FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Machine Learning for drugs prescription

Pedro Oliveira da Silva

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Carlos Milheiro Soares

July 29, 2018

Machine Learning for drugs prescription

Pedro Oliveira da Silva

Mestrado Integrado em Engenharia Informática e Computação

July 29, 2018

Resumo

Em cada consulta médica, o histórico do paciente, incluindo exames anteriores, é analisado de forma a definir um diagnóstico. Esse processo é propenso a erros, pois podem existir diversos diagnósticos possíveis. Esta análise é muito dependente da experiência do médico. Mesmo com o diagnóstico correto, prescrever medicamentos pode ser um problema, porque existem várias possibilidades para cada doença e alguns não podem ser usados devido a alergias ou custo elevado. Seria, portanto, útil se os médicos tivessem a possibilidade de usar um sistema onde para cada diagnóstico, fosse fornecido um conjunto de medicamentos adequados.

A nossa abordagem passa por apoiar o médico nesse processo. Em vez de tentar prever o medicamento adequado, pretendemos, dadas as informações disponíveis, prever o conjunto de medicamentos adequados.

O problema de prescrição de medicamentos pode ser tratado como um problema de Multi-Label classification, dado que, para cada diagnóstico, podem ser prescritos vários medicamentos ao mesmo tempo. Devido à sua complexidade, várias simplificações foram feitas para o tornar tratável. Para tal, várias abordagens foram feitas com diferentes pressupostos. Os dados fornecidos também eram complexos, com problemas importantes de qualidade, que implicam um investimento importante em preparação dos dados, em particular, engenharia de variávies (*feature engineering*).

No geral, os resultados em cada cenário são bons com desempenhos quase o dobro da *baseline*, especialmente usando *Binary Relevance* como abordagem de transformação.

Abstract

In a medical appointment, patient information, including past exams, is analyzed in order to define a diagnosis. This process is prone to errors, since there may be many possible diagnoses. This analysis is very dependent on the experience of the doctor. Even with the correct diagnosis, prescribing medicines can be a problem, because there are multiple drugs for each disease and some may not be used due to allergies or high cost. It would thus be helpful, if the doctors were able to use a system that, for each diagnosis, provided a list of the most suitable medicines.

Our approach is to support the physician in this process. Rather than trying to predict the medicine, we aim to, given the available information, predict the set of the most likely drugs.

The prescription problem may be solved as a Multi-Label classification problem since, for each diagnosis, multiple drugs may be prescribed at the same time. Due to its complexity, some simplifications were performed in order to be treatable. So, multiple approaches were done with different assumptions. The data supplied was also complex, with important problems in its quality, that led to a strong investment in data preparation, in particular, feature engineering.

Overall, the results in each scenario are good with performances almost twice the baseline, especially using Binary Relevance as transformation approach.

Keywords: Machine Learning, Classification problems, Multi-Label Classification, Health systems, Medical Diagnoses

Acknowledgements

First of all, I would like to thank Pedro Rocha and Francisco Correira for this project, for all the support and help during this year. And to everyone in Glintt, especially the OCM team, that received me with open arms and helped me during the last six months.

To professor Carlos Soares for accepting to be my supervisor, for all the knowledge, the availability and guideness that he provided.

To Adriano Rivolli that developed the package that I used and for the time he spent with me teaching me everything I should know about the topic.

To my family, especially to my parents and brother, that always believed in me and supported me to do this course, despite the days that I had to be away from home and with bad mood.

To all my friends, the ones that are with me since kids and always helped me, especially with my doubts in the health area, and that are proud of what I achieved now; to those that I met 5 years ago and are always with me and available to hear and help me in the bad and good moments and, last but not least, to those that missed me during these years because I had to study or work for the university but always said that I should continue and one day it would pay off, especially to Eduardo Ferreira.

Thank you so much,

Pedro Oliveira da Silva

“When something is important enough, you do it even if the odds are not in your favor.”

Elon Musk

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Approach . . . 2

1.3 Document structure . . . 2

2 Literature Review **3**

2.1 Multi-Label Classification . . . 3

2.2 Multi-Label Classification methods . . . 4

2.2.1 Problem transformation methods . . . 4

2.2.2 Algorithm adaptation methods . . . 6

2.3 Evaluation metrics . . . 6

2.3.1 Bipartion metrics . . . 6

2.3.2 Baseline . . . 7

2.3.3 Statistical test . . . 8

2.4 Related work . . . 8

2.4.1 Disease . . . 10

2.4.2 Variables and size . . . 10

2.4.3 Feature selection . . . 10

2.4.4 Algorithm . . . 10

2.4.5 Model validation and data split techniques . . . 10

2.4.6 Evaluation metrics . . . 11

2.4.7 Results . . . 11

3 Drug prescription using data mining **13**

3.1 Data . . . 13

3.1.1 Diagnosis . . . 13

3.1.2 Prescription . . . 14

3.1.3 Social-Demographic . . . 15

3.2 Approach . . . 15

3.3 Data preparation . . . 17

3.4 Feature Engineering . . . 17

3.5 Evaluation . . . 18

3.6 Algorithms . . . 18

3.6.1 Java implementation of C4.5 (J48) . . . 19

3.6.2 Classification And Regression Trees (CART) . . . 19

3.6.3 Sequential Minimal Optimization (SMO) . . . 19

3.6.4 Extreme Gradient Boosting (XGB) . . . 19

3.6.5 Multi-Label K-Nearest Neighbors (ML-KNN) . . . 19

3.7 Summary 19

4 Experiments and Results 23

4.1 Naive approach 23

4.1.1 Dataset with 20 medicines 23

4.1.2 Dataset with 40 medicines 24

4.1.3 Computational cost analysis 25

4.2 First prescriptions group 26

4.2.1 Dataset with 20 medicines 27

4.2.2 Dataset with 40 medicines 28

4.3 Episodes with 1 diagnosis 28

4.3.1 Dataset with 20 medicines 28

4.3.2 Dataset with 40 medicines 29

4.4 Medicines prescribed within 6 days 30

4.4.1 Dataset with 20 medicines 30

4.4.2 Dataset with 40 medicines 31

4.5 Conclusions 31

5 Conclusions and Future Work 33

5.1 Conclusions 33

5.2 Future Work 34

References 35

A Evaluation metrics 39

A.1 Bipartition metrics 39

A.1.1 Example-based metrics 39

A.1.2 Label-based metrics 39

A.2 Ranking metrics 40

B Results 43

B.1 Results from first prescription 43

B.2 Results from episodes with 1 diagnosis 44

B.3 Results from the 6 days prescriptions 45

List of Figures

3.1 Medical appointment scenario. 16
3.2 Hospitalization scenario. 16

4.1 F1 results using 20 medicines in the naive scenario. 23
4.2 Precision results using 20 medicines in the naive scenario. 23
4.3 Recall results using 20 medicines in the naive scenario. 24
4.4 Statistical results with F1 using 20 medicines in the naive scenario. 24
4.5 F1 results using 40 medicines in the naive scenario. 25
4.6 Precision results using 40 medicines in the naive scenario. 25
4.7 Recall results using 40 medicines in the naive scenario. 25
4.8 Statistical results with F1 using 20 medicines in the naive scenario. 25
4.9 Time consumed by SMO over the number of medicines. 26
4.10 SMO results over the number of medicines. 26
4.11 Time consumed by ML-KNN over the number of medicines. 26
4.12 ML-KNN results over the number of medicines. 26
4.13 F1 results using 20 medicines in the first prescription scenario. 27
4.14 Precision results using 20 medicines in the first prescription scenario. 27
4.15 Recall results using 20 medicines in the first prescription scenario. 27
4.16 Statistical results with F1 using 20 medicines in the first prescription scenario. . . 27
4.17 F1 results using 20 medicines in the single diagnosis scenario. 28
4.18 Precision results using 20 medicines in the single diagnosis scenario. 28
4.19 Recall results using 20 medicines in the single diagnosis scenario. 29
4.20 Statistical results with F1 using 20 medicines in the single diagnosis scenario. . . 29
4.21 F1 results using 40 medicines in the single diagnosis scenario. 30
4.22 Statistical results with F1 using 40 medicines in the single diagnosis scenario. . . 30
4.23 Statistical results with F1 using 20 medicines in the 6 days scenario. 31
4.24 Statistical results with F1 using 40 medicines in the 6 days scenario. 31

A.1 Confusion matrix. 40
A.2 Diagram of the MLC evaluation metrics. 41

B.1 F1 results using 40 medicines in the first prescription scenario. 43
B.2 Precision results using 40 medicines in the first prescription scenario. 43
B.3 Recall results using 40 medicines in the first prescription scenario. 43
B.4 Statistical results using 40 medicines in the first prescription scenario. 44
B.5 Recall results using 40 medicines in the single diagnosis scenario. 44
B.6 Precision results using 40 medicines in the single diagnosis scenario. 44
B.7 F1 results using 20 medicines in the 6 days scenario. 45
B.8 Precision results using 20 medicines in the 6 days scenario. 45

B.9 Recall results using 20 medicines in the 6 days scenario. 45
B.10 F1 results using 40 medicines in the 6 days scenario. 45
B.11 Precision results using 40 medicines in the 6 days scenario. 45
B.12 Recall results using 40 medicines in the 6 days scenario. 46

List of Tables

2.1 Research matrix. 9

3.1 Diagnosis information. 14
3.2 Number of diagnoses per year. 14
3.3 Prescription information. 14
3.4 Number of prescriptions per year. 14
3.5 Social-Demographic information. 15
3.7 J48 default values. 19
3.8 CART default values. 19
3.9 SMO default values. 19
3.10 XGB default values. 19
3.6 List of attributes used. 21

Abbreviations

AUC	Area under the curve
BPMLL	Back Propagation Multi-Label Learning
BR	Binary Relevance
BRPlus	Binary Relevance Plus
CART	Classification and regression trees
CC	Classifier Chains
DM	Data Mining
ECC	Ensemble of Classifier Chains
ICC	International Classification of Diseases
ID3	Iterative Dichotomiser 3
J48	Java implementation of the C4.5
ML	Machine Learning
MLC	Multi-label classification
ML-KNN	Multi-Label K-Nearest Neighbour
RF	Random Forest
SMO	Sequential Minimal Optimization
SVM	Support Vector Machines
XGB	Extreme Gradient Boosting

Chapter 1

Introduction

Each time a patient goes to a hospital, the doctor has to analyze the symptoms and define a diagnosis, and, if necessary, provide a set of medicines that will help. The definition process can be a difficult task for the doctor because every patient is unique and numerous diseases have similar symptoms. This can lead to errors and inaccurate diagnoses and patients taking the wrong medication. The process of diagnosis identification is iterative. The doctors may need multiple appointments with the patients to make a decision. In each one, they analyze the patients symptoms, their historical data and previously conducted exams and creates a list of the most suitable diseases [NBDP09] and for each one, the best drugs. In order to reduce the number of hypotheses, more exams may be required for the next appointment.

After this, a new problem rises up. Considering the final diagnosis, the doctor has to decide the best medicines for it. Like with diagnosis definition, providing the correct medicines takes into consideration patients medical history. This is an important factor because some people can not take several remedies due to allergies to them, such as Penicillin [RDM$^+$76]. The specialist has to identify multiple drugs for the diagnosis, based on their medical experience, and, from then, decide which should be taken by the patient.

1.1 Motivation

Recently, there is a huge effort in developing systems for diseases diagnose prediction in different medical areas, such as cancer and heart problems [SLD17] [PTPP15]. However, none of them are about prescriptions. This new scenario can be very useful as an extension to these systems because the doctors will have support in the prescription phase.

Glintt is a software company that develops systems for hospitals and pharmacies around the world, in particular, for Portugal. Their goal is to enhance their products with advanced features such as this one, that can be addressed with Machine Learning approaches. If their system could support the physicians, each patient could have better treatment and appointments would take less time and more patients would be treated. It may help the hospitals because will reduce the number of patients that are waiting for his appointment. Apart from this, the new system would decrease the costs associated with hospitalized patients because the doctors will have support in their decisions, releasing space to other patients.

1.2 Approach

There are some systems that use single-label classification methods for diagnosis prediction [MNS17] [AHH+13]. These methods can be used to predict the class or label for each instance [TA13]. These methods might not be the right approach for diagnosis prediction because the patient may be diagnosed with more than one and with these techniques only one is identified. The same scenario happens when prescribing medicines, each diagnosis can be associated with multiple drugs to treat the patient. So, the diagnoses definition and medicines prescription can be considered as a Multi-Label Classification problem (MLC) where the labels are the various diagnosis and medicines. This technique will be explained in further detail in Section 2.1.

Due to drugs prescription complexity, some simplifications were necessary to make this problem treatable. The naive formulation ignores the date relation between diagnoses and prescriptions. Then several others were considered like use just the first group of prescriptions for the problems, focus on scenarios where only existed one diagnosis and diagnoses whose prescriptions were within 6 days.

1.3 Document structure

Chapter 2 presents the literature review, focusing on the definition of Multi-Label classification, including algorithms and evaluation metrics, and the related work on the use of classification techniques on diagnoses prediction.

Then, Chapter 3 includes a detailed explanation of the case study used to test the approach, including the data, the data preparation, the features, the algorithms that were tested and the evaluation metrics used.

In Chapter 4, we present the results that were obtained using multiple algorithms for each problem formulations, statistical tests to verify the results and a comparison of the computation cost of the best and worst algorithms.

In the end, Chapter 5 the final conclusions are taken from the work developed and some ideias for future work.

Chapter 2

Literature Review

In the medical area, there are some projects using classification approaches taking as input demographic and medical history data in order to provide the most likely diagnosis. The most common diseases are diabetes [MNS17] [FP17] and hearth problems [TPK$^+$17].

However, no information was found regarding the use of classification techniques for drugs prescription, at least publicly. Since diagnoses definition has a strong relation with medicines prescription and there are also similarities between the characteristics of the problems, the literature review was based on the research done about the diagnoses.

2.1 Multi-Label Classification

In data mining, there are several different tasks, in particular, regression and classification [TA13]. Both of them are based on the application of algorithms to a dataset in order to predict a target value. In classification, the target is a categorical value, also known as class or label. Whereas, in regression, the target is a numerical value.

One of the simplest techniques are binary classifiers that use datasets such as D = $\{(x_i, y_i); x_i \in X; y_i \in \{0,1\}; 1 \leq i \leq n\}$. Where X is the set of n instances, x_i is an element from X with features that categorize the object and y_i the target value for that instance and can be one of two different values. Binary classifiers create a model F using a subset of the dataset, train data, and for each unseen instance x_i, $\hat{y}_i = \hat{F}(x_i)$ is applied to predict its class [Aly05]. For example, in binary classification problem about money transfers, a class can be if it is *fraudulent* or *secure*.

However, binary classifiers do not deal directly with more than two classes [SRR14]. In order to deal with this kind of problems, multiclass or multi-label techniques can be used. The main difference between them is that multiclass algorithms only assign one class from a list of more than two, whereas in multi-label ones, each object can be associated with more than one. A good example of a multiclass problem is geometric shape recognition, the shape classes can be *triangle*, *square*, *circle* and *pentagon* but each object will be assigned with only one shape. A classic Multi-Label classification problem is labelling images with tags such as *beach*, *sunset* and *holidays*.

Like in binary problems, is necessary a set of instances $X = \{x_1, x_2, ..., x_n\}$, where x_i is an object to be classified and n the number of instances and a $L = \{\lambda_1, \lambda_2, ..., \lambda_l\}$, where l is the number of labels. With these variables, can be a training set $D = \{(x_i, Y_i); 1 \leq i \leq n\}$, where $x_i \in X$ and $Y_i \subseteq L$, with Y_i being the labels for the instance x_i. This set can be used to create a function $f : x_i \rightarrow 2^{Z_i}$, that for each unseen object $x_i \in X$ predicts the more suitable labels $Z_i \subseteq L$ [ZZ14]. Multi-Label classification is suitable when there are dependencies between the labels, like in text mining or music categorization [BLSB04].

2.2 Multi-Label Classification methods

There are two approaches in Multi-Label Classification: *problem transformation methods* and *algorithm adaptation methods* that will be explained in detail on the following sections.

> "the key philosophy of problem transformation methods is to fit data to algorithm, while the key philosophy of algorithm adaptation methods is to fit algorithm to data" [TKV06]

2.2.1 Problem transformation methods

The first one uses the philosophy of divide and conquer [HQC08]. These algorithms decompose Multi-Label problems in one or more single-label problems that are solved using single-label classifiers.

Binary Relevance (BR)

Binary Relevance is the simplest approach for these problems and according to [ACMM12], is a technique that divides a Multi-Label dataset, with l labels and n instances into l binary datasets.

For each instance x_i and label $y_j \in Y_i$, a dataset $D_j = \{(x_i, y_j); x_i \in X; y_j \in \{0, 1\}; 1 \leq i \leq n\}$ is created and then, a new model is learned. For every new instance, all the models are applied to predict which subset of labels is expected to be associated with corresponding instance.

The main problem is that relations between labels are lost, because each binary problem is solved independently of the other labels. However, the information about these labels may be useful in scenarios that exists a strong relation between them. For example, in image recognition if the system detects a car, it is likely that the label wheel or door should be predicted too, because they are related.

Binary Relevance Plus (BRPlus)

In order to improve the previous technique, was created Binary Relevance Plus (BRPlus) that uses the same principles from the Binary Relevance but the feature space is incremented in order to the classifier discovers label dependencies by themselves [ACMM12]. The new feature space for each instance has the same attributes then before and $l - 1$ new features that are the other labels.

The new attributes can be represented as w_i so that $w_i = L - \{y_j\}$. In the end, the dataset is $D_j = \{(x_i \cup w_i, y_j); x_i \in X; y_j \in Y_i; y_j \in \{0,1\}; w_i = l - \{y_j\}; 1 \leq j \leq l; 1 \leq i \leq n\}$.

Chains

Classifier Chains (CC) creates l binary classifiers linked with each other, one for each label like $\{f_1 \rightarrow f_3 \rightarrow ... \rightarrow f_n\}$. Each classifier f uses the instance attributes and the predictions from the previous $n-1$ classifiers. The order of the chain is the same as in Y_i. However, the labels order may have an impact on the results, so Ensemble of Classifier Chains was developed [GV15]. ECC creates multiple instances of CC with random labels order and random number of training instances and, in the end, the output are the labels whose value is higher than a threshold [RPHF11].

These approaches are used with several binary classifiers to solve MLC problems like the following oness.

Java implementation of C4.5 (J48)

C4.5 was created as an improvement of the Iterative Dichotomiser 3 (ID3). The goal of this algorithm is to build a tree and, in each node, select an attribute to split efficiently the dataset using the attribute with highest gain ratio. In the end, the attribute with highest information gain, it is used as splitting criteria of the dataset [HMEE14].

Classification And Regression Trees (CART)

CART is very similar to C4.5. In each node, instead of choosing the attribute with higher information gain as splitting criteria, selects the attribute whose Gini index is lowest. CART, to avoid overfitting, creates a complex model with cross-validation for the parameters [HMEE14].

Sequential Minimal Optimization (SMO)

Training SVM with large dataset can be difficult and time consuming because it results in large scale Quadratic Programming (QP) problems. There are several techniques to solve these problems, Sequential Minimal Optimization is another one that decomposes the complex QP problems in smaller ones that can be solved analytically. This approach has an advantage that the datasets can fit in memory [Pla98].

Extreme Gradient Boosting (XGB)

XGB is an implementation of the gradient boosting decision tree algorithm. This method uses Boosting, a technique that builds multiple simple models that are aggregated. In each iteration, is evaluated if the new model combined with the previous ones improved the results in the training set. In true scenarios, this model is kept and the process continues, if not, the algorithm stops. The objective function has two variables: the difference between prediction and target and a term that penalizes the complexity of the model, in order to avoid overfitting [MF17].

2.2.2 Algorithm adaptation methods

These methods are created based on single label algorithms that have been changed to be able to output results with multiple labels. The most common algorithms have been modified to be used in solving MLC problems, such as, Decision Trees [CK01], Support Vector Machines [EW01] and Neural Networks [GV15].

Multi-Label K-nearest neighbor (ML-KNN)

This method was created as an extension to the K-nearest neighbor algorithm. Each test instance that the algorithm receives as input is used to find the K-nearest neighbors in the training set. From the neighbors labels, it is collected statistical information and with the *maximum a posteriori* (MAP) principle, it is determined the labelset for each test instance [ZZ07].

Multi-Label C4.5 (ML-C4.5)

ML-C4.5 is an adaptation of the C4.5 algorithm that outputs a decision tree, however the modified version can output multiple labels in each node instead of just one. To have multiple labels, the entropy formula, that measures the amount of uncertainty of the dataset, was modified. It takes into account the probability (relative frequency) of the label being in the set ($p(y_j)$) and the probability of not being ($q(y_j)$). The entropy is used to calculated which attribute best splits the instances [CK01].

$$entropy(D) = -\sum_{j=1}^{l}((p(y_j)\log p(y_j)) + (q(y_j)\log q(y_j))$$

2.3 Evaluation metrics

In order to evaluate the algorithms results, several performance indicators are used. However, MLC problems are more complex than single-label and require different metrics [ZZ14] because they need to consider the multiple labels and their relations.

Metrics can be used to evaluate *bipartition approaches* or *ranking approaches*. The first metrics group can also be divided in *example-based* and *label-based* [GV15].

From all the metrics available, only 3 of them were used to evaluate the results. The others metrics are explained in more detail in Appendix A.

2.3.1 Bipartion metrics

Bipartition metrics are generally used for algorithms that, on their output, provide two label groups. One with the relevant labels and another with the non-relevant ones. This division can be accomplished based on a threshold value. Each predicted label has a probability and if that percentage is higher than a predefined value, then it is considered as a relevant label. If not, that label will be part of the non relevant set [GV15].

Let $D = \{(x_i, Y_i); 1 \leq i \leq n\}$ be a multi-label test set with n number of instances, Y_i true labels and Z_i the predicted labels for each instance x_i.

- **Precision**

  Precision measures the percentage of labels that are correctly predicted. For each label is calculated the division between the union of the true and predicted labels and the predicted labels set. Then, all the values are summed to calculated the proportion based on the total number of instances.

$$Precision = \frac{1}{n} * \sum_{i=1}^{n} \frac{|Z_i \bigcap Y_i|}{|Z_i|}$$

- **Recall**

  Recall instead of using the predicted labels in the denominator, uses the true labels to measure the proportion of labels that are being predicted correctly.

$$Recall = \frac{1}{n} * \sum_{i=1}^{n} \frac{|Z_i \bigcap Y_i|}{|Y_i|}$$

- **F1-Score**

  F1-Score is a harmonic average of the precision and recall where the contribution of precision and recall to the F1-Score are equal.

$$F1 - Score = \frac{1}{n} * \sum_{i=1}^{n} \frac{2|Z_i \bigcap Y_i|}{|Z_i| + |Y_i|}$$

2.3.2 Baseline

It is good practice to use a simple/naive approach to obtain a reference performance for the problem. Then, this performance will be compared with the results from the ML methods.

Since single-label classification and Multi-Label classification are different in terms of output, the baselines used are different too. In single label, the baseline consists of the majority class, the class with higher frequency [MSCM15].

However, MLC problems have multiple labels with relation between them that should not be ignored, so different approaches are used [MSCM15]. Some of them try to maximize/minimize one of the previous metrics, leading sometimes to worst values on the others due to the optimization on one metric.

F1 baseline predicts the most frequent labels that result in the best F1 result in the training dataset.

Hamming-loss, that predicts the labels that are associated with more than half of the instances.

Subset-accuracy, predicts the most common labelset.

Ranking-loss, predicts a ranking based on the most frequent labels.

There is also another baseline, *General B*, that predicts the k most frequent labels, where k is the average number of labels in each instance, *Label Cardinality* (LC).

$$LabelCardinality = \frac{1}{n} * \sum_{i=1}^{n} |Y_i|$$

2.3.3 Statistical test

In order to validate the results, statistical tests can be used. *Nemenyi post hoc test* is used to compare all the algorithms pairwise, with different datasets. For each j algorithm in the i dataset, is calculated the algorithm rank, r_i^j. Two algorithms does not have the same performances if the difference between their average rank is greater than the critical difference [Dem06].

$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$, where q_α is the critical value, k is the number of algorithms used and N the number of datasets.

2.4 Related work

Recently, Machine Learning (ML) has gained a lot of recognition and people are trying to use it to solve some of their problems. Researchers have developed systems, based on classification models, in order to extract the most likely medical diagnosis [Kon01]. Some of them, use images from medical exams as input for the algorithms in order to identify the disease [TNINA+08]. Although, these type of systems were not considered in the investigation because the data that would be used only contained text.

The author could not find much work on predicting prescriptions however diagnoses definition and drugs prescriptions are related because both of them share the same decision process that take into consideration the medical history to provide a result.

In order to collect and summarize information about the classification systems for diagnosis, a matrix was created, as shown in Figure 2.1. For each paper, was stored a list of dimensions that provide relevant information for the author. This matrix was build using several articles. Since ML in health systems is a recent topic, the research only considered information from the past 5 years because was considered that the previous investigations may not be relevant.

The dimensions considered are: *Disease*, *Dataset*, *Size of the dataset*, *Number of attributes*, *Features selection attributes*, *Algorithm*, *Model validation technique*, *Data split approach*, *Metrics for results evaluation* and *Results*.

Reference	Date	Diagnosis	Data	Qty. Data	Qty. Attributes	Feature Selection	Algorithm	Model Evaluation	Performance Estimation	Evaluation Metrics	Results
Mercaldo2017	Sept.2017	Diabetes	Social and Demographic Medical history	768	8	-	J48	Holdout	20% Train 80% Test	Precision Recall F-Measure Roc Area	P=0.735 R=.738 FM=.736 RA=.751
Mercaldo2017	Sept.2017	Diabetes	Social and Demographic Medical history	768	8	-	MultilayerPerceptron	Holdout	20% Train 80% Test	Precision Recall F-Measure Roc Area	P=0.750 R=.754 FM=.751 RA=.793
Mercaldo2017	Sept.2017	Diabetes	Social and Demographic Medical history	768	8	-	HoeffdingTree	Holdout	20% Train 80% Test	Precision Recall F-Measure Roc Area	P=0.757 R=.762 FM=.759 RA=.816
Mercaldo2017	Sept.2017	Diabetes	Social and Demographic Medical history	768	8	-	JRip	Holdout	20% Train 80% Test	Precision Recall F-Measure Roc Area	P=0.755 R=.760 FM=.755 RA=.739
Mercaldo2017	Sept.2017	Diabetes	Social and Demographic Medical history	768	8	-	BayesNet	Holdout	20% Train 80% Test	Precision Recall F-Measure Roc Area	P=0.741 R=.743 FM=.742 RA=.806
Mercaldo2017	Sept.2017	Diabetes	Social and Demographic Medical history	768	8	-	RandomForest	Holdout	20% Train 80% Test	Precision Recall F-Measure Roc Area	P=0.754 R=.758 FM=.755 RA=.820

Table 2.1: Research matrix.

2.4.1 Disease

There is a small group of diseases which researchers are interested in. The reason for it, may be related to more investments in this area. Since these are complex problems, if new systems could get more useful information, it may speed the diagnoses process or even help the investigators to find a cure. The list includes diseases such as breast cancer [Aka09], heart failure [TPK$^+$17] [AHH$^+$13] [TNINA$^+$08] and diabetes [MNS17] [FP17] [WGZ17].

2.4.2 Variables and size

In general, all the systems used the same type of information, social-demographic information such as gender, age, weight, and medical history information [MNS17] [ATH$^+$13]. The type of medical history used may be different depending on the disease.

Overall, most of the datasets had more than five hundred instances [MNS17] [SLD17], in some cases, there were some with more than three thousand [PTPP15] [DBA$^+$15]. There are studies with around 10 attributes [MNS17] [GPMI14] but also some with almost 100 [AHH$^+$13]. This can be related to the disease.

2.4.3 Feature selection

Few projects used feature selection algorithms to select suitable variables for each problem, such as *Best first selection* [MNS17] and *Weight by SVM* [AHH$^+$13]. This lack of use may be justified by a previous selection when defining the dataset columns.

2.4.4 Algorithm

The most common algorithms were Naive Bayes [AHH$^+$13] and Support Vector Machines (SVM) [WRS10] [PTPP15]. The first one because performs good with irrelevant attributes, and has low computational cost [FP17], which makes it a good first approach to be used as a comparison to other more complex algorithms. SVM is more time consuming but can be used with more complex data. There were researchers that also used Random Forest [MNS17] [GPMI14] and different types of Neural Networks [SLD17] [GPMI14].

From the examples using MLC techniques, the most common algorithms were ML-KNN and Back Propagation Multi-Label Learning (BPMLL) [BZHS14] [SLLW13].

2.4.5 Model validation and data split techniques

Only two methods were found: *Holdout* [MNS17] [DBA$^+$15] and *K-Fold Cross Validation* [AHH$^+$13] [WRS10]. Holdout on datasets that have a large amount of information so that important correlations were in both train and test sets. Whereas, k-fold was used in datasets with a reduced number of instances [GPMI14].

2.4.6 Evaluation metrics

The most common metrics, as referred in Section 2.3, were accuracy and recall [SLD17] [AHH$^+$13]. Others statistical measures were F1-Measure [MNS17] and Area under the curve (AUC) [PTPP15].

2.4.7 Results

Overall, every project accomplished good results and most of them with accuracy of 70%. There were also some systems with accuracy values higher than 90%. Analyzing the conclusions from the papers, shows that the results obtained, using new attributes or algorithms, were an improvement over the approaches that were being used in the past [TNINA$^+$08] [AHH$^+$13].

Chapter 3

Drug prescription using data mining

3.1 Data

Glintt provided 3 sets of data: diagnoses, prescriptions and social-demographics. The data had information about hospitalized patients from August 2017 to December 2017 and their previous medical history. All the information about the patients has been anonymized for confidentiality reasons.

An episode will have all the medical information during the time that the patient is being treated, in other words, an episode stores the progress of each patient during the treatment.

A patient can be diagnosed with multiple diseases at the same time, however only one is the most important and needs the focus.

Each row in the prescription dataset is a drug prescribed and, for each episode, the medicines are prescribed in groups.

The data supplied lacked in medical history regarding the patients. So, was necessary to create some attributes that will be explained in Section 3.4.

3.1.1 Diagnosis

In Table 3.1, there are the diagnoses attributes.

The column *T_Episodio* (T_Episode) defines the type of episode and each diagnosis has an attribute *Flg_Princ* that shows the diagnosis priority. In total, there are 11727 rows and from those 7339 are hospitalizations, the type of episodes that will be considered because it's the most critical area according to Glintt. It has other episodes like Medical appointments, Cirurgies and Emergencies. From all the diagnoses, only 3208 were considered important diagnoses and the most common diagnosis description is "Hipertensao Essencial Nao Especificada Como Maligna Ou Benigna" (Unspecified essential hypertension disease), that represents 3.8% of descriptions. The attribute *Fim_Diag* (End_Diag) has 11291 of 11727 empty values. The reason for that will be explained in Section 1.2. The diagnoses were also grouped by initial date year to understand the distribution over the years and the results are in Table 3.2. However, the dataset had not the diagnoses equally spread.

Attribute	Description
Doente (Patient)	Patient hashed ID
Episodio (Episode)	Episode hashed ID
T_Episodio (T_Episode)	Type of episode
Inic_Diag (Start_Diag)	Diagnosis initial date
Fim_Diag (End_Diag)	Diagnosis final date
Flg_Princ	Diagnosis priority
Codigo (Code)	Diagnosis ICD9 code
Descr_Diag	Diagnosis description

Table 3.1: Diagnosis information.

Year	Number of diagnoses
2008	1
2010	6934
2011	1512
2012	575
2013	861
2014	328
2015	309
2016	553
2017	523
2018	131

Table 3.2: Number of diagnoses per year.

3.1.2 Prescription

Prescription attributes are shown in Table 3.3.

Attribute	Description
Doente (Patient)	Patient hashed ID
Episodio (Episode)	Episode hashed ID
Prescricao (Prescription)	Prescription hashed ID
Data_Presc (Date_Presc)	Prescription initial date
Medicamento (Drug)	ICD code
Nome_Med (Drug_Name)	*CNPEM*
Dose (Dosage)	Drug dosage
Via_Adm (Route_Adm)	Route of administration

Table 3.3: Prescription information.

Year	Number of prescriptions
2002	2
2004	1
2005	19
2006	378
2007	3416
2008	8020
2009	10310
2010	19010
2011	46673
2012	55595
2013	27294
2014	33596
2015	64824
2016	75792
2017	362736

Table 3.4: Number of prescriptions per year.

Each group of prescription has a unique ID that is stored in the attribute *Prescricao* (Prescription).

CNPEM is a Portuguese standard to represent the drug information and is created with the International Classification of Diseases (ICD) code, dosage and package dimension.

Paracetamol was the most prescribed drug, with 51064 prescriptions from 707666 total. The following ones were Sodium Chloridium with 38237, Metoclopramide with 30175, Pantoprazole with 25734 and Insulin with 23623.

The *Dose* (Dosage), for pills, goes from 1 milligram to 1000 milligrams and 1 millilitre to 1000 millilitres for injectables. The route of administration can be oral for pills and syrup and, for injectables, is intravenous.

In Table 3.2, there is the prescriptions distribution over the years. In 2017, the number of medicines was higher than on the other years.

3.1.3 Social-Demographic

In Table 3.5, is the information about the patients.

Attribute	Description
Doente (Patient)	Patient hashed ID
Sexo (Gender)	Patient gender
Ano_Nasc (Birth_Year)	Patient birth year

Table 3.5: Social-Demographic information.

The dataset had information about 13912 patients, 45% of them were male. Regarding the birth year, it goes from 1915 to 2017, with a mean value of 1966.

3.2 Approach

This project was developed using the CRISP-DM methodology [ML11]. The author started with Business and Data understanding to get knowledge about the problem that would be treated. In this case, understanding the hospitalizations process and its data and then was necessary to process the data and to create new attributes.

It was found that is common not to exist a match between the dates of the prescriptions and diagnoses. This may seem odd, but in hospitalizations is not mandatory that it happens. For example, when a new diagnosis is defined, before prescribing the medicines, the doctor needs to analyze the previous medical history or if the current prescriptions are good enough and it takes time, sometimes days to understand the impact of the actual treatment. This scenario was a problem, because, initially, the author thought that with hospitalizations, it would happen like in the medical appointments where the episodes are shorter and for each diagnosis, it is prescribed a set of medicines at that moment.

The data is not always collected on time, which leads to important data quality issues. In particular, it is hard to associate a diagnosis and the corresponding prescriptions, which affects the quality of the target variable. Noise in the target variable is typically more difficult to deal with than in the independent variables.

Most of the diagnoses did not have an end date. This can happen because:

- the doctor did not register it when it finished or when the patient was discharged;
- the data sample had most of its patients still hospitalized;

- some patients have chronic diseases will be dealing with them for the rest of their life, like diabetes.

So the final date of the diagnosis was ignored and that is the reason for the diagnoses not having an end in the charts.

The following images show two scenarios. In the Figure 3.1, represents an episode in a medical appointment, where diagnoses and prescriptions begin at the same time. In Figure 3.2, is part of an episode from a hospitalized patient.

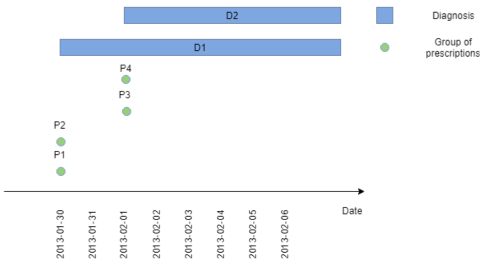


Figure 3.1: Medical appointment scenario.

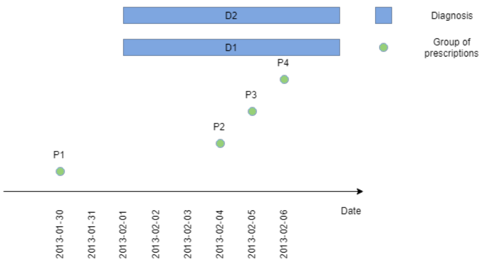


Figure 3.2: Hospitalization scenario.

In order to overcome the problem with the missing association diagnoses-prescriptions, several attempts were made.

In the first one, the time factor was ignored and tried a naive approach where all the medicines prescribed in that episode were considered as labels for all the diagnoses. In other words, each diagnosis has, as labels, the combinations of all the medicines prescribed in that episode.

Another attempt was to only focus on the first prescription of each episode. The initial prescription is the first attempt to treat the patient, from there, the following ones will be based on the results obtained with it. So this is an important scenario that had to be analyzed, in order to understand how the system would perform to predict drugs for the first prescription. For each episode, only the first group of prescriptions was used as labels.

It was also tried to use only the episodes with a single diagnosis, this way, is guaranteed that those medicines were prescribed just for those problems. The goal of this approach was an attempt to simulate the reality of the medical appointments, because each time the doctor visits his patients, can be seen as a appointment. However, there are only a few examples of these conditions and the results may not be as expected.

Finally, was idealized the formulation which was the closest to the hospitalizations reality. As referred before, sometimes exists a gap between prescriptions and diagnoses because, in hospitalizations, is not necessary that each new diagnosis leads to new prescriptions. So it was considered a period of time of 6 days to take in consideration these situations where the drugs are prescribed some days after the diagnosis. For each diagnosis, was used as labels the prescriptions from three days before the diagnosis date and until three days after. This reduced the dataset size because is a more specific scenario.

In the end, was realized that the dataset had not enough information about the historical records from the patients and the relationship between diagnoses and prescribed medicines is not established, so it was performed some feature engineering.

3.3 Data preparation

Analyzing the prescriptions, was found that the *Nome_Med* column was not atomic, as it combined two types of information into a single value, apart from the drugs name, it also had information about the dosage. However, it already existed a field for the dosage, so a new attribute was created with just the medicine name using several regex expressions.

After gathering the medicines names, the number of unique drugs was too large for the algorithms to handle [BK13]. In total, there were 420 different medicines names. So it was selected the top 20 and top 40 medicines most prescribed. The top 20 medicines represent 51% of all the prescriptions and 40 the 68%.

Each diagnosis has a unique code based on International Statistical Classification of Diseases and Related Health Problems (ICD) list [ELS+03]. This code is an numeric variable, however some diagnoses have a letter. These diagnoses are specific to the hospital and not part of ICD, due that may be ignored. ICD code has a hierarchical structure with two or more levels. For example, the diagnosis with code *518.4* belongs to the group of "Diseases Of The Respiratory System" (460-519). The diseases with code starting with 518 are from the subgroup of "Other Lungs Diseases" and the 4 specifies the disease as "Acute edema of lung, unspecified". So two attributes were created with each part of the code using the ICD R package [Was18].

The diagnosis description could be used in order to collect relevant information about it. It was considered the possibility to use a text-mining approach to extract important words that may distinguish the diagnoses. However, text-mining methods are complex and that is why was decided not to address them in this project, as there was already quite a lot of different issues to be addressed.

3.4 Feature Engineering

From the analysis performed on the dataset, it was realized that there was not enough information about medical history of each patient. In order to solve this problem, several new attributes for each patient were created using patients diagnoses and prescriptions.

It was calculated the total number of episodes and number of days since the first and last episode, for each patient.

Since each episode can have multiple diagnoses, it was computed the minimum, maximum and mean of the number of diagnoses for each patient. Considering only the important diagnoses, whose *Flg_Prin* was true, was selected, for each one, the two most prescribed medicines. The same process was used for the diagnoses that were not important.

During the years, every patient had several episodes, so, in order to take into account this information, a new attribute was created with the number of different episodes.

Considering the prescriptions, was identified the top 3 most prescribed medicines, the total number of medicines prescribed in the past, the total number of distinct medicines used in the past, the ratio between these two values and, for the number of medicines in each episode, the maximum, the minimum and mean value, for each patient.

In Table 3.6, there is a list of all the attributes used in the dataset.

3.5 Evaluation

It was used *Precision* for measuring if the drugs predicted are correct and *Recall* to measure how many drugs are being predicted. It was also considered *F1* to evaluate the combination of the two previous metrics.

General B was used in order to analyze the impact that the number of medicines used had on the results.

In order to validate the results, was used the Nemenyi post hoc test. Since there is only one dataset, it was split by year. It resulted in the 8 datasets: 2011, 2012, 2013, 2014, 2015, 2016, 2017 and 2018. Each algorithm was tested 7 times and each year was used for testing and the previous for training. For example, 2013 for testing and 2011 and 2012 for training. However, the number of diagnoses in each year is not balanced, so there are datasets with more instances than others.

Diagnoses and prescriptions time is very important. A new prescription may be necessary due to the failure of the previous one, for example. This can be seen in Figure 3.2, where the diagnoses are the same but new prescriptions were created during the time.

Based on this, the dataset was ordered by the episode initial date. And then, was split with 70% for train and 30% for test.

3.6 Algorithms

For the system, was used the *R* package, *utiML* [Riv18], because it had many MLC methods implemented.

From this package, was used the Binary Relevance (Section 2.2.1), Binary Relevance Plus (Section 2.2.1) and Ensemble of Classifier Chains (Section 2.2.1) as transformation approach, that are explained in Section 2.2. For each one, was used the following binary classifiers: Classification and regression trees (Section 2.2.1) [TA18], Java implementation of the C4.5 (Section 2.2.1) [WF05], Sequential Minimal Optimization (Section 2.2.1) [WF05] and Extreme Gradient Boosting (Section 2.2.1) [CHB^{+}18] that were available on the package. ML-KNN was used too, an adaptation of the KNN algorithm explained in Section 2.2.2.

The *mlr* package had implemented *Random Forest* for MLC. However, it only supported factors with less than 32 levels and the diagnosis description had more than that.

All the algorithms were run with the default values. It was used one adaptation method and five binary classifiers with three transformations approaches. To test all the parameters for all the algorithms would take too much time and resources, so was better to use the default values.

3.6.1 Java implementation of C4.5 (J48)

Prunning Confidence	**Min. nr. instances**
0.25	2

Table 3.7: J48 default values.

3.6.2 Classification And Regression Trees (CART)

Min. split	**Min. bucket**	**Max. compete**	**Xval**	**Max. surro.**	**Use surro.**	**Cp**	**Max. depth**
20	7	4	10	5	2	0.01	30

Table 3.8: CART default values.

3.6.3 Sequential Minimal Optimization (SMO)

Complex. const.	**Tolerance**	**Kernel**	**Exponent**
1	1.0e-3	PolyKernel	1

Table 3.9: SMO default values.

3.6.4 Extreme Gradient Boosting (XGB)

Booster	**ETA**	**Gamma**	Max. **depth**	**Min. child weight**	**Rounds**	**Objective**
gbtree	0.3	0	6	1	3	softprob

Table 3.10: XGB default values.

3.6.5 Multi-Label K-Nearest Neighbors (ML-KNN)

It was used ML-KNN implemented in *utiML* with k = 10.

3.7 Summary

There were a lot of problems with the dataset and the main one was with the missing time relation between diagnoses and prescriptions. With hospitalizations, this is a regular scenario that the author was not aware. So four scenarios were considered to overcome this problem: ignore the prescriptions date, use just the first prescription for each episode, use episodes with a diagnosis and diagnoses with prescriptions within 6 days timeframe.

After that, was found that the dataset did not have enough information about the medical history for each patient, so it was created multiple new attributes for that. Since there were over 400 unique medicines, it was only considered the top 20 and 40 most prescribed ones.

It was used 5 algorithms: XGB, SMO, CART, J48 and ML-KNN. Since the first four are binary classifiers, were used BR, BRPlus and ECC as transformation approach.

To evaluate the results, was used Precision, Recall and F1. General B was used too as a baseline.

Attribute	**Description**
EPISODIO (EPISODE)	Episode
TOP_1_MEDICAMENTO (TOP_1_MED)	Medicine most prescribed by patient
TOP_2_MEDICAMENTO (TOP_2_MED)	Second medicine most prescribed by patient
TOP_3_MEDICAMENTO (TOP_3_MED)	Third medicine most prescribed by patient
TOTAL_MEDS_POR_PAC (TOTAL_MEDS_PER_PAT)	Total number of medicines prescribed for a patient
TOTAL_MEDS_DIFF_POR_PAC (TOTAL_MEDS_DIFF_PER_PAT)	Total number of different medicines prescribed for a patient
RACIO_MEDS_DIF_TOTAIS (RATIO_MEDS_DIF_TOTALS)	Ratio between the last two attributes
MEDIA_MEDS_POR_EP (AVG_MEDS_PER_EP)	Average number of medicines in each episode
MEDIA_MEDS_POR_PAC (AVG_MEDS_PER_PAT)	Average number of medicines for each patient
NR_MEDS_PRESC	Number of medicines prescribed in an episode
TOTAL_EP_POR_PAC (TOTAL_EP_PER_PAT)	Total number of patients episodes
TOTAL_DIAGS_POR_PAC (TOTAL_DIAGS_PER_PAT)	Total number of patients diagnoses
PRIM_EP (FIRST_EP)	Date of the first patient episode
ULTM_EP (LAST_EP)	Date of the last patient episode
INIC_DIAG (START_DIAG)	Initial date of the episode
FLG_PRIN	Core diagnosis identifier
DESCR_DIAG	Diagnosis description
COD_DIAG	Diagnosis code
PRI_NIVEL (FIR_LEVEL)	First level of the diagnosis code
SEG_NIVEL (SEC_LEVEL)	Second level of the diagnosis code
MEDIA_DIAGS_POR_EP (AVG_DIAGS_PER_EP)	Average value of diagnoses in each episode
MIN_DIAGS_POR_EP (MIN_DIAGS_PER_EP)	Minimum value of diagnoses in each episode
MAX_DIAGS_POR_EP (MAX_DIAGS_PER_EP)	Maximum value of diagnoses in each episode
SEXO (GENDER)	Patient gender
ANO_NASC (BIRTH_YEAR)	Patient birth year
MELHOR_MED_PRESC_PRIN (BEST_MED_PRESC_PRIN)	Most prescribed medicine in important diagnoses
SEG_MELHOR_MED_PRESC_PRIN (SEC_BEST_MED_PRESC_PRIN)	Second most prescribed medicine in important diagnoses
MELHOR_MED_PRESC_SECUN (BEST_MED_PRESC_SECUN)	Most prescribed medicine in non important diagnoses
SEG_MELHOR_MED_PRESC_SECUN (SEC_BEST_MED_PRESC_SECUN)	Second most prescribed medicine in non important diagnoses
NR_VISITA (VISIT_NR)	Episode number

Table 3.6: List of attributes used.

Chapter 4

Experiments and Results

4.1 Naive approach

As referred before, this approach ignores if the diagnoses dates match the prescriptions. It is the simplest scenario and, thus, the first we address.

4.1.1 Dataset with 20 medicines

The final dataset has 3198 instances with the 32 attributes shown in Table 3.6.

General B is applied with $LabelCardinality = 6$ and the results from all the algorithms tested as well as the metrics considered are shown in 3 charts: *F1* (Figure 4.1), *Precision* (Figure 4.2) and *Recall* (Figure 4.3).

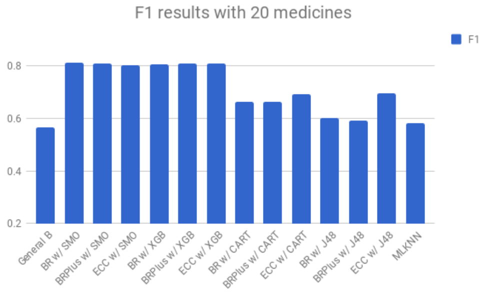


Figure 4.1: F1 results using 20 medicines in the naive scenario.

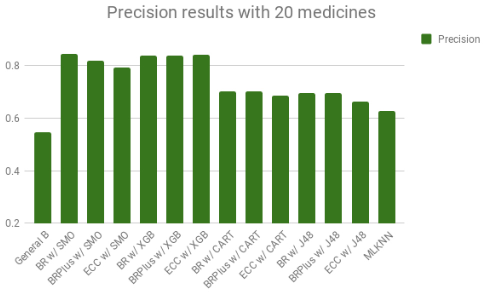


Figure 4.2: Precision results using 20 medicines in the naive scenario.

Based on Figure 4.1, every algorithm performs better than the baseline, however, ML-KNN, J48 and CART with BR and BRPlus are not so much better. SMO and XGB are the best ones with F1 values around 80%. Analyzing the *Precision* results (Figure 4.2), SMO and XGB, once again, outperform the baseline, especially SMO with BR. With XGB, all the approaches output almost the same values and ML-KNN is the worst algorithm. With the *Recall*, the same pattern occurs with SMO and XGB being the best algorithms. ML-KNN and the other binary trees classifiers have bad results, worst than the baseline. However, J48 and CART have an improvement in the

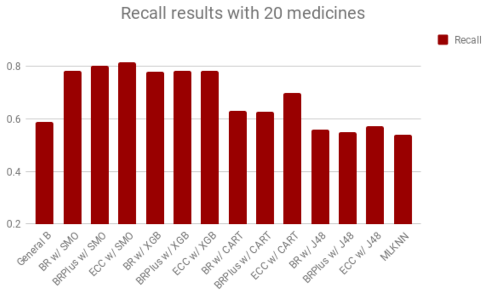


Figure 4.3: Recall results using 20 medicines in the naive scenario.

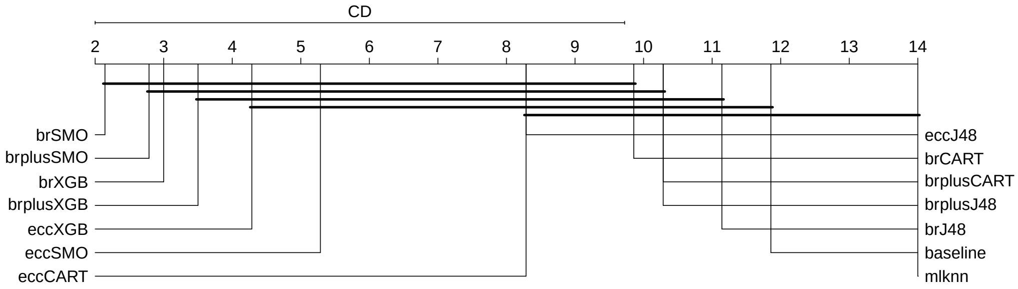


Figure 4.4: Statistical results with F1 using 20 medicines in the naive scenario.

results using the ECC. In these formulation, both SMO and XGB have the same values, so they are the best algorithms for this problem.

The dataset was decomposed in multiple small datasets using the diagnosis date. Despite that, the results in Figure 4.4 are essentially the same, with SMO and XGB having the best average ranking, especially SMO with BR (brSMO).

4.1.2 Dataset with 40 medicines

Using the top 40 most prescribed drugs, there is an increase on the dataset to 3253 diagnoses, this leads to a new *LC* value of 8.

Ensemble of Classifier Chains was the best approach for CART and J48 because the recall increased a lot compared to BR and BRPlus.

In Figure 4.8, is missing the results regarding the ECC with SMO because it takes too much memory to be calculated. Despite that, the results are the same as stated in the Section 4.1.1. SMO and XGB were the best algorithms based on the low average rank. The closest algorithm but with higher rank is CART with ECC, just like in Figure 4.5.

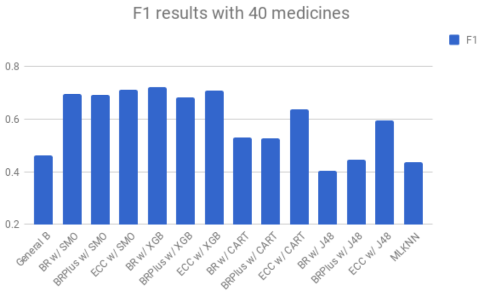


Figure 4.5: F1 results using 40 medicines in the naive scenario.

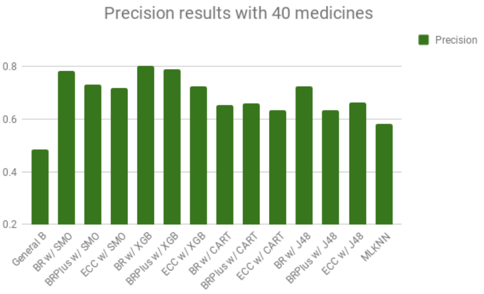


Figure 4.6: Precision results using 40 medicines in the naive scenario.

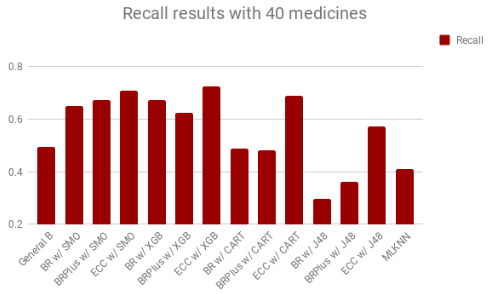


Figure 4.7: Recall results using 40 medicines in the naive scenario.

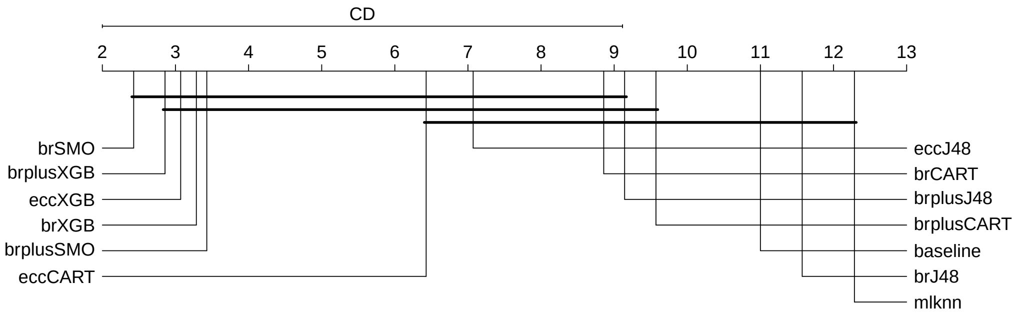


Figure 4.8: Statistical results with F1 using 20 medicines in the naive scenario.

4.1.3 Computational cost analysis

This is the most complex scenario in terms of number of instances and labels because it uses all the diagnoses and all prescriptions despite the dates not matching. So, was evaluated the performance of the best and the worst algorithm over the number of medicines. It was selected SMO as best and ML-KNN as worst one. The metric considered is F1 to evaluate the combination of precision and recall of each one.

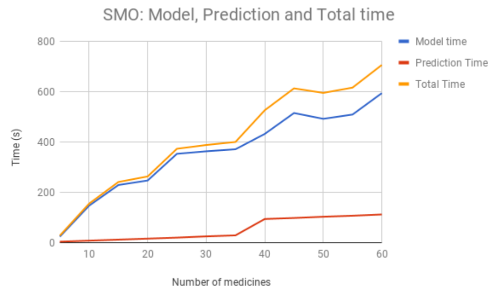


Figure 4.9: Time consumed by SMO over the number of medicines.

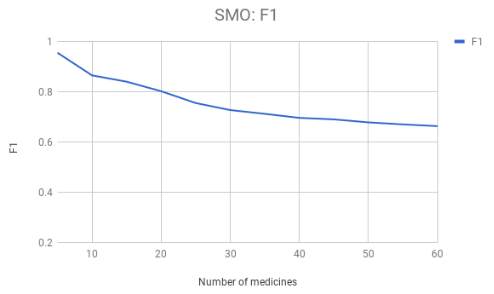


Figure 4.10: SMO results over the number of medicines.

With SMO, the time taken to create the model increases with the number of medicines, because more labels may lead to a more complex model that takes more time to be created. To predict the results, the time consumed stays almost the same until the 35 medicines. From 35 to 40 medicines, there is an increase in the time consumed and then continues with the same slope as before 35. Analyzing Figure 4.10, the results decrease with the higher number of medicines because more labels lead to more complex predictions and may exist few examples with the new labels.

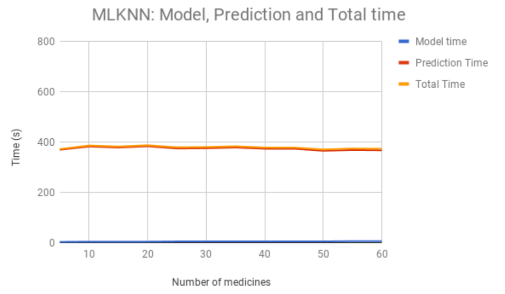


Figure 4.11: Time consumed by ML-KNN over the number of medicines.

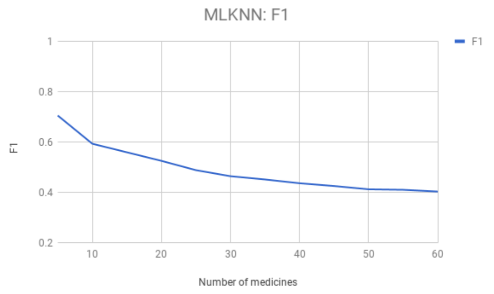


Figure 4.12: ML-KNN results over the number of medicines.

Concerning the results from Figure 4.9, the same scenario did not occur with ML-KNN as shown in Figure 4.11 because KNN is an instance-based algorithm. Instead of creating a model, it memorizes the training instances and uses it for the predictions. The process of memorizing is faster than calculating the predictions and that is why the *Prediction time* line is higher than the *Model time* one. As for the F1 results (Figure 4.12), the chart follows the same trend the F1 decreases with the increase of the medicines number, but the starting value is lower because SMO got better results than ML-KNN.

4.2 First prescriptions group

In this scenario, the idea is to focus only on the first prescription of an episode because it is the most important one as it is the first attempt to treat the patients.

4.2.1 Dataset with 20 medicines

The final dataset also has 3198 instances with the same 32 attributes but $LC = 4$.

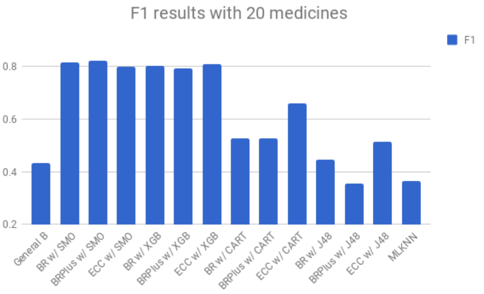


Figure 4.13: F1 results using 20 medicines in the first prescription scenario.

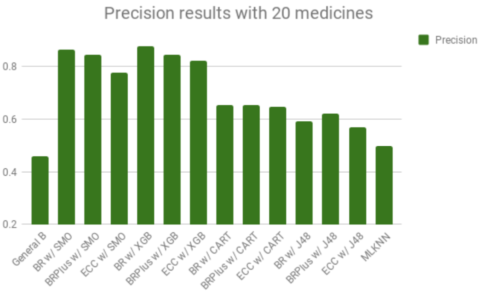


Figure 4.14: Precision results using 20 medicines in the first prescription scenario.

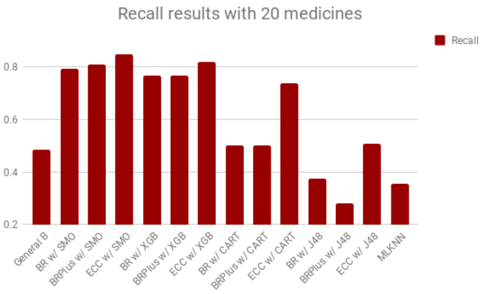


Figure 4.15: Recall results using 20 medicines in the first prescription scenario.

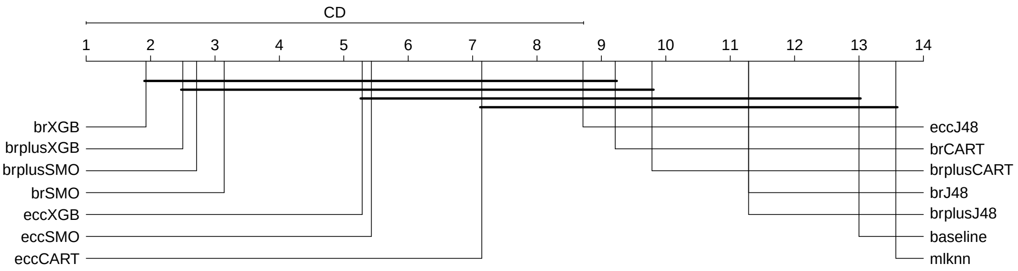


Figure 4.16: Statistical results with F1 using 20 medicines in the first prescription scenario.

Using only the first group of prescriptions, the number of labels for each diagnosis is reduced.

Overall, the results are similar to the ones in the Section 4.1.1 where SMO and XGB having the best results but J48 and CART have much better results when using ECC, like in Section 4.1.2.

4.2.2 Dataset with 40 medicines

The dataset size is the same as in Section 4.1.2, 3198, but uses more drugs so *LC* is 5. The results have a pattern just like in Section 4.2.1 but with values slightly worse.

The results using SMO and XGB are the same as in Section 4.1. The differences are on the Decision Tree algorithms that had worst performance with this formulation. For these, ECC is a good choice because it gave better results.

4.3 Episodes with 1 diagnosis

This scenario simulates how this system would perform with medical appointments instead of hospitalizations, using the episodes that have only one diagnosis.

In order to use the Nemenyi test, in this scenario, the year 2018 is not considered because there are no episodes with one diagnosis. The year 2017 only has 1 instance and some methods can not predict with just one instance so it was added to year 2016. So the datasets are from 2011 to 2016.

4.3.1 Dataset with 20 medicines

The final dataset only has 75 instances because in hospitalizations, it is unlikely to exist an episode with just one diagnosis. Despite that, this scenario could not be ignored because, in hospitalizations, each time the doctors give a check up, can be considered a medical appointment. In each visit, the doctor performs the same process as in a regular appointment.

General B was applied with a $LC = 4$.

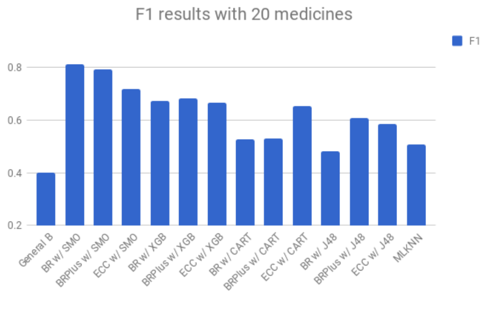


Figure 4.17: F1 results using 20 medicines in the single diagnosis scenario.

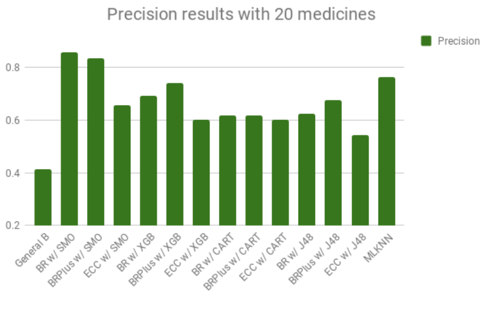


Figure 4.18: Precision results using 20 medicines in the single diagnosis scenario.

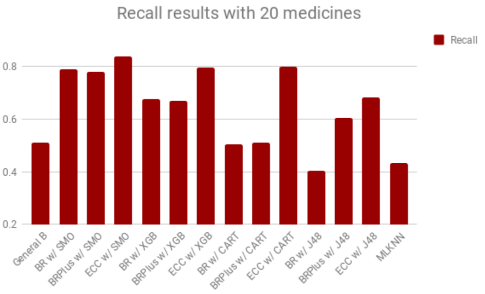


Figure 4.19: Recall results using 20 medicines in the single diagnosis scenario.

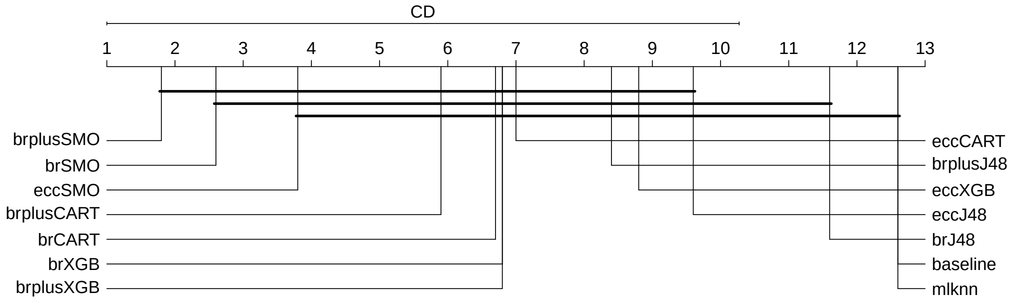


Figure 4.20: Statistical results with F1 using 20 medicines in the single diagnosis scenario.

Despite the good precision on ML-KNN, SMO continues to be the best algorithm for the problem based on Figure 4.17. XGB has slightly worse results than SMO. On the recall chart (Figure 4.19), ECC is the best approach for all the algorithms. This may be related to the small dataset that so it is easier for ECC to get the best model, since there was not many combinations as before.

The results from the statistical tests had some differences from the prediction results, because in Figure 4.20, ML-KNN and General B have the same rank and that is something that is not represented in the charts. Despite that, is clear that SMO is the best algorithm using BR or BRPlus.

4.3.2 Dataset with 40 medicines

With twice the drugs, the dataset size increases to 77 diagnoses and the *LC* is 5.

There are not big difference in the results from this section to Section 4.3.1. The number of instances is small so the results may not be reliable. However, this scenario could not be ignored due to the similarity to medical appointments.

As for the validation results in Figure 4.22, CART with ECC is the best approach and that is something that is not in Figure 4.21. The limited number of instances and all the changes with the

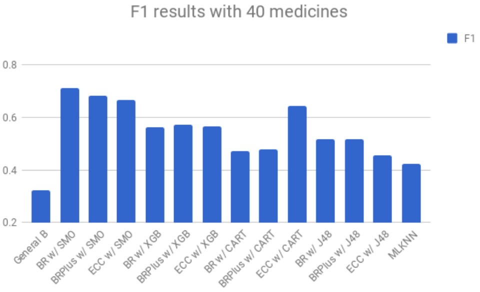


Figure 4.21: F1 results using 40 medicines in the single diagnosis scenario.

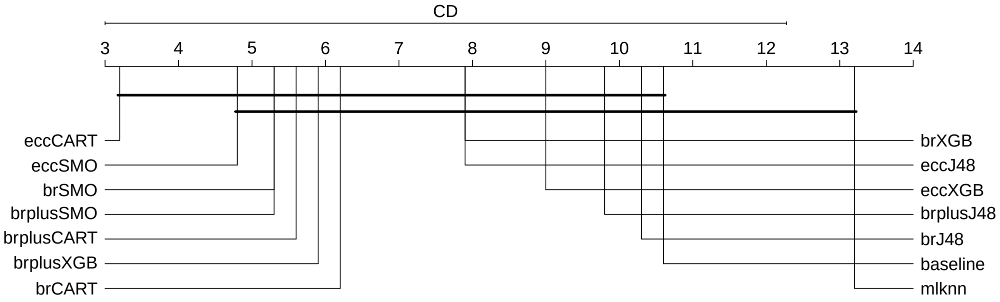


Figure 4.22: Statistical results with F1 using 40 medicines in the single diagnosis scenario.

dataset may influence the results.

Despite the simplifications that were performed and the small dataset, SMO was the best algorithm, especially with BR.

4.4 Medicines prescribed within 6 days

This last one was the best attempt to establish a time relation between diagnoses and prescriptions by using the diagnoses that have prescriptions within 6 days.

4.4.1 Dataset with 20 medicines

The final dataset has 1949 instances and LC of 6.

Despite being the closest formulation to the reallity, the results are very similar to the ones in the Section 4.1.1.

Analyzing Figure 4.23, SMO and XGB are the best algorithms, with BR and BRPlus as the best approaches due to the low average rank and ML-KNN very similar to the baseline.

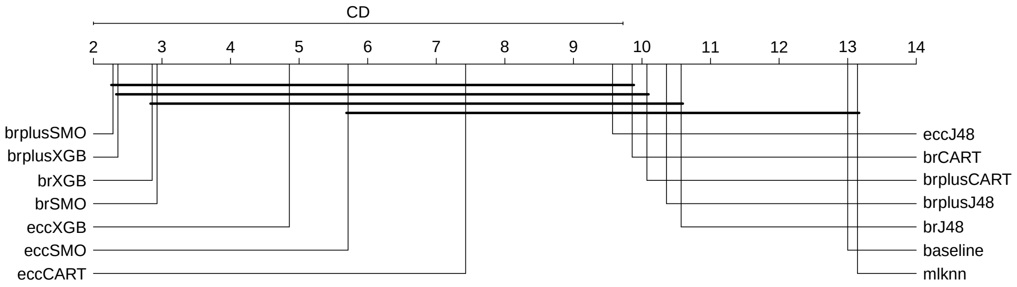


Figure 4.23: Statistical results with F1 using 20 medicines in the 6 days scenario.

4.4.2 Dataset with 40 medicines

In this scenario, the dataset size increases to 1964 with LC = 7. Just like in the previous scenarios, with 40 medicines, SMO and XGB outperform the baseline and ECC is the best approach for J48 and CART because with BR and BRPlus, they are even worse than the baseline, like in Figure 4.24.

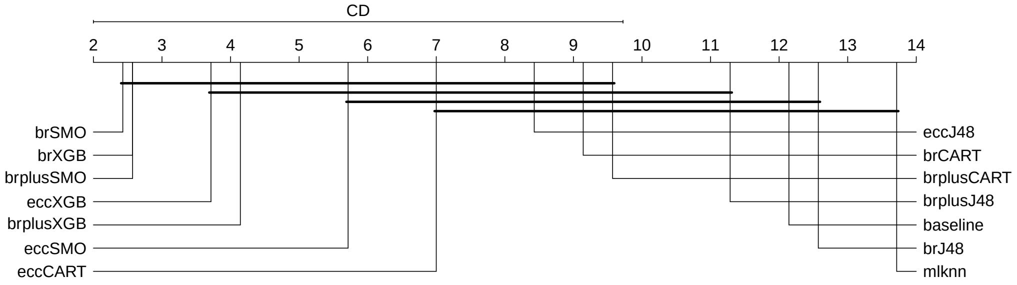


Figure 4.24: Statistical results with F1 using 40 medicines in the 6 days scenario.

4.5 Conclusions

Regarding the results, Sequential minimal optimization and Etreme Gradient Boosting are the best algorithms over all scenarios without much difference between them. With 40 drugs, the global results lowered around 10 percentage points. For the other algorithms, the overall results lower even more.

With 20 medicines, there are not important differences between J48 and CART using BR and BRPlus. Using 40, the results decreased a lot, most of the times lower than the baseline, especially with BR and BRPlus on the recall charts. In these cases, precision has high values, so despite the bad results on the recall, these algorithms predict correctly the labels when they are in the relevant group. Based on the F1 charts, for CART and J48, ECC is the best option because that is the only approach that is able to get better results both on recall and precision.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The drugs prescription is a complex process because depends on the correct diagnoses. These diagnoses were defined in multiple exams and appointments. The same diagnosis may have different prescriptions for each patient, due to health problems, for example. Since ML techniques are being used in the health systems and *Glintt* develops software for hospitals and pharmacies, was proposed the development of a system that would receive diagnoses from patients as input and provide drugs suggestions to the doctors. With this, the specialists would have some help on the prescription process.

The development followed a CRISP-DM approach, that started with the analysis of the problem and the data that would be used. The dataset was created using information from diagnoses, prescriptions and social-demographic. Since it did not have enough data about the medical history for each patient, new attributes were created. Other attributes were discarded because had a huge amount of empty values.

The number of drugs was very large so it was selected the top 20 and top 40 most prescribed drugs.

To solve the problem between diagnoses and prescriptions were create four scenarios: use all the medicines prescribed in that episode, use just the first group of drugs in each episode, focus on the episodes with only a diagnosis and for each diagnosis use the prescriptions in a period of time of 6 days.

Several algorithms were used with SMO and XGB giving the best results. SMO had the advantage that was faster than XGB, so it was considered the best method.

The main problem of this project was the data because there was not a match between the diagnoses and the prescriptions dates, due to the type of episode that was used. And also, because of the lack of information that could show which prescriptions were associated to each diagnosis. Instead of predicting the prescriptions for each diagnosis, the drugs predicted were suggestions to a group of diagnoses and not for each one.

Considering all the work developed and the problem complexity, the results obtained were good, many of them were higher than the baseline and the difference was considerable. This may be an evidence that MLC techniques can also be used in drugs prescription problems.

5.2 Future Work

When a patient is hospitalized, every time the doctor checks him up, may be considered a medical appointment. So, despite the change of context regarding the data, another approach would be to use medical appointments data where the duration of an episode is shorter and have a better relation diagnoses-prescriptions.

Another attributes could be used like the drug ICD code, instead of the drugs name.

Instead of using Multi-Label tecnhiques, it could be used Multiclass ones.

Since, was used the default values for each algorithm, could be developed a grid search system to test the parameters in order to get the best results.

The developed project will be upgraded in order to have a web interface that would allow the doctors to generate the predictions. With the interface, the suggestions will be shown in a user-friendly way for them.

References

[ACMM12] Everton Alvares-Cherman, Jean Metz, and Maria Carolina Monard. Incorporating label dependency into the binary relevance framework for multi-label classification. *Expert Systems with Applications*, 39(2):1647 – 1655, 2012.

[AHH$^+$13] Roohallah Alizadehsani, Jafar Habibi, Mohammad Javad Hosseini, Hoda Mashayekhi, Reihane Boghrati, Asma Ghandeharioun, Behdad Bahadorian, and Zahra Alizadeh Sani. A data mining approach for diagnosis of coronary artery disease. *Computer Methods and Programs in Biomedicine*, 111(1):52–61, Jul 2013.

[Aka09] 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.

[Aly05] Mohamed Aly. Survey on multiclass classification methods, 2005.

[ATH$^+$13] Peter C. Austin, Jack V. Tu, Jennifer E. Ho, Daniel Levy, and Douglas S. Lee. Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes. *Journal of Clinical Epidemiology*, 66(4):398–407, Apr 2013.

[BK13] Wei Bi and James T. Kwok. Efficient multi-label classification with many labels. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–405–III–413. JMLR.org, 2013.

[BLSB04] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, sep 2004.

[BZHS14] Stefano Bromuri, Damien Zufferey, Jean Hennebert, and Michael Schumacher. Multi-label classification of chronically ill patients with bag of words and supervised dimensionality reduction algorithms. *Journal of Biomedical Informatics*, 51:165–175, oct 2014.

[CHB$^+$18] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. *xgboost: Extreme Gradient Boosting*, 2018. R package version 0.6.4.1.

[CK01] Amanda Clare and Ross D. King. Knowledge discovery in multi-label phenotype data. In Luc De Raedt and Arno Siebes, editors, *Principles of Data Mining and Knowledge Discovery*, pages 42–53, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[DBA+15] Wuyang Dai, Theodora S. Brisimi, William G. Adams, Theofanie Mela, Venkatesh Saligrama, and Ioannis Ch. Paschalidis. Prediction of hospitalization due to heart diseases by supervised learning methods. *International Journal of Medical Informatics*, 84(3):189 – 197, 2015.

[Dem06] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.

[ELS+03] Steve Ewart, Marilyn Langfeld Layout, Reda Sadki, Ruth Bonita, Srinath Reddy, Sarah Galbraith, Douglas Bettcher, Margaret MacIntyre, Margaret Peden, Mark Rozenberg, Christie Vu, David Evans, William Savedoff, Alaka Singh, Barbara Stilwell, Wim Van Lerberghe, Eugenio Villar Montesinos, Prerna Banati, Michel Beusenberg, Sandro Colombo, Carlos Dora, Joan Dzenowagis, Helga Fogstad, Elangovan Gajraj, Gauden Galea, Claudio Garcia Moreno, Yusuf Hemed, Alan Hinman, Alex Kalache, Rania Kawar, Michele Levin, Alan Lopez, Abdelhay Mechbal, Lembit Rago, Shekhar Saxena, Philip Setel, Cyrus Shahpar, Hans Troedsson, Alice Yang, Dorjsure Bayarsaikha, Steve Begg, Christina Bernard, Dan Chisholm, Steve Ebener, Emmanuela Gakidou, Yaniss Guigoz, Patricia Hernández, Mollie Hogan, Kim Iburg, Chandika Indikadahena, Mie Inoue, Karsten Lunze, Doris Ma Fat, Takondwa Mwase, Fanny Naville, Jean-Pierre Poullier, Chalapati Rao, Darryl Rhoades, Hossein Salehi, Joshua Salomon, Angelica Sousa, Ruben M Suarez-Berenguela, U Than Sein, Niels Tomijima, Nathalie Van de Maele, Sven Volkmuth, and Hongyi Xu. WHO Library Cataloguing-in-Publication Data. 2003.

[EW01] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 681–687, Cambridge, MA, USA, 2001. MIT Press.

[FP17] Meherwar Fatima and Maruf Pasha. Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, 09(01):1–16, 2017.

[GPMI14] Gabriele Guidi, Maria Chiara Pettenati, Paolo Melillo, and Ernesto Iadanza. A machine learning system to improve heart failure patient assistance. *IEEE Journal of Biomedical and Health Informatics*, 18(6):1750–1756, Nov 2014.

[GV15] Eva Gibaja and Sebastián Ventura. A Tutorial on Multilabel Learning. *ACM Computing Surveys*, 47(3):1–38, 2015.

[HMEE14] Badr HSSINA, Abdelkarim MERBOUHA, Hanane EZZIKOURI, and Mohammed ERRITALI. A comparative study of decision tree id3 and c4.5. *International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Advances in Vehicular Ad Hoc Networking and Applications 2014*, 4(2), 2014.

[HQC08] Wei Hu, Yuzhong Qu, and Gong Cheng. Matching large ontologies: A divide-and-conquer approach. *Data and Knowledge Engineering*, 67(1):140–160, Oct 2008.

[Kon01] Igor Kononenko. Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001.

[MF17] Rory Mitchell and Eibe Frank. Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, 3:e127, Jul 2017.

[ML11] Sérgio Moro and Raul M S Laureano. Using Data Mining for Bank Direct Marketing: An application of the CRISP-DM methodology. *European Simulation and Modelling Conference*, (Figure 1):117–121, 2011.

[MNS17] Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. ScienceDirect Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning Techniques. *Procedia Computer Science*, 112(00):2519–2528, 2017.

[MSCM15] Jean Metz, Newton Spolaôr, Everton Alvares Cherman, and Maria Carolina Monard. Comparing published multi-label classifier performance measures to the ones obtained by a simple multi-label baseline classifier. *CoRR*, abs/1503.06952:189 – 198, 2015.

[NBDP09] Geoff Norman, Kevin Barraclough, Lisa Dolovich, and David Price. Iterative diagnosis. *BMJ (Clinical research ed.)*, 339(7339):b3490, 2009.

[Pla98] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Microsoft, Apr 1998.

[PTPP15] Maryam Panahiazar, Vahid Taslimitehrani, Naveen Pereira, and Jyotishman Pathak. Using EHRs and Machine Learning for Heart Failure Survival Analysis. In *Studies in Health Technology and Informatics*, volume 216, pages 40–44. NIH Public Access, 2015.

[RDM$^+$76] G P Ravelli, D.M, M.S, Zena A Stein, M.A, M.B, Mervyn Susser, and F.R.C.P. The New England Journal of Medicine Downloaded from nejm.org at UC SHARED JOURNAL COLLECTION on March 8, 2015. For personal use only. No other uses without permission. From the NEJM Archive. Copyright © 2009 Massachusetts Medical Society. All rights reser. *The New England Journal of Medicine*, 295, 1976.

[Riv18] Adriano Rivolli. utiml: Utilities for multi-label learning. https://cran.r-project.org/web/packages/utiml/index.html, 2018. Accessed: 2018-03-09.

[RPHF11] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

[SLD17] Sumedh Sontakke, Jay Lohokare, and Reshul Dani. Diagnosis of liver diseases using machine learning. *2017 International Conference on Emerging Trends Innovation in ICT (ICEI)*, pages 129–133, Feb 2017.

[SLLW13] Huan Shao, GuoZheng Li, GuoPing Liu, and YiQin Wang. Symptom selection for multi-label data of inquiry diagnosis in traditional chinese medicine. *Science China Information Sciences*, 56(5):1–13, May 2013.

[SRR14] N Satyanarayana, Ch Ramalingaswamy, and Y Ramadevi. Survey of Classification Techniques in Data Mining. *IJISET -International Journal of Innovative Science, Engineering & Technology*, 1(9), 2014.

[TA13] Divya Tomar and Sonali Agarwal. A survey on Data Mining approaches for Healthcare. *International Journal of Bio-Science and Bio-Technology*, 5(5):241–266, 2013.

[TA18] Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2018. R package version 4.1-13.

[TKV06] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. A review of multi-label classification methods. *Proceedings of the 2nd ADBIS . . .*, 2006.

[TNINA$^+$08] Tanawut Tantimongcolwat, Thanakorn Naenna, Chartchalerm Isarankura-Na-Ayudhya, Mark J. Embrechts, and Virapong Prachayasittikul. Identification of ischemic heart disease via machine learning analysis on magnetocardiograms. *Computers in Biology and Medicine*, 38(7):817–825, Jul 2008.

[TPK$^+$17] Evanthia E. Tripoliti, Theofilos G. Papadopoulos, Georgia S. Karanasiou, Katerina K. Naka, and Dimitrios I. Fotiadis. Heart Failure: Diagnosis, Severity Estimation and Prediction of Adverse Events Through Machine Learning Techniques. *Computational and Structural Biotechnology Journal*, 15:26–47, 2017.

[Was18] Jack O. Wasey. *icd: Tools for Working with ICD-9 and ICD-10 Codes, and Finding Comorbidities*, 2018. R package version 2.4.1.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[WGZ17] A Wosiak, K Glinka, and D Zakrzewska. Multi-label classification methods for improving comorbidities identification. (June):1–10, 2017.

[WRS10] Jionglin Wu, Jason Roy, and Walter F. Stewart. Prediction modeling using EHR data: Challenges, strategies, and a comparison of machine learning approaches. *Medical Care*, 48(6 SUPPL.):S106–S113, Jun 2010.

[ZZ07] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048, 2007.

[ZZ14] Min Ling Zhang and Zhi Hua Zhou. A review on multi-label learning algorithms, aug 2014.

Appendix A

Evaluation metrics

A.1 Bipartition metrics

These metrics provide two label groups. One with the relevant labels and another with the non-relevant ones based on a threshold.

A.1.1 Example-based metrics

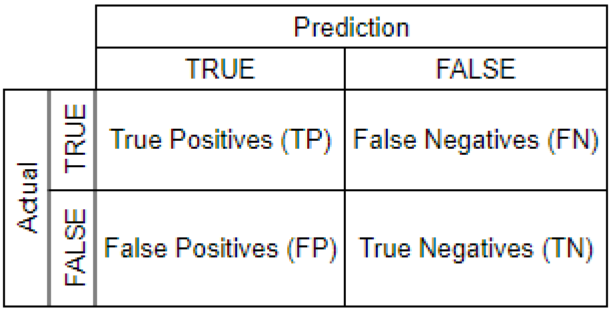
Metrics of this type are used for understanding the average difference between the actual and predicted set of labels. Let $S = \{(x_i, Y_i); 1 \leq i \leq n\}$ be a multi-label test set with n number of instances, Y_i true labels for the instance x_i, Z_i the predicted labels and $f(\cdot)$ a multi-label classifier.

- **Accuracy**
  Average proportion of correctly predicted labels from the total number of labels for each instance.
  $$Accuracy = \frac{1}{n} * \sum_{i=1}^{n} \frac{|Z_i \bigcap Y_i|}{|Z_i \bigcup Y_i|}$$

- **Hamming loss**
  Proportion of the instance-label pairs that were misclassified, where $\triangle$ stands for the difference between the two sets, predicted and actual labels, of each instance. The fraction of labels that are incorrectly predicted.
  $$Hamming\, loss = \frac{1}{n} * \sum_{i=1}^{n} \frac{1}{|L|} * |Z_i \triangle Y_i|$$

A.1.2 Label-based metrics

Label-based metrics require four values: the number of labels that were true and predicted correctly, true positives (tp), the number of labels that were false and were predicted as false, true negatives (tn), the number of labels that false but were predicted as true, false positives (fp) and the number of labels that were true but were predicted as false, false negatives (fn). With these four variables can be built a confusion matrix, Figure A.1.



		Prediction	
		TRUE	FALSE
Actual	TRUE	True Positives (TP)	False Negatives (FN)
	FALSE	False Positives (FP)	True Negatives (TN)

Figure A.1: Confusion matrix.

These values are used on different binary evaluation metrics, B, such as: accuracy, recall, precision, F1-score. There are two approaches for it:

- **Macro**

  Macro will compute the corresponding metric independently for each label and then calculate the average. This metric assumes equal weight to the labels [ZZ14].

$$B_{macro} = \frac{1}{n}\sum_{i=1}^{n} B(TP_i, FP_i, TN_i, TF_i)$$

- **Micro**

  Micro will aggregate the instances from all labels and compute the corresponding metric. Instead of assuming equal weights to the labels, this one considers equal weight to the instances [ZZ14].

$$B_{micro} = B(\sum_{i=1}^{n} TP_i, \sum_{i=1}^{n} FP_i, \sum_{i=1}^{n} TN_i, \sum_{i=1}^{n} TF_i)$$

A.2 Ranking metrics

Ranking metrics take into account the value of $rank_f(x_i, z_i)$. Label rank is like the rank in sports, represents a position for a performance and the goal is to get best one. The rank function returns the position of the label z_i in the label set Y_i, higher the rank, better is the label. The list of ranked labels can be sorted with a function $\tau_i(z_i)$ that associates the rank value to a position.

- **One Error**

  Measures the number of times in which the top-ranked label is not in the relevant label set.

$$One\text{-}Error = \frac{1}{n} * \sum_{i=1}^{n} [[arg\,min\, \tau_i(z_i) \notin Y_i]],\; z_i \in L$$

- **Coverage**

  Measures the average distance to cover all the relevant labels.

$$Coverage = \frac{1}{n} * \sum_{i=1}^{n} max\, \tau_i(z_i) - 1,\, z_i \in Y_i$$

- **Average precision**

  Measures for each label if the previous labels (lower ranking value) had been well predicted.

$$Average\, Precision = \frac{1}{n} * \sum_{i=1}^{n} \frac{1}{|Y_i|} \sum_{\lambda \in Y_i} \frac{|\{\lambda' \in Y_i | \tau_i(\lambda') \leq \tau_i(\lambda)\}|}{\tau_i(\lambda)}$$

In the Figure A.2, there is a diagram that shows hierarchically all the metrics explained before.

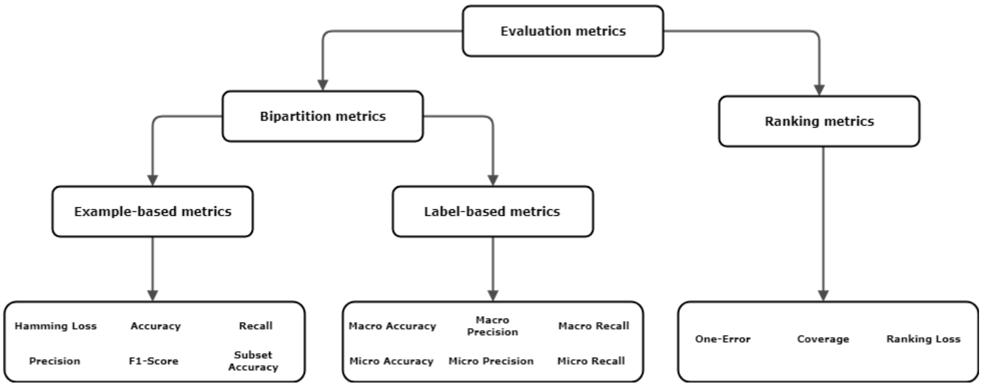


Figure A.2: Diagram of the MLC evaluation metrics.

Evaluation metrics

Appendix B

Results

B.1 Results from first prescription

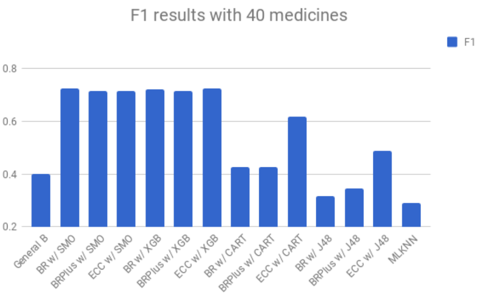


Figure B.1: F1 results using 40 medicines in the first prescription scenario.

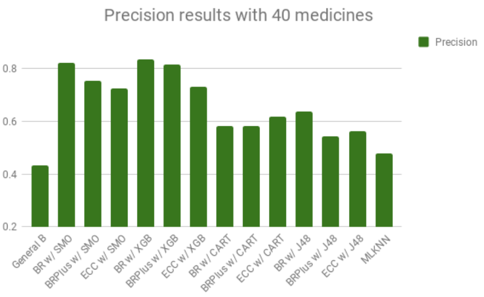


Figure B.2: Precision results using 40 medicines in the first prescription scenario.

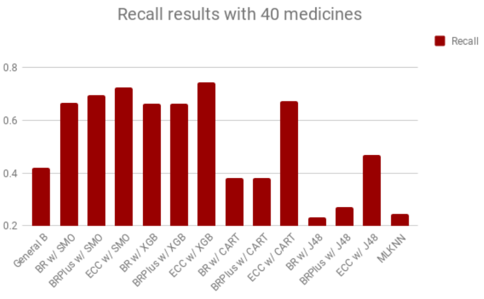


Figure B.3: Recall results using 40 medicines in the first prescription scenario.

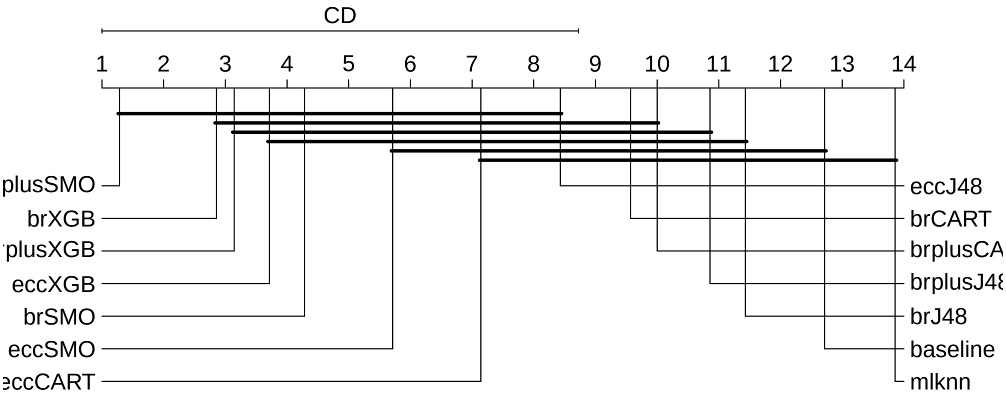


Figure B.4: Statistical results using 40 medicines in the first prescription scenario.

B.2 Results from episodes with 1 diagnosis

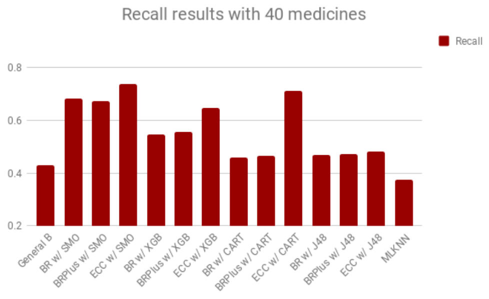


Figure B.5: Recall results using 40 medicines in the single diagnosis scenario.

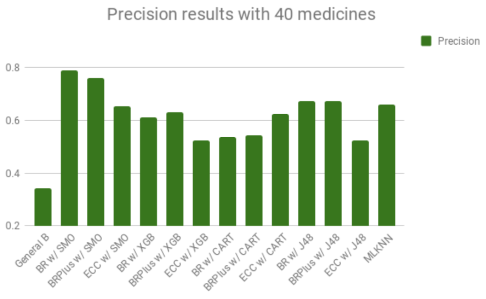


Figure B.6: Precision results using 40 medicines in the single diagnosis scenario.

B.3 Results from the 6 days prescriptions

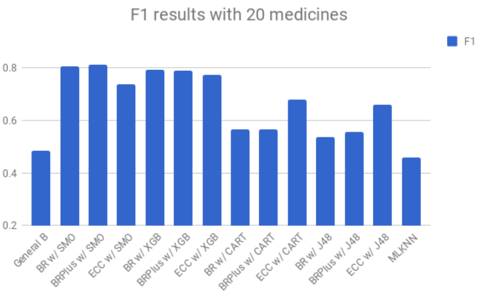


Figure B.7: F1 results using 20 medicines in the 6 days scenario.

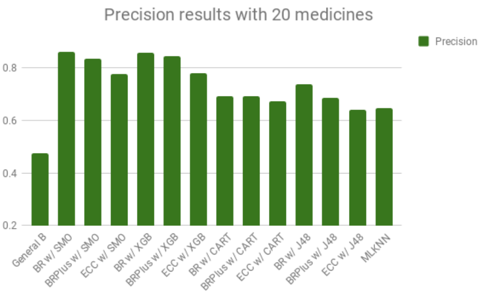


Figure B.8: Precision results using 20 medicines in the 6 days scenario.

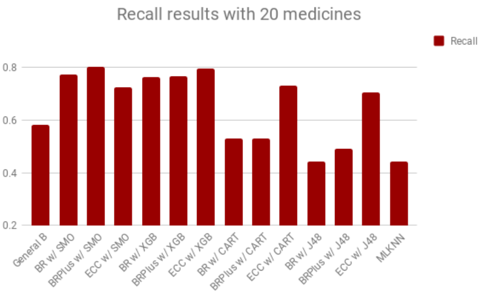


Figure B.9: Recall results using 20 medicines in the 6 days scenario.

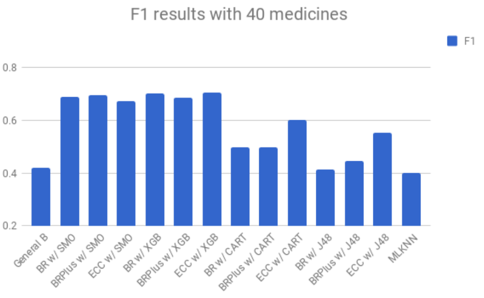


Figure B.10: F1 results using 40 medicines in the 6 days scenario.

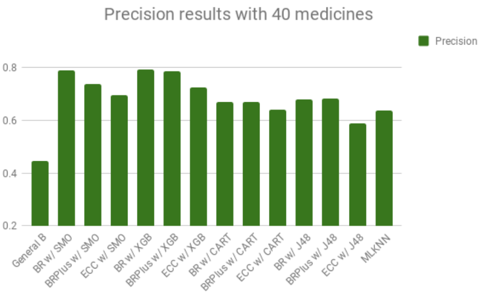


Figure B.11: Precision results using 40 medicines in the 6 days scenario.

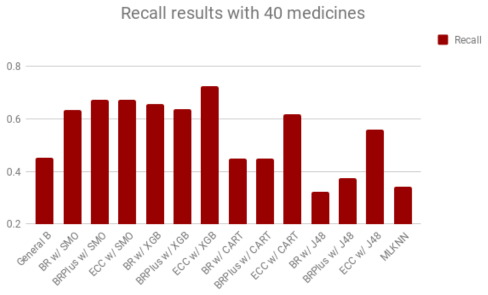


Figure B.12: Recall results using 40 medicines in the 6 days scenario.