example, if two different datasets exist for the same diagnosis, the features selected by FS method from both datasets should be similar in order be considered stable.
- The **quality** of features is the performance of the ML model using these features to predict the studied class.

Although the traditional sampling methods balance the datasets and enhance the performance of the prediction models, a new challenge needs to be addressed when sampled datasets are used with FS methods. The challenge is the difficulty to select stable features, because each time a database is sampled, some information gets lost in the case of undersampling or some information gets redundant in the case of oversampling. Therefore, different features are selected each time the dataset is sampled.

Some questions are raised in order to guide us to propose a new approach to select stable features.

1. How stable are the features when a FS method is applied on different sampled datasets?
2. How to select stable features out of imbalanced datasets?
3. Are the stable features good quality features?
4. How does the imbalance ratio of each dataset influence the quality of the stable features?
5. How to find out the values of the selected features from FS methods? knowing that traditional FS methods do not provide the values for the features.

Moreover, the new approach should not be affected from sampling methods and should provide not only stable but also good quality features.

Finally, we use Decision Trees (DT) to find out the values of the extracted features knowing that traditional FS methods do not provide them.

7.1.2 The used databases and preparation recall

Two scales of PMSI databases are used to implement the evaluations in this chapter: local and regional scale.

- The local scale of the PMSI database is extracted from the hospital of "Centre Hospitalier Intercommunal de Castres Mazamet" described in the Section 6.2.1.
- The regional scale of the PMSI database is extracted from all the regional hospital in Tarn. It contains around 1,200,000 inpatient episode records for the year of 2011.

We had limited access on the regional scale, since we were only allowed to work on it inside the hospital campus and only using hospital's computers. Therefore, we used the regional scale only to evaluate the prediction models extracted from local scale of PMSI.

All of the proposed methods in this chapter used database preparation is implemented the same way explained in the Chapter 6. The main steps of the preparation are:

1. Selecting relevant datasets to the studied problem.
2. Dataset transformation in order to transform the dataset an exploitable form by ML methods.
3. Feature preparation to put the feature into an exploitable form by ML methods.
4. Sampling the dataset to limit the imbalanced database problem.

We have selected 80 interesting DP-DS diagnoses couples. However, only the 14 original datasets original DP-DS couples are presented in most of the results in this chapter for the visibility reasons. The 14 original DP-DS couples are:

- Delirium *F05-E44* Malnutrition.
- Delirium *F05-R26* Abnormalities of gait and mobility.
- Atrial fibrillation and flutter *I48-E66* Obesity.
- Heart failure *I50-R26* Abnormalities of gait and mobility.
- Bacterial pneumonia *J15-E44* Malnutrition.
- Bacterial pneumonia *J15-R26* Abnormalities of gait and mobility.
- Pneumonia *J18-E44* Malnutrition.
- Acute bronchitis *J20-R26* Abnormalities of gait and mobility.
- Pneumonitis *J69-R26* Abnormalities of gait and mobility.
- Cholelithiasis *K80-E66* Obesity.
- Abdominal and pelvic pain *R10-E66* Obesity.
- Nervous and musculoskeletal system symptoms *R29-E44* Malnutrition.
- Cognitive function and awareness symptoms *R41-E44* Malnutrition.
- Fever *R50-R26* Abnormalities of gait and mobility.
- Fracture de femur*S72-L89* Pressure Ulcer.

7.2 Building a stable feature selection approach

The first section evaluates the features selected from sampled datasets by usual FS methods in order to compare them with our proposed approach. The second section studies the instability issue of the FS methods and proposes an approach to select stable features. The third section proposes an evaluation approach for the stable features. Finally, the fourth section studies the values of the stable features.

7.2.1 Evaluation of the features obtained by usual FS methods

The FS methods choose the most relevant features to the prediction class and ignore other features. The main advantage of FS methods is that they provide a better understanding of the underlying process that generates the data (Guyon and Elisseeff, 2003).

As we depend on FS methods, we want to evaluate first the features selected by usual FS methods. The followed approach in the scientific literature to evaluate features is to build a ML model using the selected features that predicts the studied class. The best features produce better models that predict more accurately the studied class (Chandrashekar and Sahin, 2014).

We propose an evaluation method presented in the Figure 7.1, where the first step is the **database preparation**. Afterwards, the dataset is split into training set and testing set. The training set is 80% of the database and the testing set is 20% of the dataset. The main reason for using this evaluation rather than k-fold method is to insure a fair comparison in the future evaluations by storing the testing set and use it all along the dissertation.

Feature selection step allow us to build some list of features. In the **Feature Selection** step, a training set is built using Random Undersampling method to balance the dataset. Afterwards, a FS method is used in order to select the relevant features to the prediction class.

Evaluation step allow us to evaluate the quality of the selected features. Subsequently, a prediction model is built using the selected features and the balanced dataset. A testing set is used to evaluate the prediction model using the Recall, the Precision, the F1 measure and AUC of ROC performance measurement metrics. These steps are shown in the Figure 7.1

Moreover, the evaluations allow us to compare the FS methods. By comparing these methods, we can choose the best ones and if needed to choose their best configurations.

For the FS we had to choose from three categories *Filter* methods, *Wrapper* methods and *Embedded* methods (Saeys et al., 2007). We intend to build a stable Feature Selection approach that works well with most of the ML algorithms. Therefore, the *Filter* methods are the best candidate, since they are independent from learning algorithms. Moreover, unlike *Wrapper* and *Embedded* methods, *Filter* methods are fast and they are scalable i.e. adapted to process large databases, such as PMSI medical databases. Furthermore, *Filter* methods are independent from the any classification method (Saeys et al., 2007).

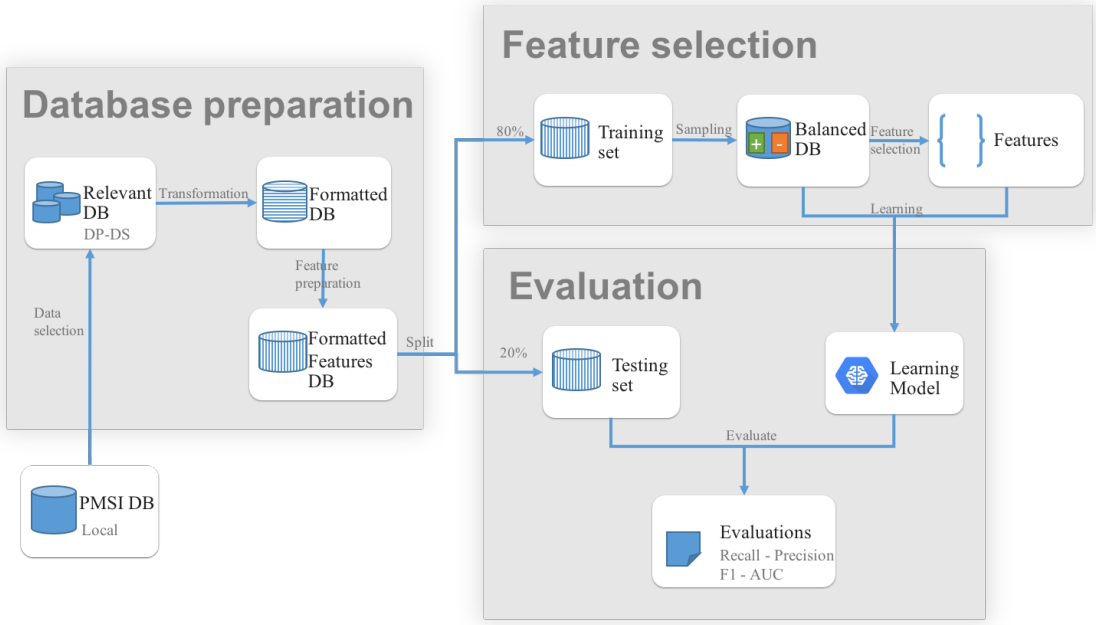


Fig. 7.1 Evaluation method of the features

Among *Filter* methods we used "*Gain Ratio (GainR)*" method used in the C4.5 Decision Tree building (Quinlan, 1993) and Correlation-based Feature Selection (CFS) proposed in (Hall, 1999) the former method does not consider the dependencies between the features called *Univariate* methods, whereas the later method considers these dependencies called *Multivariate* methods.

The **GainR** method is more advanced version of *Information Gain* method which reduces the bias toward selecting multivalued features. The attributes are ranked according to the GainR score. The features that score higher than a certain threshold are retained. Two thresholds are tested in the implementation, the values are chosen empirically in order to retain few and large number of features. The first value is 0.01 which retains 30 features on average, whereas the second value is 0.02 which retains 20 features on average.

The **CFS** method ranks feature subsets according to the degree of redundancy among the features. It searches subsets of features that are individually well correlated with the class but have low inter-correlation. The search algorithm used to search for the best subset is best first search algorithm (Pearl, 1984) starting from empty set and stopping the search for new features after 5 non improving iterations (the default parameters of the Weka platform).

The GainR and the CFS methods are representative of their categories, univariate - multivariate respectively and they can be generalised to any other method in the filter category.

Concerning the learning algorithm used in the "***Evaluation***" step of the procedure, the *CART DT* has been used for the main reason of interpretability of the model as well as the reasons mentioned in the database evaluation Section 6.3.3. The "*Naive Bayes*" is also used for the main reason of scalability and for the sake of comparison with the DT learning algorithms.

The results of the evaluation are shown in three Tables 7.1 7.2 7.3. The tables show the evaluation of the 14 original DP-DS couples retained by the physician. The first column represents the dataset identified by DP-DS CIM10 code, the second column is the features count when the FS method is applied on the dataset. The remaining columns are the performance of the two learning algorithms NB and CART DT in the means of F1, AUC of ROC, Precision and Recall performance measuring metrics.

The first Table 7.1 shows the results of the approach when it uses the Gain Ratio FS method with 0.01 value of the threshold. The second Table 7.2 shows the results of the approach when it uses Gain Ratio FS method with 0.02 value of the threshold. The third Table 7.3 shows the results when it uses CFS FS method with the best first search algorithm.

The results indicate that most of the prediction models are able to encode the studied secondary diagnoses with good accuracy except few diagnoses, such as F05_E44 i.e. "*Malnutrition*" **E44** in the inpatient episodes suffering from "*Delirium*" **F05** as primary diagnosis.

Moreover, in the inpatient episodes suffer from "*Nervous and musculoskeletal system symptoms*" **R29** as primary diagnosis, the DT prediction model encodes "*Malnutrition*" **E44** with high accuracy (i.e. F1=82% with Precision of 74% and Recall of 93%) using 24 features selected by GainR with threshold of 0.02. However, the accuracy drops when NB is used as a prediction model with (F1=73%) and drops even more when more features are retained using GainR with lower threshold (F1=66%). In the case of the CFS FS method is used, only 14 features are retained. The DT built over these features did not perform well F1=46% whereas the NB learner had acceptable performance F1=65%. In this particular case GainR FS with the threshold of 0.02 had the best performance.

Table 7.1 Prediction model performance when Gain Ratio is used with 0.01 threshold

DP-DS	Features count	CART				NB			
		F1	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**
F05_E44	15	50%	55%	55%	47%	51%	53%	55%	47%
F05_R26	38	50%	52%	54%	46%	57%	69%	73%	47%
I48_E66	23	53%	51%	52%	55%	52%	57%	61%	46%
I50_R26	43	64%	72%	69%	60%	76%	84%	85%	68%
J15_E44	59	64%	61%	59%	70%	62%	73%	75%	53%
J15_R26	46	63%	63%	63%	63%	**76%**	**86%**	**85%**	**68%**
J18_E44	34	65%	66%	62%	68%	66%	73%	73%	59%
J20_R26	41	56%	57%	57%	54%	68%	75%	74%	63%
J69_R26	31	46%	55%	55%	40%	59%	61%	63%	55%
K80_E66	26	67%	64%	60%	77%	50%	64%	65%	40%
R10_E66	33	**72%**	**69%**	**64%**	**81%**	67%	71%	70%	64%
R29_E44	33	66%	69%	65%	67%	65%	70%	68%	62%
R41_E44	38	58%	60%	58%	58%	71%	76%	76%	67%
S72_L89	56	72%	78%	73%	72%	55%	76%	78%	42%
	Average	60%	62%	60%	61%	62%	71%	72%	56%

Table 7.2 Prediction model performance when Gain Ratio is used with 0.02 threshold

DP-DS	Features count	CART				NB			
		F1	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**
F05_E44	13	55%	42%	46%	67%	55%	55%	54%	56%
F05_R26	19	64%	53%	59%	71%	48%	64%	71%	36%
I48_E66	13	41%	52%	55%	33%	61%	64%	67%	56%
I50_R26	19	56%	65%	75%	45%	**82%**	**90%**	**84%**	**80%**
J15_E44	23	57%	51%	57%	57%	77%	86%	83%	71%
J15_R26	22	69%	74%	75%	64%	81%	87%	85%	79%
J18_E44	14	55%	51%	53%	57%	57%	68%	57%	57%
J20_R26	24	56%	57%	60%	53%	71%	75%	79%	65%
J69_R26	14	78%	81%	70%	88%	47%	52%	44%	50%
K80_E66	19	69%	60%	56%	90%	59%	70%	67%	53%
R10_E66	25	68%	69%	68%	68%	61%	72%	65%	58%
R29_E44	24	**82%**	**83%**	**74%**	**93%**	73%	76%	67%	80%
R41_E44	22	69%	62%	58%	85%	67%	73%	64%	69%
S72_L89	48	74%	70%	63%	90%	53%	68%	80%	40%
	Average	64%	62%	62%	69%	64%	71%	69%	61%

On the contrary, some diagnoses had better prediction model performance using CFS FS method, such as "*Obesity*" **E66** in the inpatient episodes suffering from "*Atrial fibrillation and flutter*" **I48** as primary diagnosis using both NB and DT.

Table 7.3 Prediction model performance when CFS is used

DP-DS	Features count	CART				NB			
		F1	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**
F05_E44	31	29%	42%	32%	26%	69%	59%	56%	89%
F05_R26	29	58%	61%	70%	50%	33%	75%	75%	21%
I48_E66	17	**76**%	**64**%	**67**%	**89**%	71%	68%	85%	61%
I50_R26	14	70%	74%	77%	65%	68%	72%	63%	75%
J15_E44	12	63%	47%	56%	71%	50%	61%	60%	43%
J15_R26	11	69%	66%	75%	64%	69%	82%	67%	71%
J18_E44	12	67%	64%	69%	64%	48%	49%	47%	50%
J20_R26	27	73%	71%	75%	71%	54%	59%	50%	59%
J69_R26	19	67%	69%	71%	63%	57%	47%	46%	75%
K80_E66	12	65%	53%	52%	87%	70%	71%	54%	100%
R10_E66	17	67%	64%	65%	68%	78%	87%	82%	74%
R29_E44	14	46%	54%	55%	40%	65%	70%	63%	67%
R41_E44	22	77%	71%	77%	77%	67%	69%	64%	69%
S72_L89	15	67%	50%	50%	100%	**75**%	**84**%	**100**%	**60**%
	Average	64%	61%	64%	67%	62%	68%	65%	65%

Some general observations on the results are:

- On average, GainR method produces better features using higher threshold. Low threshold produces more features which does not yield necessarily to better performance learning algorithm. This indicates that the extra retained features in the case of the 0.01 threshold do not carry important features in order to encode the secondary diagnoses and they are considered as noisy features.
- The CFS method selects fewer features on average compared to the GainR method. However, the CFS method does not always produce better features on some diagnoses. Therefore, the number of features has big effect on the learner performance, it is very critical issue. Features should contain sufficient information to build effective prediction model without any noise to disturb the learning.

The variation of performances indicates that this is a complicated problem and there are a lot of factors that contribute to having good or bad features. Having robust sampling method is one of the factors. One of the objectives we are looking for is to create an approach that eliminates the noisy features and keeps only the few most decisive ones. Our main hypothesis is that sampling methods produce datasets that are not always representative to the learning problem and they contain sometimes noisy examples causing

irrelevant features to be retained. Consequently, we investigate in the next section the noise in the selected features by examining the features selected from different sampled databases.

For the FS methods we choose to use GainR with 0.02 threshold or CFS, since they both produce good quality learners.

7.2.2 An approach to select stable features

In this section, the effect of sampling the dataset on the selected features is discussed to answer the questions "How stable are the features when a FS method is applied on different sampled datasets?" and "How many datasets are required to select the stable features?".

We have two main objectives which are on the one hand to clarify if the selected features are different each time the dataset is sampled, on the other hand, to find out if significant number of features are common between the sampled datasets.

The main dataset sampling methods are oversampling and undersampling, where the composition of the dataset is modified either by generating more examples or by removing some examples in order to balance the number of positive and negative examples. The sampling methods affect the selection of the features, since the initial composition of the dataset is different from the sampled dataset. In this section, this effect is examined in order to help us build a new approach that selects stable feature no matter the sampling method used to balance the dataset.

In order to study the effect of sampling methods on the Feature Selection methods we built a method presented in the Figure 7.2 for each studied secondary diagnosis, as described above.

The first step is database preparation according to the Chapter 6. Then, the dataset is sampled multiple times in order to balance the positive and negative examples. Afterwards, the relevant features to the studied class are selected, using one of the FS methods. The relevant features are supposed to be different each time the dataset is sampled. Therefore, during the last step, we count the common features from two, three, four different sampled datasets respectively until the number of common features are

stable. The count of the features is plotted against the number of datasets used to select the common features.

By following this method, the number of datasets required to select a stable set of features is concluded. Moreover, the noisy features are excluded when the number of the intersected features stays the same after certain times of sampling repetitions.

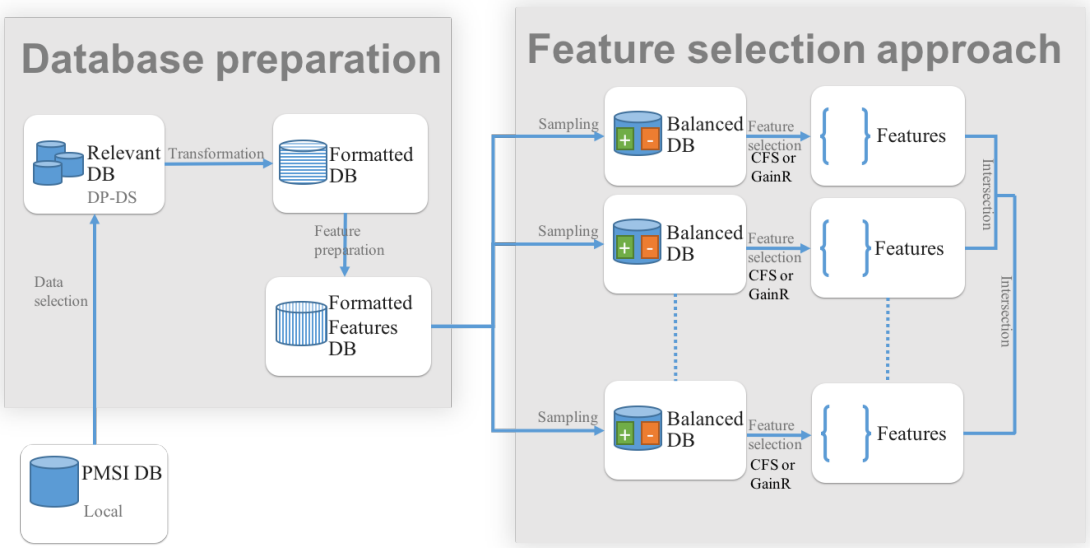
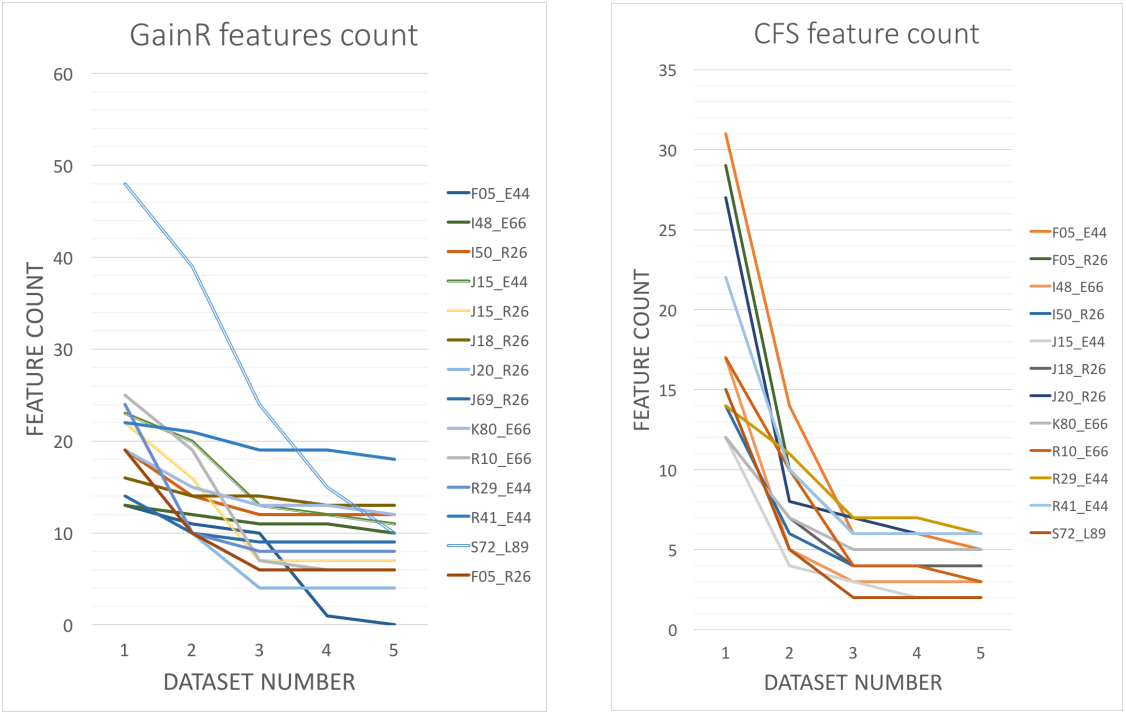


Fig. 7.2 Stable features selection approach

In order to visualise the common features, the count of the intersected features is plotted on one axis and the number of sampled dataset used to select the features on the other.

The results of the 14 original diagnoses are presented in Figure 7.3b in the case when the CFS method is used and in Figure 7.3a in the case when the GainR method is used.

The results indicate that the features of all the diagnoses start from large number then drop significantly when intersected with the features selected from another dataset, which indicates that the features are not stable and different set of features are retained each time the dataset is sampled. Moreover, the results show that the number of common features becomes stable when three datasets are used, which means that the common features of three independently sampled datasets contain stable features that appear in all the sampled datasets. However, in one case there are no common features which means



(a) In the Case of the Gain Ratio

(b) In the Case of the CFS

Fig. 7.3 The relation between the dataset count and stability of the features

that either the number of instances used in the FS method is not sufficient or the nature of the DS does not permit to encode it properly.

As a conclusion, we adopt the approach presented in the Figure 7.4 using three sampled dataset as an approach to select stable features.

We suppose that the mutual features selected from three sampled datasets contain the most relevant features to the prediction class, and exclude the features that are noisy and less relevant the prediction class. To support our claim, in the next section a classification model is built out of the stable set of features (i.e. the intersection of features from three different sampled dataset) in order to evaluate the quality of the stable features.

7.2.3 Evaluation of the the stable features

In this section, we evaluate the stable features selected using our approach. The evaluation method is presented in the Figure 7.5.

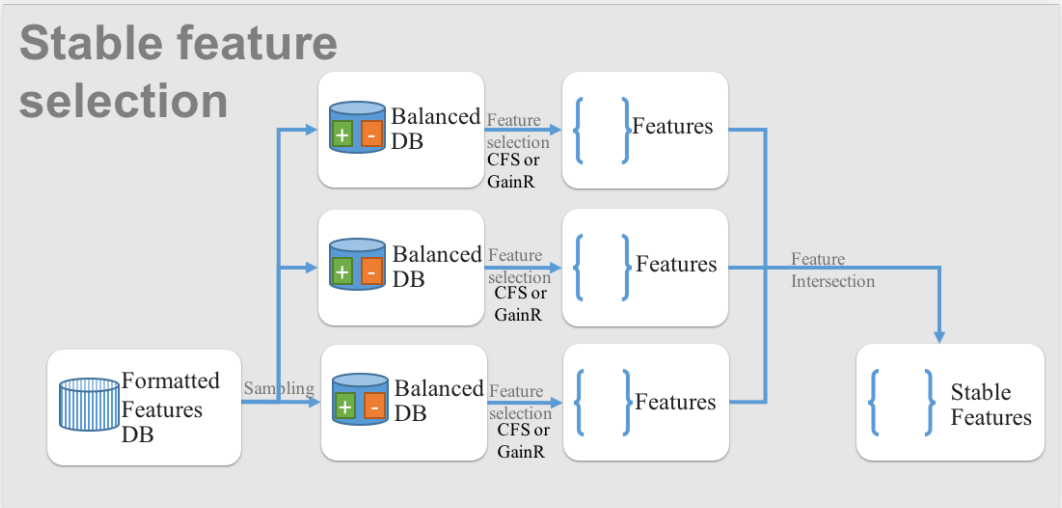


Fig. 7.4 Stable features selection approach

We answer the question, "**Are the stable features good quality features?**" by evaluating the stable features selected according to our approach.

The evaluation of the stable features is similar to the evaluation of usual features in the Section 7.2.1. We use the stable features to build a prediction model that encodes diagnoses to inpatient episodes, afterwards we measure the performance of the prediction model.

Two testing sets are used to evaluate the prediction model:

- The first testing set is extracted from local scale of PMSI database: we used the same testing dataset used in the evaluations of the usual features (presented in the Section 7.2.1) to insure fair comparison.
- The second testing set is extracted from regional scale of PMSI database which is important in order to verify the validity of the results obtained on the local scale of the PMSI database and to test the possibility to generalise the results on larger scales.

The evaluation approach mainly consists of:

- Database preparation: the PMSI database is transformed into a format that most ML methods deal with. Two scales of PMSI database are used:

  - Local scale: a part of it (80%) is used to select stable features and the other part is used to evaluate the extracted prediction model.
  - Regional scale: mainly used for evaluation purposes.

- Feature selection approach: stable features are selected from three sampled datasets.
- Evaluation: essentially, the stable features are used with a sampled dataset to build a prediction model. Then the model is evaluated using two testing sets: 20% of the local dataset and all the regional dataset. The metric used to measure the model performance are the Recall, the Precision, the F1 measure and AUC of ROC performance metrics.

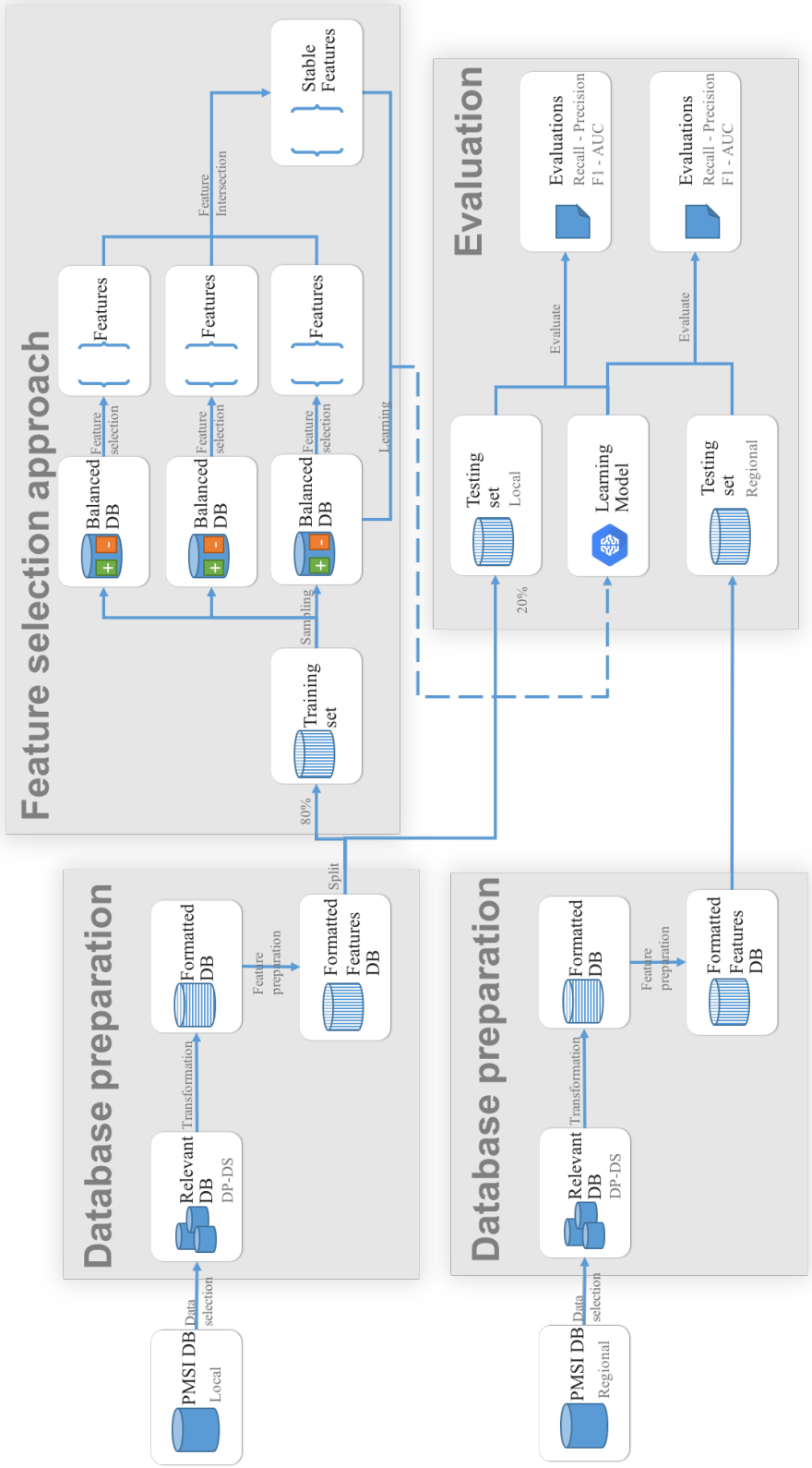


Fig. 7.5 Evaluation method of the stable features

7.2.3.1 Implementation and results

We proposed two possible use cases for our approach.

1. Help encoding secondary diagnoses at the beginning of the encoding session when only administrative and medical procedure information is available.
2. Help encoding secondary diagnoses at the end of the encoding session when all the information is available including recently encoded diagnoses.

Therefore, in order to evaluate the possible use cases, the overall approach is repeated two rounds:

1. All_Features: using all the features (Administrative - Medical procedures - Diagnoses).
2. No_Diag: using only (Administrative - Medical procedures) features.

Table 7.4 The tested use cases to evaluate stable features - Each situation is named as Test X

	PMSI Dataset scale	
	Local	**Regional**
All_Features	Test 1	Test 2
No_Diag	Test 3	Test 4

The evaluation results of the prediction models in the first round (Test 1 - Test 2) are presented in the Tables 7.5 7.6. The Table 7.5 presents the evaluations of the stable features selected by CFS FS method whereas the Table 7.6 presents the evaluations of the stable features selected by GainR FS method.

Only the 14 original datasets are presented. The first column is the DP-DS couple, the second column is the stable features which represents the output of our approach. The remaining columns are the evaluations of CART and Naive Bayes learners by the means of F1, AUC of ROC, Precision and Recall. A full reference of the used features are in the Appendix A).

Table 7.5 Evaluation of CFS stable features using NB and CART classifiers (Test 1 - Test 2)

		CART Decision Tree								**Naive Bayes**							
		Local PMSI				**Regional PMSI**				**Local PMSI**				**Regional PMSI**			
DP-DS	**CFS Features**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**
F05_E44	ModeSortie Chap01 Chap02 DIGEST04 DIGEST09 UROGEN07	64%	65%	65%	63%	48%	59%	62%	40%	69%	59%	56%	89%	61%	60%	58%	64%
F05_R26	Duree CARDIOV08 CARDIOV14 DIGEST11 PNEUMO09 TRAU_COTES	47%	45%	44%	50%	63%	57%	55%	72%	33%	75%	75%	21%	65%	70%	61%	70%
I48_E66	DIGEST15 GAUX04 UROGEN10	55%	64%	73%	44%	44%	62%	82%	30%	71%	68%	85%	61%	60%	68%	63%	58%
I50_R26	Chap01 GAUX05 NEURO05 ORLOS02	70%	73%	77%	65%	42%	61%	81%	28%	68%	72%	63%	75%	57%	67%	81%	44%
J15_E44	NbreDAS Chap01 Chap04	78%	71%	64%	100%	73%	65%	59%	96%	50%	61%	60%	43%	31%	61%	60%	21%
J15_R26	AgeAn NbreDAS Chap01 DIGEST03 GAUX05 NEURO04	**81%**	**72%**	**72%**	**93%**	54%	66%	65%	46%	69%	82%	67%	71%	58%	71%	73%	48%
J18_E44	NbreDAS Chap01	61%	50%	50%	79%	75%	68%	61%	96%	48%	49%	47%	50%	32%	50%	49%	24%
J20_R26	NbreDAS Chap01 INFECTIO03 UROGEN07	67%	69%	69%	65%	58%	61%	63%	53%	54%	59%	50%	59%	69%	75%	63%	76%
J69_R26	DERMATO10 DIGEST05 INTOX03 NEURO02 NEURO06 ORLOS08 PSY01	67%	66%	54%	88%	65%	57%	50%	93%	57%	47%	46%	75%	64%	58%	52%	83%
K80_E66	NbreDAS AUTRE02	71%	60%	56%	100%	**80%**	**74%**	**66%**	**100%**	70%	71%	54%	100%	80%	77%	66%	100%
R10_E66	ModeSortie NbreDAS AUTRE07 CARDIOV06 PNEUMO01	62%	62%	54%	74%	49%	62%	73%	37%	**78%**	**87%**	**82%**	**74%**	**74%**	**85%**	**79%**	**70%**
R29_E44	AgeAn Chap01 AUTRE02 NEURO04	69%	69%	65%	73%	52%	60%	54%	49%	65%	70%	63%	67%	36%	51%	43%	30%
R41_E44	AgeAn NbreDAS NbreActe Chap01 Chap05 AUTRE02 PLAIES_CE	73%	77%	89%	62%	65%	65%	65%	65%	67%	69%	64%	69%	58%	63%	61%	55%
S72_L89	Provenance NbreDAS	75%	80%	100%	60%	55%	63%	72%	44%	75%	84%	100%	60%	56%	74%	74%	45%

Table 7.6 Evaluation of GainR stable features using NB and CART classifiers (Test 1 - Test 2)

		CART Decision Tree								Naive Bayes							
		Local PMSI				Regional PMSI				Local PMSI				Regional PMSI			
DP-DS	**CFS Features**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**
F05_R26	Duree UROGEN07 NbreDAS CARDIOV13 GAUX05 CARDIOV09 PNEUMO09 TRAU_CRANE AgeAn Chap06	58%	66%	70%	50%	49%	56%	63%	40%	29%	47%	43%	21%	61%	69%	65%	57%
I48_E66	AgeAn DIGEST15 PSY01 GAUX04 Destination DERMATO06	59%	61%	63%	56%	60%	65%	62%	58%	77%	80%	71%	83%	60%	67%	62%	58%
I50_R26	Chap01 GAUX05 Duree PSY04 ModeSortie GAUX06 AgeAn DERMATO10 ORLOS02 UROGEN10 Destination	61%	59%	69%	55%	60%	59%	56%	64%	83%	91%	94%	75%	59%	79%	84%	45%
J15_E44	NbreDAS NEURO04 Chap01 Destination CARDIOV07 Chap04 AgeAn PSY04 GAUX01 DIGEST03 DERMATO02 UROGEN07	**83%**	**86%**	**100%**	**71%**	28%	52%	56%	19%	**83%**	**94%**	**100%**	**71%**	43%	65%	64%	32%
J15_R26	NEURO04 Chap01 Destination DERMATO10 ModeSortie NbreDAS AgeAn GAUX06 DIGEST03 AUTRE01 GAUX05 PNEUMO07 PNEUMO05	71%	75%	71%	71%	58%	65%	66%	52%	74%	87%	77%	71%	59%	80%	75%	49%
J18_E44	Chap01 NbreDAS Destination Duree GAUX01 AUTRE04 Provenance	56%	64%	64%	50%	40%	52%	47%	34%	67%	73%	80%	57%	48%	62%	57%	42%
J20_R26	NbreDAS NEURO04 NbreActe Chap01 ModeSortie GAUX03 Destination NEURO02 GAUX05 Chap07 PNEUMO07 UROGEN07 DERMATO10 Chap17	64%	74%	100%	47%	43%	55%	74%	30%	77%	85%	86%	71%	59%	72%	68%	52%
J69_R26	DERMATO10 NEURO02 Chap19 ORLOS08	58%	70%	88%	44%	42%	58%	65%	31%	70%	72%	62%	81%	44%	58%	59%	35%
K80_E66	NbreDAS NbreActe Chap17 DIGEST15 NEURO03 GAUX04 Chap06 Provenance UROGEN01	71%	60%	56%	100%	**80%**	**74%**	**66%**	**100%**	52%	70%	75%	40%	48%	77%	73%	36%
R10_E66	NbreDAS CARDIOV06 Destination AUTRE06 ModeSortie AgeAn GAUX04 AUTRE07 Duree Chap19 CARDIOV13 DERMATO02 CARDIOV07	67%	66%	65%	68%	44%	61%	78%	31%	56%	73%	59%	53%	35%	84%	90%	21%
R29_E44	AgeAn Chap01 NbreActe Destination NbreDAS AUTRE02 Chap06	79%	76%	72%	87%	14%	67%	67%	8%	63%	68%	59%	67%	52%	64%	58%	47%
R41_E44	AgeAn Duree NbreDAS Chap01 Destination NbreActe ModeSortie Chap19	54%	54%	54%	54%	67%	53%	52%	94%	69%	77%	69%	69%	**75%**	**76%**	**68%**	**84%**
S72_L89	NbreDAS AgeAn AUTRE07 Provenance AUTRE06 DIGEST03 Chap01 ModeEntree CARDIOV11 PSY03 NEURO02 CARDIOV06 CARDIOV13 Destination INFECTIO03 DERMATO05 DIGEST02 DIGEST07 Chap13 UROGEN09 UROGEN05 RHUMATO07 AUTRE05 RHUMATO04	40%	40%	40%	40%	36%	54%	59%	26%	44%	65%	50%	40%	45%	70%	79%	32%

The evaluation results of the prediction models in the second round (Test 3 - Test 4) had modest results compared to the first round when all the features are used. We used the average results in the discussion, therefore only the **average** results of the 14 original diagnoses are presented in the Table 7.7, the detailed results are presented in the Appendix A in the Tables A.2 A.3.

Table 7.7 Average performances of prediction models using stable features excluding diagnoses related features (Test 3 - Test 4)

			Local PMSI				**Regional PMSI**			
		Avg. feature count	**Avg. F1**	**Avg. AUC**	**Avg. Prec**	**Avg. Rec**	**Avg. F1**	**Avg. AUC**	**Avg. Prec**	**Avg. Rec**
CFS	**NB**	3	55%	58%	60%	54%	46%	55%	51%	46%
	CART	3	55%	60%	62%	54%	42%	55%	56%	39%
GainR	**NB**	6	58%	59%	57%	60%	45%	56%	54%	43%
	CART	6	54%	57%	58%	52%	47%	52%	56%	48%

7.2.3.2 Results discussion

The discussion and the observations of the results are organised from the following point of views.

First, we compare the results of the two FS methods (CFS - GainR). Second, we compare the evaluations using the two classification models (CART - NB). Then, we discuss the scalability of results using regional and local scales of PMSI. We compare the results with the basic approach presented in the Section 7.2.1. Finally, we discuss the use case of the application.

With regards to the FS method point of view, two different areas are observed (feature quantity and quality). In general, FS methods reduce the features based on feature relevance and redundancy with respect to prediction class. In the implementation CFS and GainR methods are used. The CFS is considered *Filter Multivariate* category which evaluates a set of features and eliminates redundancy. The GainR method is considered *Filter Univariate* category which evaluates the relevancy of each feature independently. Concerning the quantity of features retained in each method, the CFS FS methods selects fewer features, most of them are included in the features selected by GainR method. The

average number of features by CFS is 5 compared to 13 by GainR. These features are stable across the FS method. Concerning the quality of the features, both methods (CFS-GainR) produce prediction models with similar averages of performance metrics, the CFS method slightly exceeds except when NB is used with GainR method. Moreover, the performance of the prediction model produced from both methods drops a bit when tested on regional scale. Consequently, both FS methods have accepted rates of performance even thought the number of features are different which indicates that the approach has eliminated the noisy features nonetheless some redundant features are retained by GainR without affecting the performance. Both FS methods produce good prediction model but with different feature count. The FS choice is a matter of preference in the use case.

In regards to the prediction model, both models CART and NB have similar performances on average. CART method surpasses NB in terms of interpretability whereas NB surpasses CART in terms of scalability. Other differences exist on the diagnoses level in terms of Precision and Recall metrics. According to the final objectives of the application a suitable prediction model should be adapted. For example, our use case has more interpretability requirements therefore CART is better option as long as the size of the examples do not surpass the machine's capabilities. The approach is not designed to be used only with certain algorithms, other learning algorithms could perform better using the same features, before adopting a final algorithm a thorough research should be done.

With regards to the scale of the PMSI database used in the evaluation, on average, the performance on local scale of the PMSI is higher than on the regional scale. The reason is that on the regional scale the population of the patients is changed significantly. The patients have different origins and different orientations. Therefore, the generalisation of the prediction model is not appropriate in this case.

With regards to improvements brought to the usual FS methods (presented in the Section 7.2.1), the results generally indicate that the stable features selected using three sampled datasets are better compared to the features selected only from one sampled dataset. Moreover, the number of stable features is significantly lower than the number of features selected from one dataset. Therefore, having few good features indicates the elimination of most of the irrelevant features.

With regards to the use case, the objective in the dissertation is to help the coders encode all the secondary diagnoses by completing the list of potential diagnoses. The intervention to complete the potential diagnoses is possible on two levels:

1. At the beginning of the encoding, when there are not any diagnoses available only; administrative and medical procedures are available.

2. At the end of the encoding, when all the information is available including the diagnoses recently encoded.

The results presented in the Table 7.7 indicate that encoding DS using only administrative information has bad performance and thus it can not be used in the final application. The second intervention is more likely to be used in the final application, at the end of an encoding session, all the information is available and the prediction model extracted from our approach can be used to encode secondary diagnoses. (Lecornu et al., 2009) share the same observation remark, the predication performance decreases significantly when only Administrative and Medical procedures are used.

Furthermore, with regards to the use case, it is worth to mention not all the diagnoses have the same effect on the value of the inpatient episodes, therefore some diagnoses are more important than others and they are classified according to severity level index. The physician in charge of the DIM department in the hospital, he explained to us the precision of the encoded diagnoses is more important in the case of low severity i.e. we do not want to overwhelm the coders with low precision predictions and the final gain is not significant to the hospital. However, the Recall is more important when encoding high severity diagnoses i.e. the coders do not mind verifying multiple times the high severity diagnoses even if the success rate is low. The current approach provides only the important features without potential values due to lack of the FS approaches to provide them. Therefore, an important enhancement to the approach is to provide the values of the important features. This enhancement is discussed in the Section 7.2.4.

7.2.3.3 The influence of the imbalance ratio on the features quality

All the studied diagnoses datasets are imbalanced with different imbalance ratios. In this section, we are interested in studying the effect of the imbalance ratio on the quality of the stable features. Therefore, we answer the question "How does the imbalance ratios of each dataset influence the quality of the stable features?".

In order to answer the question, the imbalance ratios of all the datasets are plotted on one axe and the performances of the prediction models built using the stable features

are plotted on the other axe. The quality of the stable features is measured by the F1 performance metric of ML model to encode the secondary diagnoses. Finally, the output figures are observed for any interesting patterns.

For this experiment, we plotted the results by using "*Tableau*[1]" a professional visualization tool, in order to study the relation between the imbalance ratios of the medical datasets and the quality of the stable set of features (i.e. selected from three datasets) measured by the performance of the classification model built using the stable features.

For this experiment, all the 80 studied datasets are plotted, the 14 original diagnoses are distinguished by orange colour. The datasets are identified by the diagnoses couple (DP-DS). Two classification models are tested (CART Decision Tree, Naive Bayes), Two FS method are tested (GainR, CFS) to select stable set of features.

The results are presented in four Figures. The Figure 7.6 shows the performance of the CART DT when stable features are selected by the CFS FS method. The Figure 7.7 shows the performance of the NB classifier when stable features are selected by the CFS FS method. The Figure 7.8 shows the performance of the CART DT when stable features are selected by the GainR FS method. The Figure 7.9 shows the performance of the NB classifier when stable features are selected by the GainR FS method.

The results presented in the Figures 7.6 7.7 7.8 7.9 show that the four combination of FS methods with the learning algorithms have similar patterns. Most of the diagnoses are located in the top left corner in the figures which indicate on one hand that most of the diagnoses are heavily imbalanced with ratios lower than 20%. On the other hand, the performances of the prediction models are good, higher than F1=60% regardless to the low imbalance ratio. This is true no matter to the FS method used and no matter to the prediction model used. Moreover, the results show that some diagnoses scored even higher than F1=75%. Therefore, the imbalance ratio of the dataset does not affect the performance of the prediction model, consequently the stable features are good features regardless to the imbalance ratio.

The results also indicate that some datasets are not imbalanced. For example, three diagnoses located at the right side of the figures (N41-B96, J44-J96, J96-J96) are considered balanced datasets with imbalance ratios higher than 90%.

[1]https://tableau.com

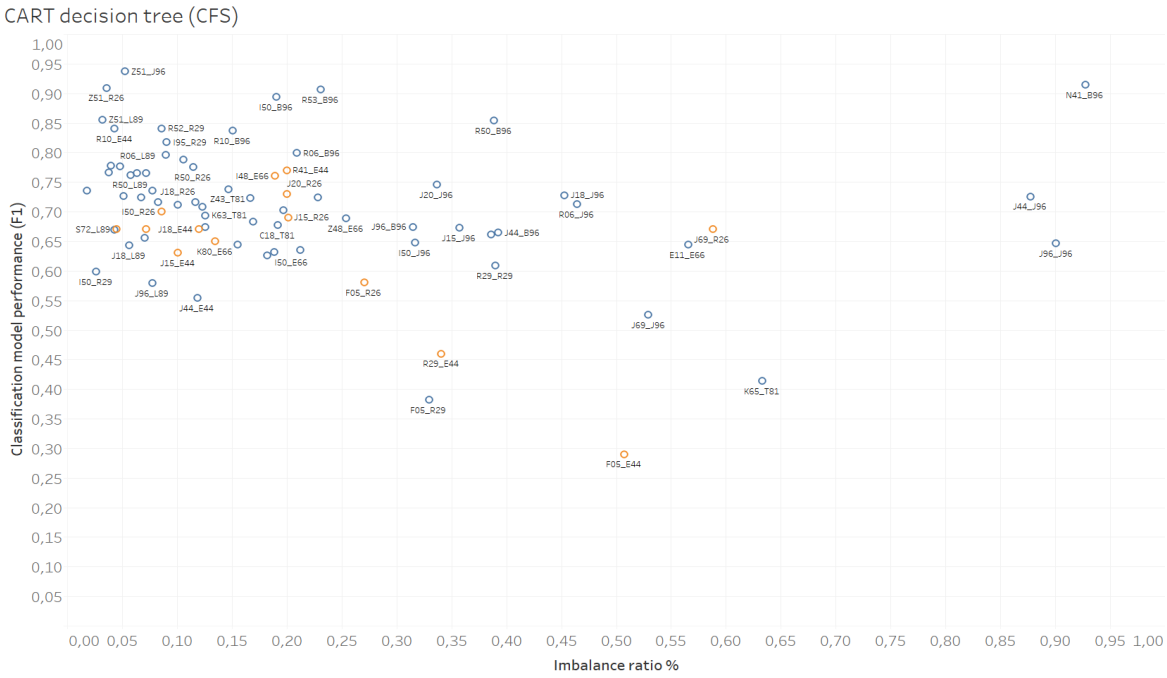


Fig. 7.6 The effect of the imbalance ratio on the performance of the "CART Decision Tree" built using stable set of features selected from "CFS" Feature Selection method

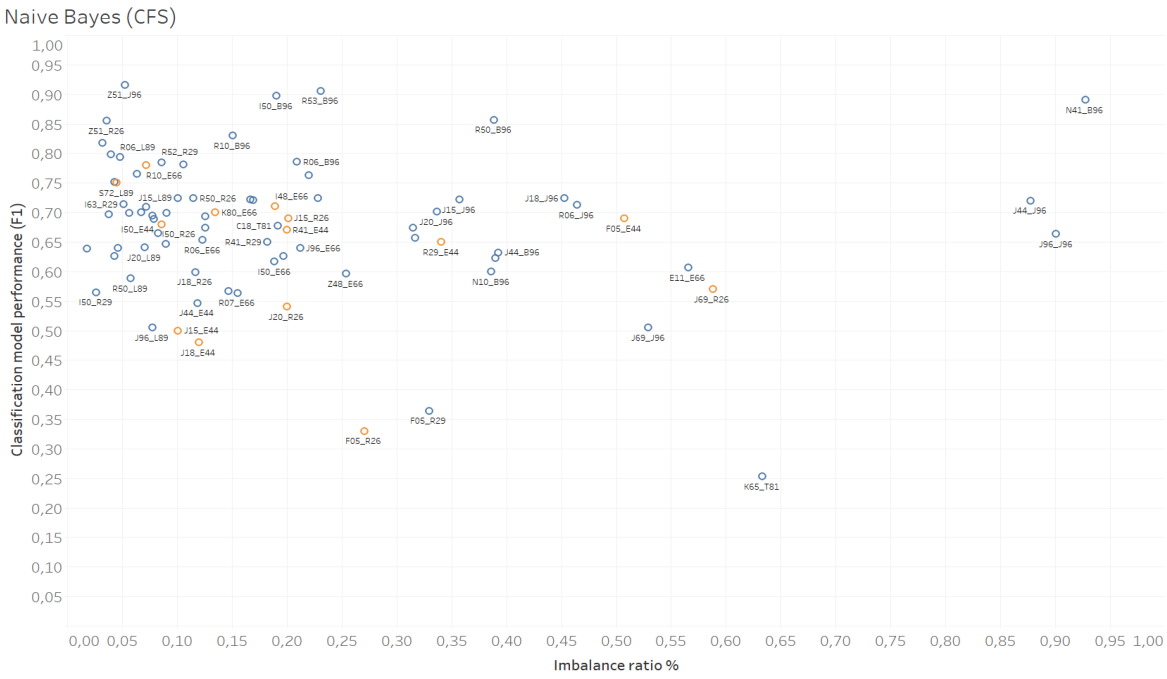


Fig. 7.7 The effect of the imbalance ratio on the performance of the "Naive Bayes" built using stable set of features selected from "CFS" Feature Selection method

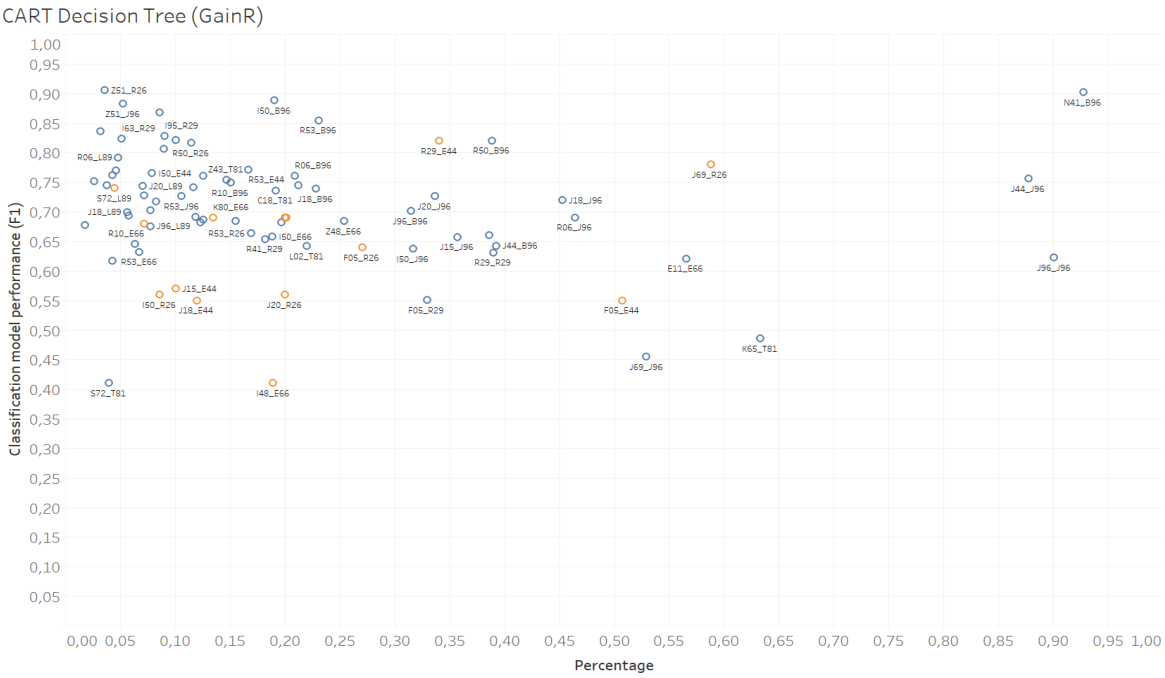


Fig. 7.8 The effect of the imbalance ratio on the performance of the "CART Decision Tree" built using stable set of features selected from "GainR" Feature Selection method

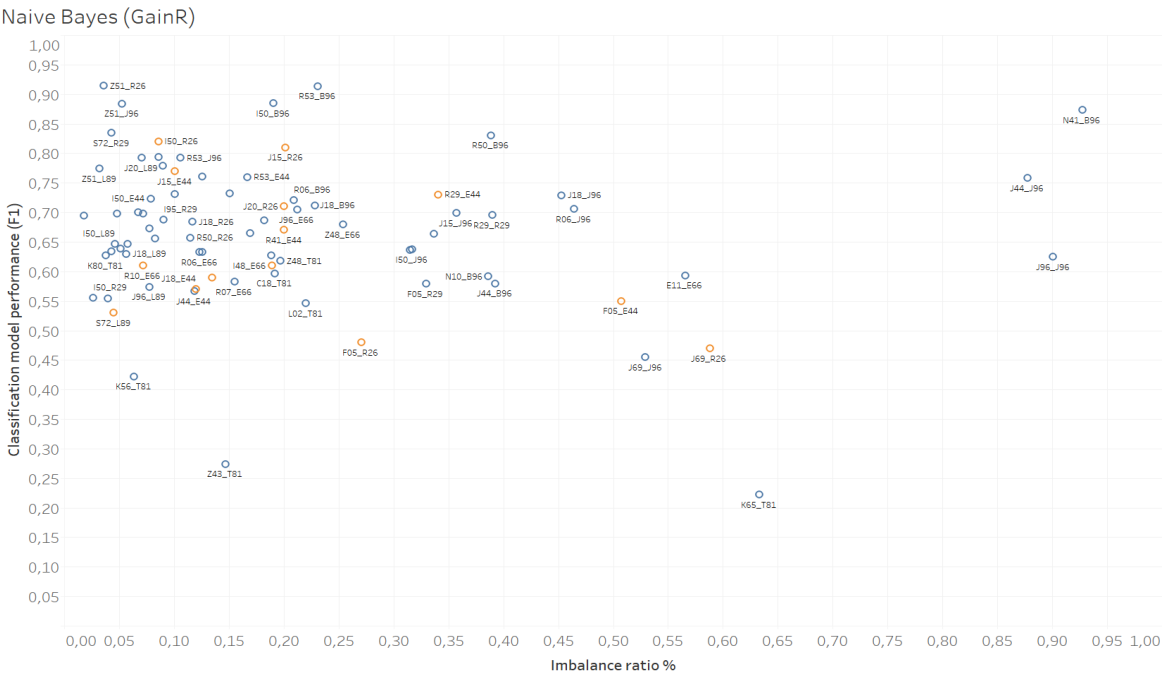


Fig. 7.9 The effect of the imbalance ratio on the performance of the "Updatable Naive Bayes" built using stable set of features selected from "GainR" Feature Selection method

Concerning the original diagnoses distinguished with orange color in the Figures 7.6 7.7 7.8 7.9, some of them are heavily imbalanced, others are mildly imbalanced. The performances of the learners are various. Therefore, the original diagnoses are representative sample of the larger diagnoses population, which in turn can validate the generalization of results of our approach to the entire diagnoses in the PMSI database.

7.2.4 Resolving feature value

The proposed approach that selects stable features does not provide values for them although it is an important information in the medical field and it supports taking more efficient decision when provided in the right moment.

One possible solution is to look into the composition of the ML algorithm used to evaluate the features. This solution is possible only if the produced ML model is interpretable and only if the model allow us to understand how the features are connected.

Since we used Decision Trees (DT) as evaluation method in our proposed approach, it is possible to look into the extracted model. The DTs are interpretable, they allow us to check out the values of the features that contributed to make a prediction. However, DTs can become large and difficult to interpret. Therefore, in order to simplify the interpretation of the features from DT, we propose to decompose the tree into IF-THEN rules that reveals the features values and their interrelations according to the prediction class. Moreover, the IF-THEN rules may be easier for humans to understand compared to DT, especially when the DT is very large (Han et al., 2012).

The steps to decompose the decision tree are straightforward, each leaf representing a rule. Since we are interested in the positive examples, we create one rule for each leaf that has positive class prediction. Each rule consists of the nodes from the root to the leaf separated with the logical AND=$\bigwedge$. Each node represents a condition. The rules implied disjunction between them using logical OR=$\bigvee$.

For example, the Decision Tree of the J20-R26 is:

NbreDAS=(low)|(mean)
| *Chap01=(0)*
| | *INFECTIO03!=(0): yes*

| *Chap01!=(0): yes*
NbreDAS!=(low)|(mean): yes

It is transformed into the following rules:
(NbreDAS=low|mean $\bigwedge$ *Chap01=0* $\bigwedge$ *INFECTIO03=1)*
$\bigvee$ *(NbreDAS=low|mean* $\bigwedge$ *Chap01=1)*
$\bigvee$ *(NbreDAS=over).*

We applied this transformation on all the DTs built using our approach presented in the Tables 7.5 7.6. The extracted rules are presented in the Tables 7.8 7.9. The first column presents the feature couple DP-DS, the second column presents the rules extracted from the Decision Tree predicting the secondary diagnoses.

Medical domain is a complicated domain, particularly if it is related to medical diagnoses, sometimes it is not possible to select any decisive features, such as the diagnoses presented in *italic* in the Tables 7.8 7.9 The diagnoses that had bad CFS features are (F05-E44; F05-R26; I48-E66; K80-E66). The diagnoses that had bad GainR features are (F05-R26; F05-R26; K80-E66, S72-L89) knowing that F05-R26 does not have any GainR features therefore there are not any rules in this case.

One of the good Decision Trees is the DT that encodes "*Abnormalities of gait and mobility*" **R26** in the presence of "Bacterial pneumonia" **J15**. The DT has F1=81% performance when the stable CFS features are used. The extracted rules from DT are:
(Chap01=0 $\bigwedge$ *NEURO04=0* $\bigwedge$ *AgeAn=Low* $\bigwedge$ *NbreDAS=over)*
$\bigvee$ *(Chap01=0* $\bigwedge$ *NEURO04=1)*
$\bigvee$ *(Chap01=1)*

The DT has F1=71% performance metric when the stable GainR features are used. The extracted rules from DT are:
(NEURO04=0 $\bigwedge$ *Chap01=0* $\bigwedge$ *PNEUMO07=1* $\bigwedge$ *AgeAn!=mean)*
$\bigvee$ *(NEURO04=0* $\bigwedge$ *Chap01=0* $\bigwedge$ *Destination!=2)*
$\bigvee$ *(NEURO04=1* $\bigwedge$ *AUTRE01=0)*

We remark that NEURO04 (Disorientation and cognitive impairment), AgeAn (Age) and Chap01 (Central, peripheral and autonomous nervous system) are common features in both (CFS and GainR) FS methods.

With regards to the use case, during coding the secondary diagnoses of the inpatient episodes who suffer from "*Bacterial pneumonia*" **J15** as primary diagnosis. When-

Table 7.8 CFS Features relation

DP_DS	**Features**
F05_E44	(ModeSortie!=9|8|9 ⋀ UROGEN07=0 ⋀ Chap0=1) ⋁ (ModeSortie=6|7 ⋀ UROGEN07=1 ⋀ Chap01=1) ⋁ (ModeSortie!=9|6|7 ⋀ UROGEN07=1)
F05_R26	*(CARDIOV14=0 ⋀ Duree=mean|over)*
I48_E66	*(GAUX04=1)*
I50_R26	(Chap01=1) ⋁ (Chap01=0 ⋀ GAUX05=1)
J15_E44	(NbreDAS=mean|over)
J15_R26	(Chap01=0 ⋀ NEURO04=0 ⋀ AgeAn=Low ⋀ NbreDAS=over) ⋁ (Chap01=0 ⋀ NEURO04=1) ⋁ (Chap01=1)
J18_E44	*(NbreDAS=mean|over)*
J20_R26	(NbreDAS=low|mean ⋀ Chap01=0 ⋀ INFECTIO03=1) ⋁ (NbreDAS=low|mean ⋀ Chap01=1) ⋁ (NbreDAS=over)
J69_R26	(PSY01=0 ⋀ DERMATO10=0 ⋀ DIGEST05=0 ⋀ ORLOS08=0) ⋁ (PSY01=0 ⋀ DERMATO10=1)
K80_E66	*(NbreDAS=mean|over)*
R10_E66	(CARDIOV06=0 ⋀ AUTRE07=1) ⋁ (CARDIOV06=1)
R29_E44	(AgeAn=mean|over ⋀ NEURO04=0 ⋀ Chap01=1 ⋀ AgeAn=low|mean)
R41_E44	(AgeAn=over)
R50_R26	(NbreDAS=mean|over ⋀ GAUX03=0 ⋀ AgeAn=mean|low ⋀ NEURO04=1) ⋁ (NbreDAS=mean|over ⋀ GAUX03=1)
S72_L89	(NbreDAS=low|mean ⋀ Provenance!=5|2) ⋁ (NbreDAS=over)

ever one of the rules is applicable on the inpatient episode, such as (NEURO04=1 ⋀ AUTRE01=0) the prediction model encodes the secondary diagnosis "*gait and mobility*" **R26**. Therefore, R26 is added to the list of potential diagnoses. Moreover, the rules extracted from the Decision Trees are presented to the coders in order to help the coders understand the prediction basis.

Table 7.9 CFS Features relation

DP_DS	Features
F05_R26	*(AgeAn=low ⋀ NbreDAS=over ⋀ Duree=over) ⋁ (AgeAn=over ⋀ NbreDAS!=low ⋀ PNEUMO09=0 ⋀ TRAU_CRANE=0 ⋀ Duree=low|mean ⋀ CARDIOV09=0 ⋀ Chap06=0 ⋀ CARDIOV13=0) ⋁ (AgeAn!=low ⋀ NbreDAS!=low ⋀ PNEUMO09=0 ⋀ TRAU_CRANE=0 ⋀ Duree=low|mean ⋀ CARDIOV09=0 ⋀ Chap06=0 ⋀ CARDIOV13=1) ⋁ (AgeAn!=low ⋀ NbreDAS!=low ⋀ PNEUMO09=0 ⋀ TRAU_CRANE=0 ⋀ Duree=mean ⋀ CARDIOV09=0 ⋀ Chap06=0 ⋀ CARDIOV13=1 ⋀ UROGEN07=1) ⋁ (AgeAn!=low ⋀ NbreDAS!=low ⋀ PNEUMO09=0 ⋀ TRAU_CRANE=0 ⋀ Duree=mean ⋀ CARDIOV09=0 ⋀ Chap06=0 ⋀ CARDIOV13=0) ⋁ (AgeAn!=low ⋀ NbreDAS!=low ⋀ PNEUMO09=0 ⋀ TRAU_CRANE=0 ⋀ Duree=low|mean ⋀ CARDIOV09=1) ⋁ (AgeAn!=low ⋀ NbreDAS!=low ⋀ PNEUMO09=0 ⋀ TRAU_CRANE=0 ⋀ Duree=over) ⋁ (AgeAn!=low ⋀ NbreDAS!=low ⋀ PNEUMO09=1)*
I48_E66	(GAUX04=0 ⋀ AgeAn=low) ⋁ (GAUX04=0)
I50_R26	(Chap01=0 ⋀ Duree=mean ⋀ GAUX05=0 ⋀ Destination!=2|7|0|D|4 ⋀ GAUX06=1) ⋁ (Chap01=0 ⋀ Destination!=0|4 ⋀ Duree=mean ⋀ GAUX05=1) ⋁ (Chap01=0 ⋀ Destination!=0|4 ⋀ Duree!=mean) ⋁ (Chap01=1 ⋀ ModeSortie=7|8 ⋀ Destination=2|0 ⋀ AgeAn=over) ⋁ (Chap01=1 ⋀ ModeSortie=7|8 ⋀ Destination!=2|0) ⋁ (Chap01=1 ⋀ ModeSortie!=7|8)
J15_E44	(Chap01=0 ⋀ DERMATO02=1) ⋁ (Chap01=1)
J15_R26	(NEURO04=0 ⋀ Chap01=0 ⋀ PNEUMO07=1 ⋀ AgeAn!=mean) ⋁ (NEURO04=0 ⋀ Chap01=0 ⋀ Destination!=2) ⋁ (NEURO04=1 ⋀ AUTRE01=0)
J18_E44	(Duree=low ⋀ Destination!=0|3|4) ⋁ (Duree!=low ⋀ Destination=0|2|3|4 ⋀ AUTRE04=0 ⋀ Chap01=1) ⋁ (Duree!=low ⋀ Destination=0|2|3|4 ⋀ AUTRE04=1) ⋁ (Duree!=low ⋀ Destination!=0|2|3|4)
J20_R26	(GAUX03=0 ⋀ UROGEN07=0 ⋀ DERMATO10=0 ⋀ NEUMO07=1 ⋀ ModeSortie=8|7|6 ⋀ NbreActe!=over) ⋁ (GAUX03=0 ⋀ UROGEN07=0 ⋀ DERMATO10=0 ⋀ NEUMO07=1 ⋀ ModeSortie!=8|7|6) ⋁ (GAUX03=0 ⋀ UROGEN07=0 ⋀ DERMATO10=1 ⋀ Destination=2|7|1|3|4 ⋀ NbreActe=over) ⋁ (GAUX03=0 ⋀ UROGEN07=1) ⋁ (GAUX03=1)
J69_R26	(DERMATO10=0 ⋀ NEURO02!=0) ⋁ (DERMATO10=1)
K80_E66	*(NbreDAS!=low)*
R10_E66	(CARDIOV06=0 ⋀ Destination!=1|0|4|3|6|D) ⋁ (CARDIOV06=1)
R29_E44	(AgeAn=over ⋀ Chap01=1 ⋀ Chap06=0 ⋀ NbreActe=mean ⋀ Destination=2|4) ⋁ (Chap01=1 ⋀ Chap06=0 ⋀ NbreActe=mean ⋀ Destination=7|2|4|0|3 ⋀ AgeAn=over ⋀ NbreDAS=mean) ⋁ (Chap01=1 ⋀ Chap06=0 ⋀ NbreActe=mean ⋀ Destination=0|3 ⋀ AgeAn=over) ⋁ (Chap01=1 ⋀ Chap06=0 ⋀ NbreActe=over ⋀ Destination=0|7|3|4 ⋀ AgeAn=over)
R41_E44	*(Destination!=1|0)*
S72_L89	*(AUTRE07=1)*

7.3 Discussion

The use case of the result is to complete the list of potential diagnoses in order to help the coders spot all the Secondary Diagnoses (DS) knowing the examined Primary Diagnosis (DP).

The number of features is an important factor to consider in the final application therefore the CFS method is a better choice, since it retains few and good quality features.

Each different hospital has different coding strategy which is not necessarily similar to the other hospitals. Moreover, the data in each hospital is not perfect, and it can contain some errors. Therefore, the features selected from the one hospital database were not generalised on the regional scale of PMSI. We propose in the future perspectives, to apply our approach on the regional scale or even on the national scale in order to select general features that can be applicable on more than one hospital. This would reduce the error effect of each hospital and select general features by taking into account the vast majority of the cases that represents most of the hospitals.

Although there are a lot of studies targeting the classification algorithms applied on imbalanced datasets, very few targeted Feature Selection method (Maldonado et al., 2014; Martín-Félez and Mollineda, 2010; Yin et al., 2013) described in the state of the art.

The main drawback of the proposed approach by (Maldonado et al., 2014) compared to ours is the dependency to the classification algorithm (SVM algorithm). While we implemented CART and NB classification methods in our approach, the choices are not limited to these two algorithms. Any other classification algorithm that satisfies the interpretability and scalability criteria can also be used in the approach, such as ID3, C4.5 and SVM. Moreover, although we mainly study our approach using two filter based FS methods (CFS-GainR), our approach can also be applied using other FS methods, such as entropy-based method and classifier-dependent methods.

Furthermore, the approaches that use clustering methods to balance the dataset (Martín-Félez and Mollineda, 2010; Yin et al., 2013) are dependent on the user input. Our proposed approach is user independent, since the used sampling methods do not require any domain knowledge in order to be applied, whereas clustering methods require user input in order to work properly, and moreover do not eliminate necessarily the heavily imbalanced problem. All the prior works are exploratory studies, and the problem of developing a general approach for Feature Selection method on imbalanced data remains open. This dissertation represents a step forwards in this direction.

The strength of our approach is providing a generic approach independent from sampling methods, FS methods or learning algorithms. Several sampling methods could be used to balance the dataset, any FS method could be used to select features, any

classification method could be used to build a predicting model. The weakness of the approach is that it does not consider the semantic level of features which improves the understanding of the features and the relationship between each other. Moreover, the diversity of the medical information that could exist between hospitals is not considered in the evaluation, i.e. what is considered true in one hospital could be false in another hospital. Technically, the proposed approach requires long preparation of the dataset in order to transfer it into a suitable format. Another weakness of the approach that it does not compete with methods that use non-structured data (usually text) as input, because such methods have higher prediction rates. However, our approach uses alternative structured sources with acceptable accuracy to support encoding diagnoses.

7.4 Conclusion

This chapter outlined our approach to select stable features and their values from imbalanced medical databases.

First, we tested FS approaches by applying FS methods directly on sampled dataset, the results indicated that a lot of selected features are noisy and are not necessarily relevant to the studied problem. The principal cause of the noisy features was the sampling methods used to balance the database.

Second, a new approach was proposed to balance the dataset and to evaluate the approach using prediction models. The approach proved that it eliminates the noisy features without losing the prediction power of the remaining features, which indicated they are the relevant features to the studied case.

The results proved that the stable features of the approach produced good prediction model regardless to the imbalance ratio. Finally, we proposed an extension to the approach based on the Decision Trees to provide the value of the selected features.

The approach was applied on the PMSI medical database. Two scales of PMSI databases were evaluated, local scale with 90,000 instances and regional scale with 1,200,000 instances. The approach was evaluated by measuring the quality of a classification model based on "CART Decision Tree" and "Naive Bayes" algorithms in terms of F1-measure, Recall, Precision and AUC of ROC.

The results indicated good stability of the features which led to good predicting model of secondary diagnoses on the local scale of the hospital and acceptable encoding performance model on the regional scale.

Chapter 8

Conclusion

"*Have no fear of perfection - you'll never reach it.*"

-Salvador Dali

Contents

In this chapter, we firstly provide general remarks on the thesis, then we summary the main contributions and results accomplished during this PhD thesis. Afterwards, we present a set of enhancements and give an insight into new research directions that could enrich this work.

8.1 General remarks

The main sources of encoding diagnoses are clinical reports, physician's interpretations, discharge letters and other medical documents that are usually written in free text. The approaches that use these sources have high prediction rates of diagnoses encoding. However, these sources are complicated to manipulate and requires a lot of preparation in order to be ready to exploitable by Machine Learning methods.

This thesis was an attempt to investigate and to find an alternative source of information that is easy to manipulate in order to produce a model with acceptable prediction rates to encode diagnoses. In this thesis we investigated the PMSI database which has well structured information and has information that is relatively easy to manipulate. The PMSI database contains previously encoded data of inpatient episodes. Coders can not encode diagnoses unless they have access to the main encoding sources.

Therefore, we experimented the possibility of a machine to encode diagnoses using only the available information in PMSI database, which is exhaustive, reliable and rich of standard well encoded inpatient episodes. We believe that information, such as diagnoses (different from the one being encoded) and medical procedures occurred during the inpatient information could contain valuable information sufficient to encode a diagnoses with acceptable accuracy rates. Therefore, we built a Machine Learning method that uses this information stored in PMSI to encode diagnoses and addressed important scientific challenges related to the exploration of the PMSI database. However, the unsteady results obtained by our approach prove that PMSI information alone is not completely sufficient for a machine to encode diagnoses with high accuracy. The research area to encode diagnoses remains open, specially the approaches that use only the PMSI information efficiently. Nonetheless, integrating multiple sources, such as "Electronic Medical Records" or any contextual information beside the PMSI can enhance the encoding accuracy and can open new technical challenges.

With the obtained results in the thesis, we proposed a new process to encode diagnoses adapted to the real followed process in the hospitals. In each step of the process we addressed specific challenges, such as data selection, database format, features construction, sampling and stable feature selection.

However, we did not reinjected the proposed process in the hospital due to time constrains and difficulty to make the proposition to interact with intern softwares of encoding diagnoses.

An example of the proposed process would be: during an encoding session of a 70 years old female patient, entered by the urgency service of the hospital for abnormal coughing suspecting *Inhalational pneumonia* problem, the coders follow these steps:

First, the coders make their first search in the medical sources for diagnoses. They encode all the obvious diagnoses and they look into the signs for potential encodings. The uncertain diagnoses are saved in the potential list. At this point, we intervene to suggest diagnoses according to the encoded information in the inpatient episode. For instance, the tool detects the rule **IF** (*Chap01*=0 $\bigwedge$ *NEURO04*=1) is satisfied **THEN** DS=R26. In simpler words the feature *Chap01*=0 means there is no "*Central, peripheral and autonomous nervous system*" medical procedure in the inpatient episode and the feature *NEURO04*=1 means there is a diagnoses related to "*Disorientation and cognitive impairment*". Therefore, the tool adds the **R26** diagnosis in the potential list of diagnoses. The tool adds also the rule led to this encoding i.e. the rule (*Chap01*=0 $\bigwedge$ *NEURO04*=1). Third, the coders need to confirm or to exclude the diagnosis in the potential list. Therefore, the coders go further and search for evidences and criteria in other sources, such as medical records, lab tests and radio images.

8.2 Conclusion

In this thesis we proposed a generic approach to prepare medical databases for Machine Learning methods. We also proposed an approach to select stable features in addition to their values effectively from databases. Our approach was particularly effective in areas where the databases are imbalanced, such as PMSI database.

The application domain of our approach is to support encoding secondary diagnoses by completing the potential diagnoses list. The background knowledge to understand the application domain is presented in the second, fourth and fifth chapters. In the second chapter, we presented the medical databases in general and the PMSI database in particular. We introduced the encoding of medical information in the PMSI in general and encoding diagnoses in particular as well as the challenges related to it. In the fourth chapter, we introduced the main studies done in the scientific literature to encode diagnoses in the hospital. We provided details on different approaches followed to encode diagnoses in the context of PMSI. In the fifth chapter, we presented our experience in the real observation sessions on the encoding diagnoses and proposed a use case to our contribution in the hospital.

Finally, we proposed our approach to prepare and select stable features in the sixth and seventh chapters. In the sixth chapter, we addressed the medical database preparation, in particular the challenges related to the data selection, data transformation, feature preprocessing and the imbalanced datasets. We also proposed our evaluation approach of the preparation phase. Finally, we implemented and evaluated our approach on the PMSI database issued from a local hospital. Chapter seven complemented chapter six by studying the influence of imbalanced dataset on the stability of the selected features. Moreover, we presented our approach to select stable feature from imbalanced datasets. Furthermore, we evaluated our approach on local and regional scales of PMSI database.

The evaluations showed that the selected stable features are effective to encode certain diagnoses, and less effective to encode others. Hence, our method can be applied to increase the integrity of the encoded diagnoses only on certain diagnoses. However, the feasibility of the proposed approach in the real encoding environment is not mature yet, more tests are required in order to integrate and to detect all the diversity of the medical information stored in different hospitals.

8.3 Perspectives

Research conducted during this PhD thesis helped addressing challenges which hinder the development of Machine Learning and Feature Selection techniques. The outcomes

of our research work open important and interesting research perspectives. Some of them are listed below.

- Our approach used a local scale of PMSI database to extract stable features and build predictive model. Moreover, the evaluations have been carried out on the local and regional scale of PMSI. However, our approach is scalable. Therefore, bigger databases could be used to build the model if the right learning methods are used. Therefore, a good generalisation to the results will be using a regional or a national scale of PMSI database to build the model. Consequently, more instances will be taken into account and more generalised model will be generated.
- Due to difficulty and time constrains, we did not evaluate the workflow in the hospital. A future perspective will be evaluating the proposed workflow in real diagnosis encoding environment.
- We used few methods to implement and validate our proposed approach. A future perspective will be testing and evaluating more implementation choices in order to choose the best method that gives the best evaluation results. For example, we try our approach using more sampling and choose the best one, likewise we choose the best Feature Selection method and the best the Learning method. Moreover, we choose the best combination of these methods that work the best in our approach.
- Integrate encoding sources other than PMSI database in the approach, such as "Electronic Medical Records" and contextual information.
- The general design of the proposed approach in the dissertation is prone to be applied in other application domains where the databases are heavily imbalanced, such as fraud detection and security attacks.

Appendix A

The used PMSI features

Table A.1 The PMSI database features

Category	Features	Label	Values
Pathology code	AUTRE01	Iatrogeny and post surgical complication SAI	1=Yes; 0=No
Pathology code	AUTRE02	Application for certificates, testing, counseling	1=Yes; 0=No
Pathology code	AUTRE03	Psychosocial, socio-economic difficulties	1=Yes; 0=No
Pathology code	AUTRE04	Remedies related to the organization of continuity of care	1=Yes; 0=No
Pathology code	AUTRE05	Reorientation, fugues, refusal of care	1=Yes; 0=No
Pathology code	AUTRE06	Control, monitoring and maintenance care	1=Yes; 0=No
Pathology code	AUTRE07	Other recourse	1=Yes; 0=No
Pathology code	CARDIOV01	Angina and other ischemic heart disease	1=Yes; 0=No
Pathology code	CARDIOV02	Cardiac arrest	1=Yes; 0=No
Pathology code	CARDIOV03	Cardio-circulatory shock	1=Yes; 0=No
Pathology code	CARDIOV04	Aortic dissection	1=Yes; 0=No
Pathology code	CARDIOV05	Unsolved precordial or thoracic pain	1=Yes; 0=No
Pathology code	CARDIOV06	High blood pressure and blood pressure	1=Yes; 0=No
Pathology code	CARDIOV07	Hypotension without mention of shock	1=Yes; 0=No
Pathology code	CARDIOV08	Myocardial infarction	1=Yes; 0=No
Pathology code	CARDIOV09	Heart failure	1=Yes; 0=No
Pathology code	CARDIOV10	pericarditis	1=Yes; 0=No
Pathology code	CARDIOV11	Peripheral Phlebitis	1=Yes; 0=No
Pathology code	CARDIOV12	Peripheral arterial thrombosis	1=Yes; 0=No
Pathology code	CARDIOV13	Rhythm and Conduction Disorder	1=Yes; 0=No
Pathology code	CARDIOV14	Other cardio-vascular diseases	1=Yes; 0=No
Pathology code	CONT_LS	Contusions and superficial mucosal lesions (excluding wounds and CE)	1=Yes; 0=No
Pathology code	DERMATO01	Abscess, phlegmons, boils, ...	1=Yes; 0=No
Pathology code	DERMATO02	Atopic dermatitis, contact, pruritus	1=Yes; 0=No
Pathology code	DERMATO03	erysipelas	1=Yes; 0=No
Pathology code	DERMATO04	Erythema and other eruptions	1=Yes; 0=No
Pathology code	DERMATO05	Mycoses, parasitoses and other skin infections	1=Yes; 0=No
Pathology code	DERMATO06	Localized edema and swelling	1=Yes; 0=No
Pathology code	DERMATO07	Bites of arthropod, insects, ...	1=Yes; 0=No
Pathology code	DERMATO08	Urticaria	1=Yes; 0=No
Pathology code	DERMATO09	Cutaneo-mucosal viruses	1=Yes; 0=No
Pathology code	DERMATO10	Other dermatological disorder	1=Yes; 0=No
Pathology code	DIGEST01	Appendicitis and other appendicular pathology	1=Yes; 0=No
Pathology code	DIGEST02	Ascites, jaundice and hepatopathy	1=Yes; 0=No
Pathology code	DIGEST03	Constipation and Other Intestinal Functional Disorder	1=Yes; 0=No
Pathology code	DIGEST04	Diarrhea and gastroenteritis	1=Yes; 0=No
Pathology code	DIGEST05	Unspecified abdominal pain	1=Yes; 0=No

Continued on next page

Table A.1 – continued from previous page

Category	Features	Label	Values
Pathology code	DIGEST06	Gastritis, Gastroduodenal ulcer not hemorrhagic	1=Yes; 0=No
Pathology code	DIGEST07	Gastrointestinal haemorrhage without mention of peritonitis	1=Yes; 0=No
Pathology code	DIGEST08	Lithiasis, infection and other bile duct injury	1=Yes; 0=No
Pathology code	DIGEST09	Nausea, vomiting	1=Yes; 0=No
Pathology code	DIGEST10	Occlusion of any origin	1=Yes; 0=No
Pathology code	DIGEST11	Esophagitis and gastroesophageal reflux disease	1=Yes; 0=No
Pathology code	DIGEST12	Acute pancreatitis and other pancreas	1=Yes; 0=No
Pathology code	DIGEST13	Peritonitis any origin	1=Yes; 0=No
Pathology code	DIGEST14	proctology	1=Yes; 0=No
Pathology code	DIGEST15	Other digestive and alimentary diseases	1=Yes; 0=No
Pathology code	GAUX01	AEG, asthenia, sliding syndrome, ..	1=Yes; 0=No
Pathology code	GAUX02	Anemia, aplasia, other hematological disorder	1=Yes; 0=No
Pathology code	GAUX03	Hydro-electrolyte dehydration and turbidity	1=Yes; 0=No
Pathology code	GAUX04	Diabetes and blood sugar disorders	1=Yes; 0=No
Pathology code	GAUX05	Unspecified acute and chronic pain, palliative care	1=Yes; 0=No
Pathology code	GAUX06	Other pathologies and general signs	1=Yes; 0=No
Pathology code	INFECTIO01	Fever	1=Yes; 0=No
Pathology code	INFECTIO02	Influenza	1=Yes; 0=No
Pathology code	INFECTIO03	Septicemia and sepsis	1=Yes; 0=No
Pathology code	INFECTIO04	Subject in contact with a communicable disease	1=Yes; 0=No
Pathology code	INFECTIO05	Other general and unspecified infections	1=Yes; 0=No
Pathology code	INTOX01	Alcohol poisoning	1=Yes; 0=No
Pathology code	INTOX02	Carbon monoxide poisoning	1=Yes; 0=No
Pathology code	INTOX03	Drug Intoxication	1=Yes; 0=No
Pathology code	INTOX04	Poisoning by other substances	1=Yes; 0=No
Pathology code	MB_ELUX	Sprains and limb dislocations	1=Yes; 0=No
Pathology code	MB_FRACT	Member Fractures	1=Yes; 0=No
Pathology code	MSLV01	Disorders without PC or unspecified	1=Yes; 0=No
Pathology code	MSLV02	Syncope, lipothymia and malaise with PC	1=Yes; 0=No
Pathology code	MSLV03	Dizziness and dizziness	1=Yes; 0=No
Pathology code	NEURO01	Cranial nerve damage	1=Yes; 0=No
Pathology code	NEURO02	Stroke, TIA, Hemiplegia and Related Syndromes	1=Yes; 0=No
Pathology code	NEURO03	Comas, tumors, encephalopathies and other SNC disease	1=Yes; 0=No
Pathology code	NEURO04	Disorientation and cognitive impairment	1=Yes; 0=No
Pathology code	NEURO05	Epilepsy and seizures	1=Yes; 0=No
Pathology code	NEURO06	Meningitis, meningitis, encephalitis and SNC infections	1=Yes; 0=No
Pathology code	NEURO07	Migraine and Headache	1=Yes; 0=No
Pathology code	NEURO08	Sensitive, motor and tonic disorders other	1=Yes; 0=No
Pathology code	ORLOS01	Angina, tonsillitis, rhinopharyngitis, cough	1=Yes; 0=No
Pathology code	ORLOS02	Dental pain, stomatology	1=Yes; 0=No
Pathology code	ORLOS03	Eye pain, conjunctivitis, other ophthalmic	1=Yes; 0=No
Pathology code	ORLOS04	Epistaxis	1=Yes; 0=No
Pathology code	ORLOS05	Laryngitis, tracheitis and other laryngeal disease	1=Yes; 0=No
Pathology code	ORLOS06	Otalgia, ear infections and other otological pathologies	1=Yes; 0=No
Pathology code	ORLOS07	Acute and chronic sinusitis	1=Yes; 0=No

Continued on next page

Table A.1 – continued from previous page

Category	Features	Label	Values
Pathology code	ORLOS08	Other disorders of the upper respiratory tract	1=Yes; 0=No
Pathology code	PLAIES_CE	Wounds and foreign bodies cutaneo-mucosa	1=Yes; 0=No
Pathology code	PNEUMO01	Asthma	1=Yes; 0=No
Pathology code	PNEUMO02	BPCO and chronic respiratory failure	1=Yes; 0=No
Pathology code	PNEUMO03	Acute bronchitis and bronchiolitis	1=Yes; 0=No
Pathology code	PNEUMO04	Dyspnea and respiratory gene	1=Yes; 0=No
Pathology code	PNEUMO05	Pulmonary embolism	1=Yes; 0=No
Pathology code	PNEUMO06	hemoptysis	1=Yes; 0=No
Pathology code	PNEUMO07	Acute respiratory insufficiency	1=Yes; 0=No
Pathology code	PNEUMO08	Pleurisy and pleural effusion	1=Yes; 0=No
Pathology code	PNEUMO09	pneumonia	1=Yes; 0=No
Pathology code	PNEUMO10	Non-traumatic Pneumothorax	1=Yes; 0=No
Pathology code	PNEUMO11	Other lower airway involvement	1=Yes; 0=No
Pathology code	PSY01	Agitation, personality and behavioral disorder	1=Yes; 0=No
Pathology code	PSY02	Anxiety, stress, neurotic or somatoform disorder	1=Yes; 0=No
Pathology code	PSY03	Depression and Mood Disorders	1=Yes; 0=No
Pathology code	PSY04	Schizophrenia, delirium, hallucinations	1=Yes; 0=No
Pathology code	RHUMATO01	Arthralgia, arthritis, tendonitis, ...	1=Yes; 0=No
Pathology code	RHUMATO02	Cervical gland, neuralgia and other cervical involvement	1=Yes; 0=No
Pathology code	RHUMATO03	Dorsalgia and spinal pathology	1=Yes; 0=No
Pathology code	RHUMATO04	Pain of limb, contracture, myalgia, ...	1=Yes; 0=No
Pathology code	RHUMATO05	Chest wall pain	1=Yes; 0=No
Pathology code	RHUMATO06	Lumbago, lumbosclerosis, lumbar spine	1=Yes; 0=No
Pathology code	RHUMATO07	Other rheumatoid and peripheral nervous system	1=Yes; 0=No
Pathology code	TRAU_COTES	Sprain, fractures and costo-sternal lesions	1=Yes; 0=No
Pathology code	TRAU_CRANE	Cranial trauma	1=Yes; 0=No
Pathology code	TRAU_ODM	OPN fractures, jaw teeth and lesions	1=Yes; 0=No
Pathology code	TRAU_OPHT	Lesions of the eye or orbit	1=Yes; 0=No
Pathology code	TRAU_PROF	Prof lesion of the tissues (tendons, vx, nerves, ..) or internal organs (excluding TC)	1=Yes; 0=No
Pathology code	TRAU_RACHIS	Sprains, dislocations and fractures of the spine or pelvis	1=Yes; 0=No
Pathology code	TRAU_SP	Other and unspecified trauma	1=Yes; 0=No
Pathology code	UROGEN01	Renal colic and urinary stones	1=Yes; 0=No
Pathology code	UROGEN02	Pelvic Pain	1=Yes; 0=No
Pathology code	UROGEN03	Testicular pain and other andrology	1=Yes; 0=No
Pathology code	UROGEN04	GEU, miscarriage, obstetric haemorrhage	1=Yes; 0=No
Pathology code	UROGEN05	hematuria	1=Yes; 0=No
Pathology code	UROGEN06	Urinary Tract Infection	1=Yes; 0=No
Pathology code	UROGEN07	Renal failure	1=Yes; 0=No
Pathology code	UROGEN08	Méno - métrorragie and other genital haemorrhage	1=Yes; 0=No
Pathology code	UROGEN09	Prostatitis, orchi-epididymitis	1=Yes; 0=No
Pathology code	UROGEN10	Urinary retention, bp probe, dysuria	1=Yes; 0=No
Pathology code	UROGEN11	Vulvo-vaginitis, salpingitis and other gynecology	1=Yes; 0=No
Pathology code	UROGEN12	Other uro-genital affection	1=Yes; 0=No
Pathology code	UROGEN13	Other obstetric remedies	1=Yes; 0=No
Code Discipline / Topography	AUTRE	Other emergencies	1=Yes; 0=No

Continued on next page

Table A.1 – continued from previous page

Category	Features	Label	Values
Code Discipline / Topography	CARDIOV	Chest pain, cardiovascular disease	1=Yes; 0=No
Code Discipline / Topography	DERMATO	Dermato-allergology and cutaneous-mucosal disorders	1=Yes; 0=No
Code Discipline / Topography	DIGESTIF	Abdominal pain, digestive pathologies	1=Yes; 0=No
Code Discipline / Topography	GAUX	General signs and other pathologies	1=Yes; 0=No
Code Discipline / Topography	INFECTIO	General fever and infectiology	1=Yes; 0=No
Code Discipline / Topography	INTOX	Acute non-food poisoning	1=Yes; 0=No
Code Discipline / Topography	MLSV	Malaise, lipothymia, syncope, dizziness and dizziness	1=Yes; 0=No
Code Discipline / Topography	NEURO	Headache, non-SNP neurological pathologies	1=Yes; 0=No
Code Discipline / Topography	ORLOS	ORL, ophtalmo, stomato and aero-digestive crossroads	1=Yes; 0=No
Code Discipline / Topography	PNEUMO	Dyspnea, pathologies of the lower airways	1=Yes; 0=No
Code Discipline / Topography	PSY	Psychiatric disorders, psychiatric disorders	1=Yes; 0=No
Code Discipline / Topography	RHUMATO	Rheumatology, Orthopedics, SNP	1=Yes; 0=No
Code Discipline / Topography	TRAU_MINF	Traumatology of the lower limb	1=Yes; 0=No
Code Discipline / Topography	TRAU_MSUP	Traumatology of the upper limb	1=Yes; 0=No
Code Discipline / Topography	TRAU_SP	Other and unspecified trauma	1=Yes; 0=No
Code Discipline / Topography	TRAU_TETEC	Traumatology of the head and neck	1=Yes; 0=No
Code Discipline / Topography	TRAU_TRONC	Thoracic-abdominal-pelvic traumatology	1=Yes; 0=No
Code Discipline / Topography	UROGEN	Pelvic pain, uro-genital diseases	1=Yes; 0=No
Code Type Urgences	AUTRE	Other recourse	1=Yes; 0=No
Code Type Urgences	MED-CHIR	Medical and surgical	1=Yes; 0=No
Code Type Urgences	TRAUMA	Traumatology	1=Yes; 0=No
Code Type Urgences	TOXICO	Toxicological	1=Yes; 0=No
Code Type Urgences	PSY	Psychiatric	1=Yes; 0=No
Personal information	Sex	Patient's gender	F=Female;M=Male
Inpatient episode	Age	Patient's age at admission	Below;Mean;Over
Inpatient episode	Duree	Time interval between admission date and discharge date	Below;Mean;Over
Inpatient episode	ModeEntree	Patient's admission type	1=Emergency; 2=Urgent; 3=Elective; 4=Newborn; 5=Trauma; 9=Information not available

Continued on next page

Table A.1 – continued from previous page

Category	Features	Label	Values
Inpatient episode	Provenance	The place where the patient is coming from	1=Acute care unit; 2= Rehabilitation unit 3=Long-term care unit; 4=Psychiatric unit; 5=Passing through the institution's emergency facility; 6=Hospitalized at home
Inpatient episode	ModeSortie	Patient's discharge status	1=Discharge to home; 2=Transferred to short-term facility; 3=Transferred to skilled nursing facility; 4=Transferred to intermediate care facility; 5=Transferred to other healthcare facility; 6=Transferred to home health care; 7=Left AMA(Against Medical Advice); 20=Expired/Mortality
Inpatient episode	Destination	The place where the patient is going after the discharge	1=Acute Care Unit; 2=Rehabilitation unit; 3=Long Term Care Unit 4=Psychiatric unit; 6=home hospitalization; 7=Medico-social housing structure
Inpatient episode	Season	The season at the admission	Summer; Winter; Fall; Spring
Inpatient episode	Frequency	The count of the inpatient episodes of the patient during his life.	Below; Mean; Over
Inpatient episode	Delay	Time interval between admission date and first medical procedure	Below; Mean; Over
Inpatient episode	Inpatient transfer count	The count of the transfers between medical units in the inpatient episode	Below; Mean; Over
Inpatient episode	NbreDAS	The count of the diagnoses in the inpatient episode	Below; Mean; Over
Inpatient episode	NbreActe	The count of the medical procedures during the inpatient episode	Below; Mean; Over
Inpatient episode	Classified	A flag indicating whether the inpatient stay has a classified/important medical procedure or not.	1=Yes; 0=No
Inpatient episode	Emergency	A flag indicating whether the inpatient stay has an emergency case or not.	1=Yes; 0=No
Medical procedure	Chap1	Central, peripheral and autonomous nervous system	1=Yes; 0=No
Medical procedure	Chap2	Eye and appendices	1=Yes; 0=No
Medical procedure	Chap3	Hearing	1=Yes; 0=No
Medical procedure	Chap4	Circulatory apparatus	1=Yes; 0=No
Medical procedure	Chap5	Immune system and hematopoietic system	1=Yes; 0=No
Medical procedure	Chap6	Respiratory system	1=Yes; 0=No
Medical procedure	Chap7	Digestive	1=Yes; 0=No
Medical procedure	Chap8	Urinary and genital apparatus	1=Yes; 0=No
Medical procedure	Chap9	Acts relating to procration, pregnancy and new born	1=Yes; 0=No

Continued on next page

Table A.1 – continued from previous page

Category	Features	Label	Values
Medical procedure	Chap10	Endocrine glands and metabolism	1=Yes; 0=No
Medical procedure	Chap11	Osteoarticular and muscle apparatus of the head	1=Yes; 0=No
Medical procedure	Chap12	Osteoarticular and muscle apparatus of the neck and of the trunk	1=Yes; 0=No
Medical procedure	Chap13	Oseoarticular and muscle apparatus of the superior member	1=Yes; 0=No
Medical procedure	Chap14	Oseoarticular and muscle apparatus of the lower member	1=Yes; 0=No
Medical procedure	Chap15	Osteoarticular and muscle apparatus, without topographic precision	1=Yes; 0=No
Medical procedure	Chap16	Tegumentary system - mammary gland	1=Yes; 0=No
Medical procedure	Chap17	Acts without topographic precision	1=Yes; 0=No
Medical procedure	Chap18	Complementary anestheties and complementary gestures	1=Yes; 0=No
Medical procedure	Chap19	Adaptations for the transitional CCAM	1=Yes; 0=No

Table A.2 Evaluation of CFS stable features (excluding diagnoses related features) using NB and CART classifiers

		CART Decision Tree								Naive Bayes							
		Local PMSI				Regional PMSI				Local PMSI				Regional PMSI			
DP-DS	**CFS Features**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**
F05_E44	ModeSortie Chap02 Chap08 Chap14	58%	60%	57%	59%	34%	50%	40%	30%	53%	44%	48%	60%	56%	55%	52%	62%
F05_R26	AgeAn Chap06	63%	57%	55%	76%	64%	56%	54%	79%	52%	54%	54%	50%	53%	55%	49%	57%
I50_R26	Chap01	53%	65%	80%	40%	30%	57%	76%	19%	64%	65%	76%	55%	28%	54%	56%	19%
J15_E44	AgeAn Chap01	64%	70%	72%	57%	27%	52%	55%	18%	65%	64%	57%	76%	54%	55%	46%	65%
J15_R26	AgeAn Chap01	56%	61%	64%	50%	28%	52%	56%	18%	70%	65%	65%	76%	57%	60%	52%	64%
J18_E44	Chap01	67%	73%	69%	64%	48%	58%	63%	39%	36%	54%	47%	29%	21%	51%	45%	14%
J20_R26	Chap01 Chap07	35%	56%	67%	24%	24%	55%	75%	14%	42%	58%	73%	29%	36%	54%	53%	27%
K80_E66	Duree	50%	50%	43%	60%	49%	56%	45%	54%	45%	40%	44%	45%	59%	56%	55%	64%
R10_E66	Provenance Chap06 Chap16 Chap19	50%	50%	45%	55%	39%	57%	40%	39%	53%	57%	70%	42%	46%	57%	56%	39%
R29_E44	AgeAn NbreActe Chap01	67%	64%	61%	73%	50%	53%	50%	51%	65%	71%	60%	72%	43%	50%	48%	40%
R41_E44	Destination Duree Chap01	42%	58%	67%	31%	65%	65%	65%	65%	62%	61%	63%	62%	51%	56%	51%	51%

Table A.3 Evaluation of GainR stable features (excluding diagnoses related features) using NB and CART classifiers

		CART Decision Tree								Naive Bayes							
		Local PMSI				Regional PMSI				Local PMSI				Regional PMSI			
DP-DS	**CFS Features**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**	**F1**	**AUC**	**Prec**	**Rec**
F05_E44	Destination Chap01 Provenance ModeSortie Sexe	56%	56%	53%	60%	67%	49%	50%	99%	60%	60%	54%	67%	_	_	_	_
F05_R26	Duree ModeSortie Destination Chap06 Provenance Chap16 AgeAn	54%	57%	58%	50%	62%	43%	51%	81%	45%	57%	47%	43%	40%	53%	40%	41%
I48_E66	AgeAn Provenance Destination	36%	50%	40%	33%	47%	55%	56%	40%	56%	44%	48%	67%	49%	44%	48%	50%
I50_R26	Chap01 Duree Destination ModeSortie AgeAn Chap19 Sexe	55%	60%	61%	50%	30%	56%	76%	19%	71%	70%	72%	70%	0%	?	0%	0%
J15_E44	Chap01 ModeSortie Destination AgeAn Chap04 Chap14	57%	57%	57%	57%	27%	52%	55%	18%	58%	60%	55%	62%	54%	59%	57%	52%
J15_R26	Chap01 AgeAn Destination Duree ModeSortie Chap05 Provenance ModeEntree	56%	62%	63%	51%	29%	52%	56%	19%	62%	66%	60%	64%	52%	68%	67%	42%
J20_R26	Chap01 NbreActe ModeSortie Chap08 Destination Chap19 Chap07 Duree AgeAn Chap16 Provenance	48%	55%	55%	43%	46%	53%	53%	41%	50%	57%	65%	41%	57%	62%	57%	58%
J69_R26	Chap19 Chap01 Destination Provenance	63%	65%	62%	65%	49%	59%	66%	40%	49%	60%	55%	44%	_	_	_	_
J18_E44	Chap01 Destination AgeAn ModeSortie Duree Provenance	59%	60%	57%	62%	57%	55%	55%	59%	54%	58%	58%	50%	_	_	_	_
K80_E66	ModeSortie Destination	43%	57%	63%	33%	32%	52%	54%	23%	59%	44%	50%	72%	7%	51%	58%	3%
R10_E66	Destination ModeSortie Chap04 Chap12 AgeAn Duree Chap19 NbreActe Chap06	50%	66%	78%	37%	51%	62%	72%	39%	50%	60%	62%	42%	47%	54%	52%	42%
R29_E44	Chap01 NbreActe AgeAn ModeSortie Destination	64%	63%	65%	63%	40%	44%	44%	36%	64%	65%	54%	80%	51%	52%	51%	51%
R41_E44	Duree Destination AgeAn Chap01 NbreActe Chap19 ModeSortie	56%	46%	48%	67%	50%	42%	44%	58%	64%	70%	61%	68%	51%	60%	58%	45%
S72_L89	Chap01 Destination ModeSortie AgeAn Provenance Chap17 ModeEntree NbreActe Sexe	52%	50%	50%	55%	67%	50%	50%	100%	63%	58%	57%	70%	_	_	_	_

Appendix B

Observation notes

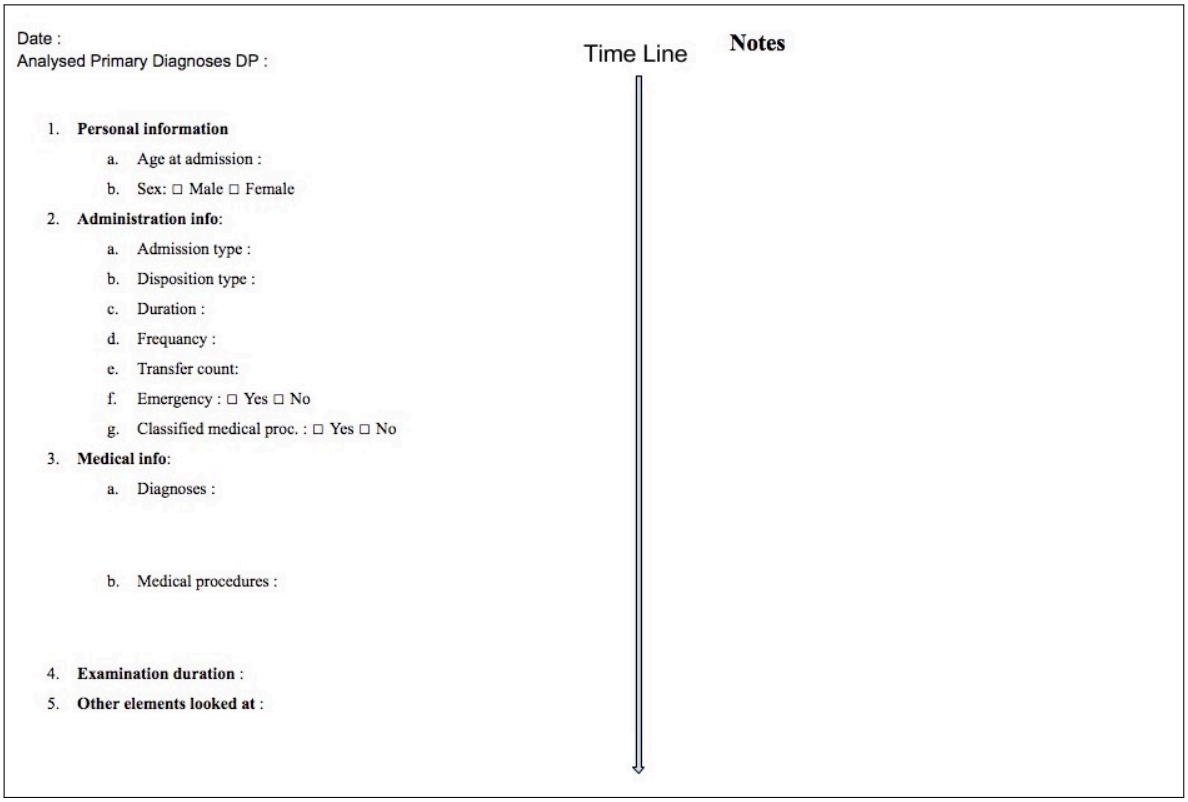


Fig. B.1 The used sheet to take observation notes

Table B.1 Observation notes

N.	Date	Coder	DP	codes	DS	DS count	Medical proc.	Medical proc. ct.	Age	Gender	Admission type	Disposition type	Duration day	Frequency	Emergency	Classified proc.	Obs duration	Information sources	Other
1	07/03/2017		J691	NEURO02; CARDIOV13; NEURO08; CARDIOV14; PSY02; GAUX05; GAUX05; INFECTIO03; GAUX06; INFECTIO05; AUTRE07; NEURO04	G811; I489; R26; I831; F418; R52; Z515; R650; E441; Z290; Z822+1; R418	12	-	-	86	F	Urgent	Domicile	18	3	1	0	20	Discharge letter- lab-EMR	weight-height
2	07/03/2017		J691	DIGEST03; RHUMATO01; CARDIOV06; CARDIOV13; PNEUMO07; PNEUMO07; NEURO04; PSY01; NEURO03; AUTRE01; PSY01; GAUX03	k573 ;M179; I10; I493; J960; J9609; F03+02; F80; R4018; R4018; Y450; R451; E8700	13	-	2	80	M	Urgent	Retirement home	1	6	1	1	15	Discharge letter- lab-EMR	-
3	07/03/2017		F051	GAUX06; RHUMATO07; CARDIOV06; DIGEST03; NEURO04; NEURO04	E059; M8058; I10; K590; F00102; G301	6	-	-	90	F	Urgent	Domicile	7	2	1	0	8	Discharge letter- lab-EMR	traitement-weight-height
4	07/03/2017		F051	AUTRE04; ORLOS06; CARDIOV13; CARDIOV06; AUTRE06; AUTRE07; AUTRE07; DIGEST03; DIGEST05; DIGEST14; NEURO04; NEURO04	Z742; H911; I456; I119; Z966; Z854; Z907; K590; R104; k625; F00102; G301	12	-	4	78	M	Planned	Mutation	5	1	0	0	8	Discharge letter- lab-EMR	traitement-weight-height
5	07/03/2017		J180	CARDIOV06; CARDIOV01; AUTRE06; AUTRE06; GAUX04; PNEUMO02; AUTRE07; UROGEN06; INFECTIO05; NEURO03; NEURO04; NEURO04; NEURO04; NEURO04; NEURO08; NFECTIO05; PNEUMO07; GAUX06	I119; I255; Z9580; Z950; E1190; J448; Z921; N390; B952; I673; F0010; G301; F0180; R410; R261; B962; J9600; E441	18	-	4	92	M	Urgent	Domicile	12	3	1	0	16	Discharge letter- lab-EMR	albomine-IMC
6	07/03/2017		J180	CARDIOV13; AUTRE07; PNEUMO07; CARDIOV14; CARDIOV06; RHUMATO07; CARDIOV14; DERMATO06; CARDIOV14; PNEUMO08; GAUX06; GAUX01; AUTRE0; NEURO04; DERMATO10; PSY01; GAUX05; RHUMATO01	I480; Z921; J960; I350; I10; M316; I7020; R605; I831; J90; E43; R2630; Y442; F03; L890; F918; R522; M171	18	AL; ZF; ZF	3	97	M	Urgent	Retirement home	17	3	1	0	14	Discharge letter- lab-EMR	-
7	07/03/2017		J690	NEURO04; GAUX01; AUTRE06; AUTRE07; PNEUMO0; GAUX06	F03+02; R2630; Z966; Z853; J9600; E441	6	ZF; ZF	2	97	M	Urgent	Death	4	1	1	1	10	Discharge letter- lab-EMR	-
8	07/03/2017		F0010	CARDIOV01; CARDIOV14; CARDIOV14; CARDIOV13; CARDIOV06; NEURO03; GAUX06; AUTRE01; UROGEN10; GAUX03; GAUX03; PNEUMO09; NEURO04; NEURO04; CARDIOV09; GAUX06	I255; I340; I341; I482; I10; I693; E43; Y442; R33; E8700; E86; J690; G301; F0180; I509; E559	16	AL; AL; ZF; JD	4	92	M	transfer	Mutation	25	2	0	1	11	Discharge letter- lab-EMR	medical procedure
9	07/03/2017		J13	AUTRE06; AUTRE07; GAUX06; PNEUMO07; DIGEST15; GAUX03; DERMATO06	Z998; Z922; E039; J9600; R630; E8718; R60	7	-	1	83	F	Urgent	Domicile	5	6	1	1	9	Discharge letter- lab-EMR	-
10	07/03/2017		J13	PSY03; PSY03; PNEUMO08 ;PNEUMO08; UROGEN07; GAUX01; DIGEST15; INFECTIO03; AUTRE06	F329; F319; Z998; J91; R630; R53+2; N179; E8718; J90; R650	10	G; G; G; G; G; G; G; G; G; D	9	62	M	Urgent	Domicile	11	2	1	0	12	Discharge letter- lab-EMR	-
11	07/03/2017		J440 ou J181	GAUX03; PNEUMO09; PNEUMO07; PNEUMO02; DIGEST03; DIGEST08; RHUMATO03; DERMATO10; AUTRE07; AUTRE06; AUTRE06	J159; J96100; Z998; Z991+1; M485; Z911; E872; J9600; K590; K802; R61	11	GL	1	82	M	Urgent	Domicile	7	-	1	1	12	Discharge letter- lab-EMR	-
12	07/03/2017		J159	GAUX06; GAUX04; CARDIOV06; CARDIOV13; CARDIOV14; PNEUMO07; DIGEST09; NEURO04; MSLV01	E1190; I10; I482; E039; R53+1; R11; R410; I831; I482; J9600	10	GL;GL	2	84	F	Mutation	Domicile	8	4	0	1	10	Discharge letter- lab-EMR	prescription
13	07/03/2017		J13	CARDIOV06; CARDIOV01; CARDIOV09; PNEUMO02; PNEUMO08; PNEUMO07: ORLOS02; AUTRE04; AUTRE06	I501; I255; I10; J91; J47; Z998; Z501; J9600; k120	9	GL; GL; GL	3	89	F	réanimation	Domicile	26	1	0	1	15	Discharge letter- lab-EMR	hospitalisation
14	07/03/2017		F051	NEURO04; NEURO04; ORLOS06; CARDIOV06; CARDIOV13; CARDIOV07; DIGEST10; GAUX01	I10; H919; I491; K564; R296; I951; F00102; G301	8	ZF; ZF; AL; AL	4	93	M	directe	Mutation	10	4	0	1	14	Discharge letter- lab-EMR	-

Continue Table B.1

N.	Date	Coder	DP	codes	DS	DS count	Medical proc.	Medical proc. ct.	Age	Gender	Admission type	Disposition type	Duration day	Frequency	Emergency	Classified proc.	Obs duration	Information sources	Other
15	07/03/2017		J209	NEURO04; NEURO04; NEURO04; NEURO05; NEURO03; CARDIOV14; NEURO04; INFECTIO03; AUTRE02; AUTRE07	I7020; G409; Z904; I691; Z713; R650; F00102; G301; F01802; R410	10	ZF;ZF	2	87	F	Urgent	Domicile	8	12	1	1	10	Discharge letter- lab- EMR	-
16	07/03/2017		F00101	INFECTIO05; GAUX04; GAUX03; CARDIOV06; NEURO02; CARDIOV14; AUTRE04; AUTRE06	I652; I7020; E1190; I10; Z966; Z742; E875; B962	8	AL; AL; ZF; ZF	4	79	F	Urgent	Domicile	9	1	1	1	10	Discharge letter- lab- EMR	-
17	07/03/2017		J189	GAUX06; NEURO03; DERMATO10; UROGEN12; TRAU_SP; TRAU_SP	N40; H949; T796; E85; I673; W190; L890	7	AL; AL; ZF; ZF	4	98	H	Urgent	transfer de sejour	11	3	1	1	8	Discharge letter- lab- EMR	-
18	07/03/2017		J189	GAUX06; GAUX06; NEURO04; NEURO04; CARDIOV06; RHUMATO03; TRAU_SP; AUTRE07; AUTRE07	E050; I10; Z907; Z904; E6600; W190; M4780; F00102; G301; C308	10	AA; AL; AL; ZF	4	86	F	Urgent	Home	11	8	1	1	10	Discharge letter- lab- EMR	-
19	08/03/2017		F051	PSY01; PNEUMO09; DERMATO06; GAUX05; AUTRE04	Z742; L880; F070; J180; Z515; R220	6	ZF;ZF	2	80	H	Urgent	Death	11	2	1	1	11	Discharge letter- lab- EMR	-
20	08/03/2017		F051	GAUX06; NEURO04; NEURO04; DIGEST04; AUTRE06	Z081; R299; I851; K580; F00102; G301; E43	7	ZF;ZF	2	88	F	directe	Mutation	3	7	0	1	6	Discharge letter- lab- EMR	-
21	08/03/2017		F070	INFECTIO05; INFECTIO05; INFECTIO05; GAUX06; GAUX06; GAUX03; CARDIOV06; CARDIOV13; CARDIOV07; UROGEN07; MSLV01; DIGEST15; AUTRE01; AUTRE04	Z742; E780; I10; I19; N189; R630; E441; R53+1; I495; N380; B962; I95; B338; Y517; E8758; B951	16	A; A; A; D; Z	5	89	F	Urgent	Domicile	12	2	1	1	10	Discharge letter- lab- EMR	-
22	08/03/2017		F051	DERMATO10; GAUX04; NEURO04; NEURO04; CARDIOV01; CARDIOV14; CARDIOV14; CARDIOV13; DERMATO02; AUTRE06	I255; Z955; I340; I350; E1190; I440; L300; F00201; G301; C444	10	AL;AL;ZF	3	94	H	Planned	Retirement home	7	2	0	1	10	Discharge letter- lab- EMR	-
23	08/03/2017		F051	GAUX06; NEURO04; NEURO04; NEURO04; CARDIOV06; CARDIOV01; CARDIOV14; DIGEST04	I255; I359; I10; E039; K580; F00102; G301; F01002	8	ZF; ZF; AL	3	82	H	Urgent	Domicile	5	3	1	1	6	Discharge letter- lab- EMR	-
24	08/03/2017		R410	DERMATO05; ORLOS02; GAUX06; GAUX06; GAUX03; GAUX03; CARDIOV06; CARDIOV13; CARDIOV14; CARDIOV14; INFECTIO02; PNEUMO02; DIGEST04; RHUMATO06; DIGEST09; ORLOS01; DIGEST15; GAUX01; UROGEN10; CARDIOV03; CONT_LS	I10; R296; J448; I899; I872; R33; I493; K654; S000; E43; R13; E872; E8700; R571; E85; R11; R393; K529; F0+0; J111; M4856; R195; B368; B370	24	-	-	91	H	Urgent	Retirement home	26	1	1	1	9	Discharge letter- lab- EMR	-
25	08/03/2017		J189	NEURO04; PSY02; CARDIOV06; CARDIOV01; CARDIOV13	F03+02; I10; I255; I489; F412	5	-	0	93	F	Urgent	Domicile	12	2	1	1	9	Discharge letter- lab- EMR	-
26	08/03/2017		F051	PSY02; ORLOS03; CARDIOV06; CARDIOV13; AUTRE04	I10; H353; Z742; I493; F412	5	AL;AL;ZF	3	93	H	Urgent	Retirement home	20	3	1	1	8	Discharge letter- lab- EMR	-
27	09/03/2017		J158+D963+I10	PSY01; PNEUMO02; PNEUMO02; INFECTIO03; AUTRE07	F17202; J439; J448; Z922; R650	5	GL; GL; GL	3	69	M	Urgent	Mutation	9	3	1	1	14	Discharge letter- lab- EMR	-
28	09/03/2017		J159	GAUX06; GAUX06; CARDIOV06; CARDIOV01; PNEUMO07; INFECTIO03; CONT_LS; AUTRE07; AUTRE06; AUTRE06; AUTRE06; AUTRE06	J9600; Z998; I255; E039; I10; E782; Z955; Z966; Z922; Z951; T210; R651	12	GL;GL	2	84	M	Urgent	Mutation	9	4	1	1	11	Discharge letter- lab- EMR	-
29	09/03/2017		J181	GAUX04; CARDIOV06; CARDIOV14; CARDIOV13; CARDIOV13; PNEUMO02; UROGEN12; INFECTIO03	I481; E1190; I10; N40; J440; R042,J9600; I451; R651; I350	9	DZ; DZ; GE; DF	4	82	M	Urgent	Mutation	11	4	1	1	10	Discharge letter- lab- EMR	-
30	09/03/2017		J181	GAUX04; PSY01; PSY03; CARDIOV06; CARDIOV09; AUTRE06															

Appendix C

PMSI structure

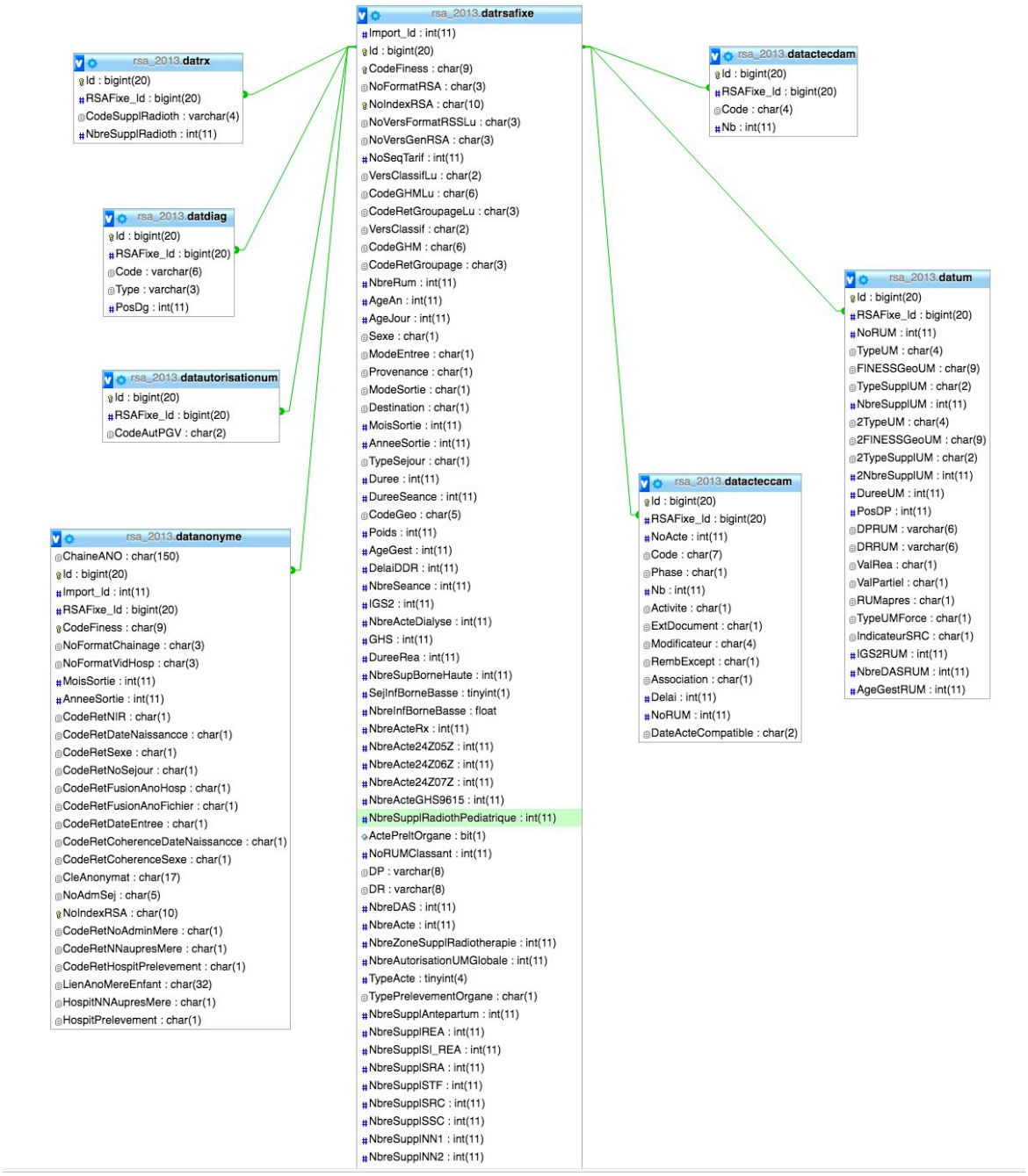


Fig. C.1 The PMSI database relational model

References

Thomas Abeel, Thibault Helleputte, Yves de Peer, Pierre Dupont, and Yvan Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2010.

Mehmet Fatih Akay. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2 PART 2):3240–3247, 2009. ISSN 09574174. doi: 10.1016/j.eswa.2008.01.009.

Ali Al-Shahib, Rainer Breitling, and David Gilbert. Feature selection and the class imbalance problem in predicting protein function from sequence. *Applied Bioinformatics*, 4 (3):195–203, 2005.

Salem Alelyani, Jiliang Tang, and Huan Liu. Feature Selection for Clustering: A Review. *Data Clustering: Algorithms and Applications*, 29:110–121, 2013.

Carlos J. Alonso-González, Q. Isaac Moro, Oscar J. Prieto, and M. Aránzazu Simón. Selecting Few Genes for Microarray Gene Expression Classification. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5988 LNAI, pages 111–120. 2010. ISBN 364214263X. doi: 10.1007/978-3-642-14264-2_12.

Fabrizio Angiulli. Fast Condensed Nearest Neighbor Rule. In *Proceedings of the 22nd international conference on Machine learning*, pages 25–32. ACM, 2005.

Alan R. Aronson, Olivier Bodenreider, Dina Demner-Fushman, Kin Wah Fung, Vivian K. Lee, James G. Mork, Aurélie Névéol, Lee Peters, and Willie J. Rogers. From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches. pages 105–112, 2007.

ATIH. Guide Méthodologique De Production Des Informations Relatives et à sa Facturation en MCO. 2016a.

ATIH. Guide Méthodologique De Production Des Recueils D'informations Standardisés De L'hospitalisation à Domicile. 2016b.

ATIH. Guide Méthodologique De Production Du Recueil D'informations Médicalisé en Psychiatrie. 2016c.

ATIH. Guide Méthodologique De Production Des Informations Relatives à L'activité Médicale Et à Sa Facturation En Soins De Suite Et De Réadaptation. 2016d.

Ricardo Barandela, Rosa Maria Valdovinos, and José Salvador Sánchez. New applications of ensembles of classifiers. *Pattern Analysis & Applications*, 6(3):245–256, 2003.

Rukshan Batuwita and Vasile Palade. Adjusted geometric-mean: a novel performance measure for imbalanced bioinformatics datasets learning. *Journal of Bioinformatics and Computational Biology*, 10(04):1250003, 2012.

Amir Ben-Dor, Laurakay Bruhn, Nir Friedman, Iftach Nachman, Michèl Schummer, and Zohar Yakhini. Tissue classification with gene expression profiles. *Journal of computational biology*, 7(3-4):559–583, 2000.

Albert Bifet and Ricard Gavaldà. *Adaptive Learning from Evolving Data Streams*, pages 249–260. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-03915-7. doi: 10.1007/978-3-642-03915-7_22.

Albert Bifet and Richard Kirby. Data Stream Mining. A Practical Approach, 2009.

Jose Bins and Bruce A Draper. Feature Selection from Huge Feature Sets. pages 159–165, 2001.

Jeffrey P Bradford, Clayton Kunz, Ron Kohavi, Cliff Brunk, and Carla E Brodley. Pruning decision trees with misclassification costs. In *European Conference on Machine Learning*, pages 131–136. Springer, 1998.

Leo Breiman. RANDOM FORESTS. *Machine Learning*, 45(1):5–32, 2001. ISSN 08856125. doi: 10.1023/A:1010933404324.

Leo Breiman, JH Friedman, RA Olshen, , and CJ Stone. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN 0412048418.

Reinhard Busse, Alexander Geissler, and Wilm Quentin. *Diagnosis-Related Groups in Europe: Moving towards transparency, efficiency and quality in hospitals*. McGraw-Hill Education (UK), 2011.

Godwin Caruana, Maozhen Li, and Man Qi. A MapReduce based parallel SVM for large scale spam filtering. *Proceedings - 2011 8th International Conference on Fuzzy Systems*

and Knowledge Discovery, FSKD 2011, 4:2659–2662, 2011. doi: 10.1109/FSKD.2011.6020074.

Gabriele Cavallaro, Morris Riedel, Matthias Richerzhagen, Jón Atli Benediktsson, and Antonio Plaza. On understanding big data impacts in remotely sensed image classification using support vector machine methods. *IEEE journal of selected topics in applied earth observations and remote sensing*, 8(10):4634–4646, 2015.

Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers and Electrical Engineering*, 40(1):16–28, 2014. ISSN 00457906. doi: 10.1016/j.compeleceng.2013.11.024.

Nitesh Chawla, Aleksandar Lazarevic, Lawrence Hall, and Kevin Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. *Knowledge Discovery in Databases: PKDD 2003*, pages 107–119, 2003.

Nitesh V Chawla. Data Mining for Imbalanced Datasets: An Overview. In *Data Mining and Knowledge Discovery Handbook*, pages 853–867. Springer, 2005. ISBN 9780387254654. doi: 10.1007/0-387-25465-X_40.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Nitesh V. Chawla, Nathalie Japkowicz, and Prentice Drive. Editorial : Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004. ISSN 1931-0145. doi: http://doi.acm.org/10.1145/1007730.1007733.

Hui-ling Chen, Bo Yang, Jie Liu, and Da-you Liu. Expert Systems with Applications A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. *Expert Systems With Applications*, 38(7):9014–9022, 2011. ISSN 0957-4174. doi: 10.1016/j.eswa.2011.01.120.

Michal R. Chmielewski and Jerzy W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning*, 15(4):319–331, 1996. ISSN 0888613X. doi: 10.1016/S0888-613X(96)00074-6.

Grigorios Chrysos, Panagiotis Dagritzikos, Ioannis Papaefstathiou, and Apostolos Dollas. HC-CART: A parallel system implementation of data mining classification and

regression tree (CART) algorithm on a multi-FPGA system. *ACM Transactions on Architecture and Code Optimization*, 9(4):1–25, jan 2013. ISSN 15443566. doi: 10.1145/2400682.2400706.

Kyriacos Chrysostomou. Wrapper feature selection. In *Encyclopedia of Data Warehousing and Mining, Second Edition*, pages 2103–2108. IGI Global, 2009.

Nadia A Chuzhanova, Antonia J Jones, and Steve Margetts. Feature selection for genetic sequence classification. *Bioinformatics*, 14(2):139–143, 1998.

David A. Cieslak and Nitesh V. Chawla. Learning Decision Trees for Unbalanced Data. In *Machine Learning and Knowledge Discovery in Databases*, volume 5211 LNAI, pages 241–256. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Edgar F Codd. *The relational model for database management: version 2*. Addison-Wesley Longman Publishing Co., Inc., 1990.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 160–167, New York, New York, USA, jul 2008. ACM Press. ISBN 9781605582054. doi: 10.1145/1390156.1390177.

Kevin R Coombes, Keith A Baggerly, and Jeffrey S Morris. Pre-processing mass spectrometry data. In *Fundamentals of Data Mining in Genomics and Proteomics*, pages 79–102. Springer, 2007.

Cortes and Vapnik. Support Vector Networks. *Machine Learning*, 20(3):273˜–˜297, sep 1995. ISSN 08856125. doi: 10.1007/BF00994018.

Wei Dai and Wei Ji. A MapReduce Implementation of C4. 5 Decision Tree Algorithm. *International Journal of Database Theory & Application*, 7(1):49–60, 2014.

Manoranjan Dash and Yew-Soon Ong. Relief-c: Efficient feature selection for clustering over noisy data. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 869–872. IEEE, 2011.

Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Proceedings of 6th Symposium on Operating Systems Design and Implementation*, pages 137–149, 2004. ISSN 00010782. doi: 10.1145/1327452.1327492.

Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Bioinformatics and Computational Biology*, 3(2):185–206, 2005.

Mehdi Djennaoui, Grégoire Ficheur, Régis Beuscart, and Emmanuel Chazard. Improvement of the quality of medical databases: data-mining-based prediction of diagnostic codes from previous patient codes. *Studies in health technology and informatics*, 210: 419–23, 2015. ISSN 0926-9630.

Thanh Nghi Do and François Poulet. Classifying one billion data with a new distributed SVM algorithm. In *Proceedings of the 4th IEEE International Conference on Research, Innovation and Vision for the Future, RIVF'06*, pages 59–66, 2006. ISBN 1424403162. doi: 10.1109/RIVF.2006.1696420.

Carlotta Domeniconi and Dimitrios Gunopulos. Incremental support vector machine construction. *Proceedings 2001 IEEE International Conference on Data Mining*, pages 589–592, 2001. ISSN 0963-9292. doi: 10.1109/ICDM.2001.989572.

Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164. ACM, 1999.

Pedro Domingos and Michael Pazzani. On the Optimality of the Simple Bayesian Classifier under Zero-One Los. *Machine Learning*, 29(1):103–130, 1997. ISSN 08856125. doi: 10.1023/A:1007413511361.

James Dougherty, Ron Kohavi, Mehran Sahami, et al. Supervised and unsupervised discretization of continuous features. In *Machine learning: proceedings of the twelfth international conference*, volume 12, pages 194–202, 1995.

Chris Drummond and R.C. Holte. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. *Workshop on Learning from Imbalanced Datasets II*, pages 1–8, 2003. doi: 10.1.1.68.6858.

Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

Kevin Dunne, Padraig Cunningham, and Francisco Azuaje. Solutions to instability problems with sequential wrapper-based approaches to feature selection. *Journal of Machine Learning Research*, pages 1–22, 2002.

Sašo Džeroski. Relational data mining. *Data Mining and Knowledge Discovery Handbook*, pages 887–911, 2010.

Michael Egmont-Petersen, Dick De Ridder, and Heinz Handels. Image processing with neural networks- A review. *Pattern Recognition*, 35(10):2279–2301, 2002. ISSN 00313203. doi: 10.1016/S0031-3203(01)00178-9.

Charles Elkan. The foundations of cost-sensitive learning. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

Madhav Erraguntla, Belita Gopal, Satheesh Ramachandran, and Richard Mayer. Inference of Missing ICD 9 Codes Using Text Mining and Nearest Neighbor Techniques. In *2012 45th Hawaii International Conference on System Sciences*, pages 1060–1069. IEEE, 2012. ISBN 978-1-4577-1925-7. doi: 10.1109/HICSS.2012.323.

Wei Fan, Salvatore J Stolfo, Junxin Zhang, and Philip K Chan. AdaCost: misclassification cost-sensitive boosting. In *Icml*, volume 99, pages 97–105, 1999.

Richárd Farkas and György Szarvas. Automatic construction of rule-based ICD-9-CM coding systems. *BMC bioinformatics*, 9 Suppl 3:S10, jan 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-S3-S10.

Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Usama Fayyad and Ramasamy Uthurusamy. Data mining and knowledge discovery in databases. *Communication of ACM*, 39(11):24–26, 1996. ISSN 0001-0782. doi: 10.1145/240455.240463.

Jose C. Ferrao, Monica D. Oliveira, Filipe Janela, and Henrique M. G. Martins. Clinical coding support based on structured data stored in electronic health records. In *2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pages 790–797. IEEE, oct 2012. ISBN 978-1-4673-2747-3. doi: 10.1109/BIBMW.2012.6470241.

Jose C. Ferrao, Filipe Janela, Monica D. Oliveira, and Henrique M.G. Martins. Using Structured EHR Data and SVM to Support ICD-9-CM Coding. In *2013 IEEE International Conference on Healthcare Informatics*, pages 511–516. IEEE, sep 2013. ISBN 978-0-7695-5089-3. doi: 10.1109/ICHI.2013.79.

Jose C. Ferrao, Siemens Healthcare, and Filipe Janela. Predicting length of stay and assignment of diagnosis codes during hospital inpatient episodes. In *Proceedings of the First Karlsruhe Service Summit Workshop-Advances in Service Research, Karlsruhe, Germany, February 2015*, volume 7692, page 65. KIT Scientific Publishing, 2015.

Robert B Fetter. Diagnosis Related Groups: Understanding Hospital Performance. *Interfaces*, 21(1):6–26, 1991. doi: 10.1287/inte.21.1.6.

George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

Ira Goldstein, Anna Arzrumtsyan, and Ozlem Uzuner. Three approaches to automatic assignment of icd-9-cm codes to radiology reports. *AMIA. Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 279–83, 2007. ISSN 1942-597X.

Hans Peter Graf, Eric Cosatto, Leon Bottou, Igor Durdanovic, and Vladimir Vapnik. Parallel Support Vector Machines : The Cascade SVM. In *In Advances in Neural Information Processing Systems*, pages 521–528, 2005. ISBN 0262195348.

Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003. ISSN 15324435. doi: 10.1162/153244303322753616.

Isabelle Guyon, Jason Weston, S Barnhill, and VN Vapnik. Gene Selection for Cancer Classification using Support Vector Machines. *Mach Learn*, 46(1-3):389–422, 2002. doi: 10.1023/A:1012487302797.

Haibo He and E.A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, sep 2009. ISSN 1041-4347. doi: 10.1109/TKDE.2008.239.

Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017. ISSN 09574174. doi: 10.1016/j.eswa.2016.12.035.

Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: An attribute-oriented approach. *Vldb*, 92:24–27, 1992.

Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Elsevier, 2012. ISBN 978-0-12-381479-1. doi: 10.1007/978-3-642-19721-5.

Anne-Claire Haury, Pierre Gestraud, and Jean-Philippe Vert. The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures. *PLoS ONE*, 6(12), dec 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0028210.

Robert C Holte, Liane Acker, Bruce W Porter, and Others. Concept Learning and the Problem of Small Disjuncts. In *IJCAI*, volume 89, pages 813–818, 1989.

Matthijs Hovelynck and Boris Chidlovskii. Multi-modality in one-class classification. *Proceedings of the 19th international conference on World wide web - WWW '10*, (June): 441, 2010. doi: 10.1145/1772690.1772736.

Julie Jacques, Julien Taillard, David Delerue, Clarisse Dhaenens, and Laetitia Jourdan. Conception of a dominance-based multi-objective local search in the context of classification rule mining in large and imbalanced data sets. *Applied Soft Computing*, 34: 705–720, sep 2015. ISSN 15684946. doi: 10.1016/j.asoc.2015.06.002.

Krzysztof J.Cios and G.William Moore. Uniqueness of Medical Data Mining. *Artificial Intelligence in Medicine Journal*, 26(1):1–24, 2002.

Mahesh V Joshi, Vipin Kumar, and Ramesh C Agarwal. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 257–264. IEEE, 2001.

Alan Jovic, Karla Brkic, and Nicolas Bogunovic. A review of feature selection methods with applications. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, (May):1200–1205, 2015. doi: 10.1109/MIPRO.2015.7160458.

Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial Intelligence in Medicine*, 65(2):155–166, oct 2015.

Sunduz Keles, Mark J. van der Laan, and Michael B. Eisen. Identification of regulatory elements using a feature selection method. *Bioinformatics*, 18(9):1167–1175, 2002.

Josef Kittler. Pattern Recognition and Signal Processing. *On a Threshold Model, ed. by CH Chen, Amsterdam: Sijthoff and Noordhoff*, pages 41–60, 1978.

Ron Kohavi. Scale-Up the accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybird. (Utgoff 1988), 2011.

Ron Kohavi and Mehran Sahami. Error-Based and Entropy-Based Discretization of Continuous Features. pages 114–119, 1996.

François Kohler. Historique Du Pmsi En France : De 1984 À La Tarification À L'activité. 2006. URL https://www.canal-u.tv/video/canal_u_medecine/historique_du_pmsi_en_france_de_1984_a_la_tarification_a_l_activite.1818.

Sotiris Kotsiantis and Dimitris Kanellopoulos. Discretization Techniques : A recent survey. In *GESTS International Transactions on Computer Science and Engineering*, volume 32, pages 47–58. Elsevier, 2006. ISBN 9780444527813. doi: 10.4236/oalib.1100481.

Nada Lavrac and Saso Dzeroski. Inductive logic programming. In *WLP*, pages 146–160. Springer, 1994.

Laurent Lecornu, Gregoire Thillay, Clara Le Guillou, Pierre-Jean Garreau, Philippe Saliou, H Jantzem, J Puentes, and Jean M. Cauvin. REFEROCOD: a probabilistic method to medical coding support. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3421–3424. IEEE, 2009.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, may 2015. ISSN 0028-0836. doi: 10.1038/nature14539.

Jae Won Lee, Jung Bok Lee, Mira Park, and Seuck Heun Song. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, 48(4):869–885, 2005.

Bingguo Li, Xiaojun Chen, Mark Junjie Li, Joshua Zhexue Huang, and Shengzhong Feng. Scalable Random Forests for Massive Data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7301 LNAI, pages 135–146. 2012. ISBN 9783642302169. doi: 10.1007/978-3-642-30217-6_12.

Jinyan Li, Lian-sheng Liu, Simon Fong, Raymond K Wong, Sabah Mohammed, Jinan Fiaidhi, Yunsick Sung, and Kelvin KL Wong. Withdrawn: Adaptive swarm balancing algorithms for rare-event prediction in imbalanced healthcare data, 2016.

Rafael Franca Lima and Adriano C M Pereira. Feature Selection Approaches to Fraud Detection in e-Payment Systems. volume 278, pages 111–126. 2017. ISBN 978-3-319-53675-0. doi: 10.1007/978-3-319-53676-7_9.

Lucian Vlad Lita, Shipeng Yu, Stefan Niculescu, and Jinbo Bi. Large Scale Diagnostic Code Classification for Medical Patient Records. In *Proceeding sof the International Joint Conference on Natural Language Processing (IJCNLP'08)*, pages 877–882. Citeseer, 2008.

Bingwei Liu, Erik Blasch, Yu Chen, Dan Shen, and Genshe Chen. Scalable Sentiment Classification for Big Data Analysis Using NaÏve Bayes Classifier. *2013 IEEE International Conference on Big Data*, pages 99–104, oct 2013. doi: 10.1109/BigData.2013.6691740.

Tao Liu, Shengping Liu, Zheng Chen, and Wei-Ying Ma. An evaluation on feature selection for text clustering. In *Icml*, pages 488–495. 2003.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, nov 2013. ISSN 00200255. doi: 10.1016/j.ins.2013.07.007.

Jay Magidson. The chaid approach to segmentation modeling: Chi-squared automatic interaction detection. *Advanced methods of marketing research*, pages 118–159, 1994.

George D Magoulas and Andriana Prentza. Machine Learning in Medical Applications. In *Machine Learning and Its Applications*, volume 2049, pages 300–307. Springer, 2001.

Sebastián Maldonado, Juan Pérez, Richard Weber, and Martine Labbé. Feature selection for Support Vector Machines via Mixed Integer Linear Programming. *Information Sciences*, 279:163–175, sep 2014. ISSN 00200255. doi: 10.1016/j.ins.2014.03.110.

Antonio Maratea, Alfredo Petrosino, and Mario Manzo. Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*, 257(February):331–341, 2014. ISSN 00200255. doi: 10.1016/j.ins.2013.04.016.

Raúl Martín-Félez and Ramón A. Mollineda. On the Suitability of Combining Feature Selection and Resampling to Manage Data Complexity. In *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 5988 LNAI, pages 141–150. 2010. ISBN 364214263X. doi: 10.1007/978-3-642-14264-2_15.

Kate McCarthy, Bibi Zabar, and Gary Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? In *Proceedings of the 1st international workshop on Utility-based data mining*, pages 69–77. ACM, 2005.

Stephen Muggleton, Ramon Otero, and Alireza Tamaddoni-Nezhad. *Inductive logic programming*, volume 38. Springer, 1992.

Lino Murali, A Anjali, and Athira Raj. A Comparative Study of Machine Learning Approaches for Text Classification. pages 74–78, 2016.

Mario Mustra, Mislav Grgic, and Kresimir Delac. Breast Density Classification Using Multiple Feature Selection. *Automatika, Journal for Control, Measurement, Electronics, Computing and Communications*, 53(4):362–372, 2012.

Duc-Hien Nguyen and Manh-Thanh Le. Improving the Interpretability of Support Vector Machines-based Fuzzy Rules. *arXiv preprint arXiv:1408.5246*, 2014.

Wenxin Ning, Ming Yu, and Runtong Zhang. A hierarchical method to automatically encode Chinese diagnoses through semantic similarity estimation. *BMC medical informatics and decision making*, 16(1):30, 2016.

Kazuya Okamoto, Toshio Uchiyama, Tadamasa Takemura, Takayuki Adachi, Naoto Kume, Tomohiro Kuroda, Tadasu Uchiyama, and Hiroyuki Yoshihara. Automatic Selection of Diagnosis Procedure Combination Codes Based on Partial Treatment Data Relative to the Number of Hospitalization Days. *Proc. APAMI 2012*, (4):1031, 2012.

Serguei V S Pakhomov, James D Buntrock, and Christopher G Chute. Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *Journal of the American Medical Informatics Association : JAMIA*, 13(5):516–25, 2006. ISSN 1067-5027. doi: 10.1197/jamia.M2077.

Biswanath Panda, Joshua S. Herbach, Sugato Basu, and Roberto J. Bayardo. PLANET. *Proceedings of the VLDB Endowment*, 2(2):1426–1437, aug 2009. ISSN 21508097. doi: 10.14778/1687553.1687569.

Judea Pearl. Heuristics: intelligent search strategies for computer problem solving. 1984.

Suzanne Pereira, Aurélie Névéol, Philippe Massari, Michel Joubert, and Stefan Darmoni. Construction of a semi-automated ICD-10 coding help system to optimize medical and economic coding. In *Studies in health technology and informatics*, volume 124, pages 845–50, 2006. ISBN 1586036475.

Emanuel F Petricoin, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, and Others. Use of proteomic patterns in serum to identify ovarian cancer. *The lancet*, 359(9306): 572–577, 2002.

Jessica Pinaire, Julien Rabatel, Jérôme Azé, Sandra Bringay, and Paul Landais. Recherche et visualisation de trajectoires dans les parcours de soins des patients ayant eu un infarctus du myocarde, 2015.

Robi Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, 2006. ISSN 1531-636X. doi: 10.1109/MCAS.2006.1688199.

Mihail Popescu and Mohammad Khalilia. Improving disease prediction using ICD-9 ontological features. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 1805–1809. IEEE, jun 2011. ISBN 978-1-4244-7315-1. doi: 10.1109/FUZZY.2011.6007410.

C Potignon, A Musat, P Hillon, P Rat, L Osmak, D Rigaud, B Vergès, and Others. P146-Impact financier pour les établissements hospitaliers du mauvais codage PMSI de la dénutrition et de l'obésité. Étude au sein du pôle des pathologies digestives, endocriniennes et métaboliques du CHU de Dijon. 2010.

Ross Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986. ISSN 15730565. doi: 10.1023/A:1022643204877.

Ross Quinlan. Constructing decision tree. *C4*, 5:17–26, 1993.

Md Geaur Rahman and Md Zahidul Islam. Discretization of continuous attributes through low frequency numerical values and attribute interdependency. *Expert Systems with Applications*, 45:410–423, 2016.

Piyush Rai, Hal Daum, and Suresh Venkatasubramanian. Streamed learning: One-pass SVMs. *IJCAI International Joint Conference on Artificial Intelligence*, abs/0908.0:1211–1216, 2009. ISSN 10450823.

Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the Gini Index and Information Gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1): 77–93, 2004. ISSN 10122443. doi: 10.1023/B:AMAI.0000018580.96245.c6.

Habtom Ressom, Rency Varghese, D Saha, E Orvisky, L Goldman, E F Petricoim, T P Conrads, T D Veenstra, M Abdel-Hamid, C A Loffredo, and R Goldman. Particle Swarm Optimization for Analysis of Mass Spectral Serum Profiles. *Clinical Proteomics*, 21(21): 431–438, 2005.

Duda Richard, Hart Peter, and Stork David. Pattern classification. *A Wiley-Interscience*, pages 373–378, 2001.

Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine learning*, 53(1-2):23–69, 2003.

Patrick Ruch, Julien Gobeill, I Tbahriti, P Tahintzi, Christian Lovis, A Geissbühler, and F Borst. From clinical narratives to ICD codes: automatic text categorization for medico-economic encoding. *Swiss Medical Informatics*, 23(61):29–32, 2007.

Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007. ISSN 13674803. doi: 10.1093/bioinformatics/btm344.

Steven L. Salzberg. *C4.5: programs for machine learning*, volume 240. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1558602380. doi: 10.1016/S0019-9958(62)90649-6.

Robert E Schapire. A brief introduction to boosting. In *Ijcai*, volume 99, pages 1401–1406, 1999.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Marc Sebban, Richard Nock, Jean-Hugues Chauchat, and Ricco Rakotomalala. Impact of learning set quality and size on decision tree performances. *International Journal of Computers, Systems and Signals*, 1(1):85–105, 2000.

Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2010.

Daniel Shiffman. The Nature of Code: Simulating Natural Systems with Processing. *The Nature of Code*, page 520, 2012.

Wojciech Siedelecky and Jack Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(02):197–220, 1998.

Petr Somol and Jana Novovi. Evaluating Stability and Comparing Output of Feature Selectors that Optimize Feature Subset Cardinality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1921–1939, 2010.

Jyoti Soni, Ujma Ansari, Dipesh Sharma, and Sunita Soni. Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction. *International Journal of Computer Applications*, 17(8):43–48, 2011. ISSN 09758887. doi: 10.5120/2237-2860.

Mary H Stanfill, Margaret Williams, Susan H Fenton, Robert A Jenders, and William R Hersh. A systematic literature review of automated clinical coding and classification systems. *Journal of the American Medical Informatics Association : JAMIA*, 17(6):646–51, 2010. ISSN 1527-974X. doi: 10.1136/jamia.2009.001024.

Alexander Statnikov, Constantin F Aliferis, Ioannis Tsamardinos, Douglas Hardin, and Shawn Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–643, 2005.

Chao Ton Su and Chien Hsin Yang. Feature selection for the SVM: An application to hypertension diagnosis. *Expert Systems with Applications*, 34(1):754–763, 2008. ISSN 09574174. doi: 10.1016/j.eswa.2006.10.010.

Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

Zhanquan Sun and Geoffrey Fox. Study on Parallel SVM Based on MapReduce. *International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 16–19, 2012.

Johan A. K. Suykens, Marco Signoretto, and Andreas Argyriou. *Regularization, Optimization, Kernels, and Support Vector Machines*. MIT press, 2015. ISBN 9781482241402.

Mahlet Tadesse, Marina Vannucci, and Pietro Lio. Identification of DNA regulatory motifs using Bayesian variable selection. *Bioinformatics*, 20(16):2553–2561, 2004.

Xinmin Tao, Furong Liu, and Tingxian Zhou. A novel approach to intrusion detection based on support vector data description. In *Industrial Electronics Society, 2004. IECON 2004. 30th Annual Conference of IEEE*, volume 3, pages 2016–2021. IEEE, 2004.

Ellen Taylor-Powell and Sara Steele. Collecting evaluation data: Direct observation. *Program Development and Evaluation. Wiscounsin: University of Wisconsin-Extension*, pages 1–7, 1996.

Andreas Theissler, Andreas Theissler, and Ian Dear. Autonomously determining the parameters for SVDD with RBF kernel from a one-class training Autonomously determining the parameters for SVDD with RBF kernel from a one-class training set. (November), 2015.

Kai Ming Ting. A comparative study of cost-sensitive boosting algorithms. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer, 2000.

Jack V. Tu. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49(11): 1225–1231, 1996. ISSN 0895-4356. doi: 10.1016/S0895-4356(96)00002-9.

Stephane Tuffery. Data mining et statistique décisionnelle : l'intelligence des données. 2007.

Marco Vannucci and Valentina Colla. Meaningful disretization of continuous features for association rules mining by means of a SOM. *Proceedings of the ESANN2004 European Symposium on Artificial Neural Networks*, (April):489–494, 2004.

Chi Kai Wang, Yung Ting, and Yi Hung Liu. An approach for raising the accuracy of one-class classifiers. *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, (December):872–877, 2010. doi: 10.1109/ICARCV.2010.5707217.

Shuo Wang and Xin Yao. Diversity analysis on imbalanced data sets by using ensemble models. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 324–331. IEEE, 2009.

Yu Wang, Igor V. Tetko, Mark A. Hall, Eibe Frank, A Facius, K F X Mayer, and H W Mewes. Gene selection from microarray data for cancer classification - a machine learning approach. *Computational Biology and Chemistry*, 29(1):37–46, 2005. doi: 10.1016/j.compbiolchem.2004.11.001.

QIAN Wang-Wei. Research of the ID3decision tree classification algorithm based on MapReduce. *Computer and Modernization*, 2:9, 2012.

Cheng G. Weng and Josiah Poon. A new evaluation measure for imbalanced datasets. In *Conferences in Research and Practice in Information Technology Series*, volume 87 of *AusDM '08*, pages 27–32, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc. ISBN 9781920682682. doi: xdAQaqqqAAAAAAAAAAAAE4E44.

Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107, 2014.

Jian Wu Xu, Shipeng Yu, Jinbo Bi, Lucian Vlád Lita, Radu Stefan Niculescu, and R. Bharat Rao. Automatic medical coding of patient records via weighted ridge regression. In *Proceedings - 6th International Conference on Machine Learning and Applications, ICMLA 2007*, pages 260–265. IEEE, 2007. ISBN 0769530699. doi: 10.1109/ICMLA.2007.22.

Yan Yan, Glenn Fung, Jennifer G Dy, and Romer Rosales. Medical Coding Classification by Leveraging Inter-Code Relationships. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 193–201. ACM, 2010. ISBN 9781450300551. doi: 10.1145/1835804.1835831.

Feng Yang and K Z Mao. Robust feature selection for microarray data based on multicriterion fusion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(4):1080–1092, 2011.

Yee Hwa Yang, Yuanyuan Xiao, and Mark R Segal. Identifying differentially expressed genes from microarray experiments via statistic synthesis. *Bioinformatics*, 21(7):1084–1093, 2004.

Ka Yee Yeung, Roger E Bumgarner, and Adrian E Raftery. Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics*, 21(10):2394–2402, 2005.

Liuzhi Yin, Yong Ge, Keli Xiao, Xuehua Wang, and Xiaojun Quan. Feature selection for high-dimensional imbalanced data. *Neurocomputing*, 105:3–11, apr 2013. ISSN 09252312. doi: 10.1016/j.neucom.2012.04.039.

Wei Yin, Yogesh Simmhan, and Viktor K. Prasanna. Scalable regression tree learning on Hadoop using OpenPlanet. In *Proceedings of third international workshop on MapReduce and its Applications Date - MapReduce '12*, page 57, New York, New York, USA, 2012. ACM Press. ISBN 9781450313438. doi: 10.1145/2287016.2287027.

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 435–442. IEEE, 2003.

Jianping Zhang and Inderjeet Mani. kNN Approach to Unbalanced Data Distributions: A Case Study involving Information Extraction. *Workshop on Learning from Imbalanced Datasets II ICML Washington DC 2003*, pages 42–48, 2003.

Jingjing Zhang, Kuaini Wang, Wenxin Zhu, and Ping Zhong. Least Squares Fuzzy One-class Support Vector Machine for Imbalanced Data. 8(8):299–308, 2015.

Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.

Ling Zhuang and Honghua Dai. Parameter optimization of kernel-based one-class classifier on imbalance learning. *Journal of Computers (Finland)*, 1(7):32–40, 2006. ISSN 1796203X. doi: 10.4304/jcp.1.7.32-40.