TÉCNICO LISBOA

RedParkMiner

A tool for data analysis in car parking business

Nuno Miguel Sousa da Silva Pinto

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Cláudia Martins Antunes

Examination Committee

Chairperson: Prof. João Emílio Segurado Pavão Martins

Supervisor: Prof. Cláudia Martins Antunes

Member of the Committee: Prof. Sara Alexandra Cordeiro Madeira

May 2017

Acknowledgment

The accomplishment of this work is the end of a very important cycle in my life and I want to use this section to thank my supervisor Professor Cláudia Antunes, whom I've known since my first year here at IST, that has taught me a lot since then and has helped me during the development of this work with her guidance, perseverance and knowledge.

The completion of my degree involved many days and nights of learning but at the end of the day I would be compensated by my great friends, which I had the pleasure of knowing in this great institution, so a very special thanks to those that have accompanied me during this journey.

In a more sentimental note, I would like to thank my family for their never-ending support during my life, my parents for all the love and values that have transmitted me, my aunts for their love and for always being there for me, my uncles for always making me feel safe, my girlfriend and partner for her encouragement and kindness and finally to my grandmother that I love very much, that has raised me since I was a little boy and that I'm very proud to have in my life.

Resumo

Atualmente as empresas geram cada vez mais grandes quantidades de dados, que por sua vez estão ligados ao seu negócio operacional/financeiro. Dentro destas empresas, há uma necessidade crescente de explorar esses dados para entender melhor que tipo de ações podem tomar para melhorar o crescimento a nível de negócio, de forma que não seja exclusivamente empírica ao ser apoiado por um modelo científico válido.

Neste trabalho o foco, em termos de negócio, tem sido direcionado para o negócio de estacionamento, com especial atenção às sessões de estacionamento, que consistem numa entrada de um carro num parque, seguido de um pagamento e de uma saída do parque. Para resolver a questão da exploração de dados neste negócio, desenvolvemos uma ferramenta de análise de dados cujo objetivo é receber um conjunto de dados, manipular esse conjunto recebido ao transformá-lo em conjuntos de dados mais convenientes e interessantes e para cada um desses conjuntos de dados devolver o melhor conjunto de metodologias a serem a aplicadas em análises preditivas, de acordo com um conjunto predefinido de métricas.

No que diz respeito ao desenvolvimento da ferramenta de análise de dados, central neste trabalho, descreveremos a análise necessária realizada, bem como todo o conjunto de metodologias utilizadas pela ferramenta.

Também apresentaremos um caso de estudo, dentro do âmbito do negócio em questão onde a ferramenta foi desenvolvida e em que a ferramenta é usada, por forma a compreender melhor a sua finalidade. Neste caso de estudo, também compreenderemos quais as metodologias que melhor se aplicam no âmbito do negócio discutido.

Palavras chave: Estacionamento, ferramenta para análise de dados, análise preditiva

Abstract

Nowadays companies are increasingly generating large amounts of data, which in turn are mostly connected to their operational / financial business. Within these companies, there is an increasing necessity to explore this data in order to better understand what type of actions they can take to improve their business growth, in a way that is not exclusively empirical by being supported by a valid scientific model.

In this work the focus, in terms of business, has been directed to the car parking business, with special attention to the car parking sessions, that consist on a car entering a park followed by a payment and a park exit. To address the data exploration concern in this business, we've developed a data analysis tool whose goal is to receive a dataset, transforming the received dataset into more convenient and interesting datasets, and for each of these datasets return the best set of methodologies to perform predictive analysis, according to a predefined set of metrics.

Regarding the development of the data analysis tool, central in this work, we will describe the necessary analysis performed, as well as all the set of methodologies used in the tool.

We will also present a case study, within the business scope of which the tool has been developed and in which the tool is used, in order to better understand its purpose. In this case study, we will also understand which methodologies should apply in the current discussed business scope.

Keywords: Parking, data analysis tool, predictive analysis

Index

List of tables

List of figures

List of acronyms

NPV – Negative Predictive Value

SMOTE – Synthetic Minority Oversampling Technique

HMM – Hidden Markov Models

MISVM – Multi-Instance Support Vector Machines

MISMO – Multi-Instance Sequential Minimal Optimization

MDD – Modified Diverse Density

MIDD – Modified Inverted Diverse Density

MIEMDD – Multi-Instance Expectation Maximization with Diverse Density

QuickDDIterative – Quick Diverse Density Iterative

ETL – Extract, Transform, Load

NFR – Non-Functional Requirement

FR – Functional Requirement

1- Introduction

Data analysis has gained a dominant role in the boost and development of companies' business, which until now would save historic data mostly for business security purposes, in the eventuality of business inspections. The evolution of machine learning techniques, artificial intelligence and statistics lead to the possibility of future trend and behavior prediction, which allows businesses to make proactive decisions according to the harnessed knowledge. These techniques can help answer traditional time-consuming questions by discovering hidden patterns in the data.

This work has been developed in the car parking business scope, with the objective of understanding the behavior of its customers. It began with the exploration and manipulation of the existing data, along with the use of known methodologies to acquire information from it. After multiple iterations, we have developed a tool capable of automating and configuring, not only the data manipulation but also the data methodology exploration, with the objective of retrieving the best set of methodologies to be applied to the explored dataset.

There are many classification tools and these provide many methodologies that we can use in order to perform data analysis. Their usual behavior, and we can consider Weka [11] as an example, is limited by receiving a dataset with a predetermined structure followed by the application of a set of methodologies and a result set describing the analysis performed with the already discussed dataset.

In this work, the necessity of manipulating temporal data into multiple and different datasets lead to the proposal of a tool that helps the mitigation of the lack of flexibility in terms of data manipulation in the currently used data analysis tools. Finally, to help our analysis this tool will not only use data analysis methodologies to explore the datasets it generates, but it will also provide us with the set of methodologies that best suits these datasets according to a maximized predefined metric.

It is important to refer that we have not invented or discovered a new technology, but that we have used existing technologies to create the described tool. In terms of architecture, this is a modular tool, composed by two main modules, the first responsible for the data manipulation and the second for the data classification.

This document is organized as follows: first, we present the motivation and general definitions that lead to the development of this work in the motivation chapter. This chapter also contains a detailed description section about the domain where this project is inserted, the source data model, containing the model from which the structures defined in the data description section were based. It finishes with a presentation of a few data balancing and classification methods, along with the

necessary classification metrics that will help us rate classifications. Second, we present the requirement analysis chapter, which will provide insight on who are these tools' stakeholders, its requirements, its use cases, its structure and expected behavior. After the requirement analysis chapter, we will present the system architecture chapter where we will define all the developed processes, system components, technical stack used and the tools' limitations. After the already described chapters, we will study the tools' results in the case study chapter, by exploring each type of methodologies provided by the tool followed by a critical analysis of the overall results. This work finishes with a conclusion chapter discussing what has been learned so far.

2- Motivation

In this first chapter, we will explain in detail, not only the purpose, but also the motivations and the evolution, in terms of focus, behind the development of this work.

Professionally, I work as a Software Engineer in a software house and the work I develop is mostly in the area of Data Analysis and Business Intelligence. My work is developed in the car parking business scope, and so, one of my ongoing developments is a data warehouse that holds data regarding the car parking business. More specifically, it contains data about parking sessions, which can be simply understand as a park entry followed by a park exit. The purpose of this data warehouse is to provide business insights, so that our client can be aware of what is going on within the discussed business. Even though the existing data warehouse model could respond to many of the business questions, it lacked a data analysis component to completely achieve its purpose.

With the help of Prof. Cláudia Antunes, we have started to analyze how to exploit the existing data warehouse to generate knowledge about the business. Therefore, our first decision was to analyze the existing model, to better understand which of the model attributes would be more interesting to explore.

By exploring the rich temporal component of the data, we defined the dataset model we would be exploring, extracted the data from the data warehouse and began to perform classification on it. The results obtained were not good, especially in terms of specificity, mainly due to the lack of data balancing, and so using known balancing methods, we have tried again, only to improve little in terms of specificity. In the end of this first approach, we have decided to take advantage of the data's temporal component to generate more datasets with the application of the concept of sliding windows. In simple terms, they can be understood as a series of sets whose instances correspond to a concatenation of N contiguous events, considering that N corresponds to the window size. Even though our temporal windows were richer in terms of data, the results were not good enough.

After this first iteration, that occurred during the first half of the masters' project elaboration, we've realized that in terms of data, our scope was immensely broad, so we decided to partition the original data into sub-datasets that better represent the business in the existing nested scopes. Nevertheless, the dataset division was not the only change we have decided to make, we already knew that our set of occurrences was nothing more than a set of sequenced events and so, with that in mind, we decided to experiment methods for dealing with sequential data to achieve better classification

results. Even though we were almost certain our results would be better with these new classification approaches they were not.

In the end of this last iteration, we have come to realize that the effort made along these developments lead to the definition and creation of a data analysis tool that could be independent of a targeted domain and a great help to various data analysis general stakeholders.

Following a requirement analysis phase, where we defined the expected behavior and the structure of the tool, we specified the system in terms of technological stack and component behavior, so that the tool can be used in the case study that implicitly has led to the tool development. As a simple definition, we can state that this tool, which we will refer from this point forward as RedParkMiner, is a highly configurable tool that executes ETL / integration processes to obtain datasets that will be used to perform classifications and conclusively retrieve the best way to classify them, based on the reuse of existing software packages.

2.1- Domain

In terms of business, the major focus is the sale of car parking subscriptions to be used in the parks owned by the company. These subscriptions are associated with a specific pricing, with a specific use case and they enable their user to park his vehicle inside the company's business infrastructures, which in our case are car parks.

Nowadays it is very important for a company to analyze how its products are being sold, how its resources are being allocated but above all, using historic data, to understand how its customers behave in order to adapt, evolve and optimize its products offer.

In this particular business, it is important to understand how the parks are used by their customers not only to foresee the occupation rate of the parks but also to optimize the company parking offers. This can be achieved not only by the analysis of the past, by seeing if the use of the products is according to the purpose for which they were created, but also by predicting the end customer's behavior to suggest a better product or to change the product purpose, warranting the end customers satisfaction and the company's products profitability.

In this work, we have created RedParkMiner to help us find the best way to predict customer behavior, in simple terms we have created it to help us find the best way to

predict, according to the customers' history, if he will park again in a specific period of time. The customer behavior depends solely on parking sessions. A *parking session* can be concisely defined as an entry in a specific park followed by an exit, but for research and development purposes, we will consider that a parking session has more defining attributes, which will be discussed in the next section.

2.2- Source Data Model

To address the prediction of movements in these urban contexts, as referred previously, we focused on the car parking business, which in simple terms is understood as various customer cars entries and exits from various parks in the Iberian Peninsula.

The data warehouse from which the dataset was extracted is a star based schema (Figure 1), composed by six dimensions (of which only four are relevant for this work) and one fact table. In this schema, the fact table contains the events reporting each specific parking session, the dimension *date* is used as a referrer for the parking session start date and end date, the dimension *product* is used to describe each parking session according to the subscription type used in the parking session, the dimension *customer* has specific data about the customer responsible for the parking session and the *park* dimension describes the location where the parking session occurs.

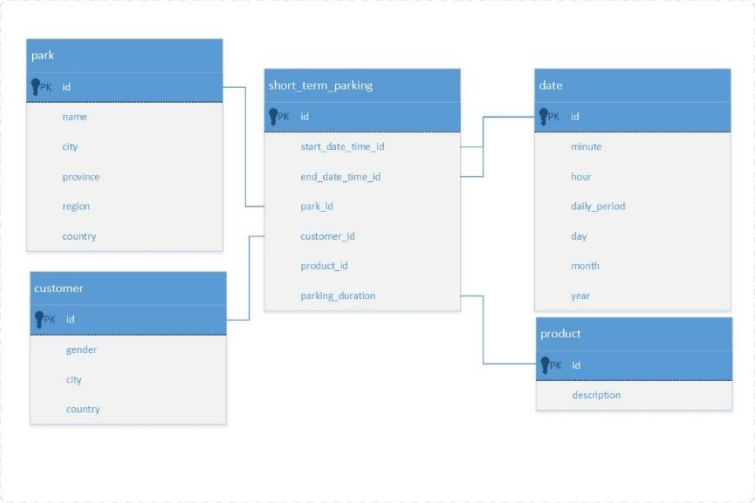

***Figure 1**- Star schema from which the dataset was extracted.*

The dataset explored is composed by 195.399 records, resulting from the denormalisation of the fact table and corresponding dimensions, which correspond to the same number of parking sessions made by park customers between 2011 and 2016

in four parks across the Spanish territory. In this dataset, parking sessions that could not be associated with specific park customers were ignored as well as parking sessions from Portugal parks due to privacy issues.

The parking session instances in the dataset, extracted from the star schema and grouped by the main entities, have the following attributes:

➔ In terms of customer information:

• The gender/context of the customer, considering it may be a single person or a company. This attribute will have the following values ‘unknown’, ’male’, ’female’ and ’company’;

• City;

• Country;

➔ In terms of product info:

• Type of product, which can have the following values: resident subscription, multi-use subscription, other subscriptions, short term subscriptions, motorcycle subscription, free transit pass, 24H subscription, nightly subscription and daily subscription;

➔ In terms of the park info:

• Name of the park;

• The name of the city where the park is located, which will be the following: Barcelona, Santiago de Compostela, Madrid, Granada;

• The Spanish province of which the park city belongs to, which will have the following values: Barcelona, La Coruña, Madrid and Granada;

• The Spanish region of which the park province belongs to, which will have the following values: Andalucía, Cataluña, Galicia and Madrid;

• Country, which will have always the same value: Spain;

Some of the datasets attributes have been synthesized from the star schema attributes. The synthesized attributes are the following:

- ➔ From the start/end date we've extracted the start/end day of the week when the parking session took place, which will have the values 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday' and 'Sunday';
- ➔ Also from the start/end date we've extracted the period of the day when the parking session started/ended;

  • With the following values: Morning as the [5AM, 12AM[interval, Noon as the [12PM, 13PM[interval, Afternoon as the [13PM, 18PM[interval, Evening as the [18PM, 20PM[interval, Night as the [20PM, 0AM[interval, Midnight as the 00AM time and Late Night as the]0AM, 5AM[interval;

- ➔ From the session duration, we've extracted the parking session duration in quarters of hour;

2.3- Data Description

With the above source dataset structure and data values detailed, we will discuss how we have manipulated the referred data into new data models to potentiate its use.

In order to improve not only the results, but also to present more concise representations of the environment and of the behaviors, we decided to divide the global dataset into 6 sub-datasets. The division was made by park and by weekday type (weekdays or weekend days).

The normalized source data, which we have stratified, was a set of parking sessions, and so we created three structure types, which enable us to use the same dataset information in 3 different approaches: an instance data structure, a Time-Window based data structure and a multi-instance data structure.

2.3.1 - Instance Data Structure

Our instance data structure is a particular case of the time-window based data structure described in the next session. It consists on a dataset composed by all individual parking sessions and a flag regarding the possible occurrence of a following parking session.

In the table 1 we can observe an example of an element that follows the instance data structure.

A1	A2	A3	A4	A5	customer_gender	customer_city	customer_id	business_unit_name	product_description	parks_next_period	id
sunday	Afternoon	5	Afternoon	sunday	C	AEREA CENTRAL	11130	Area Central	Abono Otros	FALSO	1

Table 1 *- In this table, we can observe an element of an instance data structure. For presentation purposes, customer information, business information and temporal data attribute names were shortened.*

2.3.2 – Time-Window based Data Structure

The time-window data structure is based on the temporal window concept. In simple terms, these types of datasets contain parking sessions and their respective contiguous parking session's data for a variable period, aggregated by customer. By taking advantage of the data temporality, this perspective enables the enrichment of the dataset parking session's temporal information by adding their following session's temporal data, which is the same as saying that every parking session in the dataset has been transformed into a set of contiguous parking sessions. These sets of parking sessions are variable and so we have used various sizes of temporal windows. It enables us to consider each parking session starting day and calculate a variable parking period for a specific customer, so that we can focus on predicting if that customer will park again in the following time period, considering the previous N sessions.

In the new datasets, each instance corresponds to a temporal window, which consists of the set of N sessions, that occur contiguously and the data about if in the alleged N+1 following contiguous session the customer parks again or not, which corresponds to our class from this point forth. For example, as an instance of a temporal window with N equal to 2, customer X parks on Monday so we have all data about this session but he also parks on Tuesday which means we'll have all of Tuesdays session data and he may or not park on Wednesday. The chosen values for N were from 1 to 7.

As referred in the previous section, instance data structure is a particular case of time-window based data structure, due to the fact that it is a temporal window of size 1.

In the Table 2 we can observe an example of an element that follows the time-window based data structure.

A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5	customer_gender	customer_city	customer_id	business_unit_name	product_description	parks_next_period	id
sunday	Afternoon	5	Afternoon	sunday	NA	NA	0	NA	NA	tuesday	Morning	13,6	Afternoon	tuesday	C	AEREA CENTRAL	11130	Area Central	Abono Otros	VERDADEIRO	1

Table 2 *- In this table, we can observe an element of the time-window based data structure with window value 3. For presentation purposes, customer information, business information and temporal data attribute names were shortened.*

2.3.3 - Multi-Instance Data Structure

Multi-Instance data structure, is generated from the time-window based data structure, which means that for each entry set in each time-window based dataset with window size N, N entries will be generated along with an id that represents the respective entry set along with a flag that represents the existence of a contiguous parking session in the end of the specified set.

In the Table 3 we can observe an example of an element that follows the multi-instance data structure.

A1	A2	A3	A4	A5	customer_gender	customer_city	customer_id	business_unit_name	product_description	parks_next_period	id
sunday	Afternoon	5	Afternoon	sunday	C	AEREA CENTRAL	11130	Area Central	Abono Otros	VERDADEIRO	1
NA	NA	0	NA	NA	C	AEREA CENTRAL	11130	Area Central	Abono Otros	VERDADEIRO	1
tuesday	Morning	13,6	Afternoon	tuesday	C	AEREA CENTRAL	11130	Area Central	Abono Otros	VERDADEIRO	1

Table 3 *- In this table, we can observe an element of the multi-instnce data structure with window value 3. For presentation purposes, customer information and business information were encapsulated into representative general objects.*

2.4- Data Balancing

Generally, our data is unbalanced, which in this case means that there are more clients parking in the following day than clients not parking in the following day, as we can see in Figure 2 if we compare the class counts of each of the temporal windows from each park.

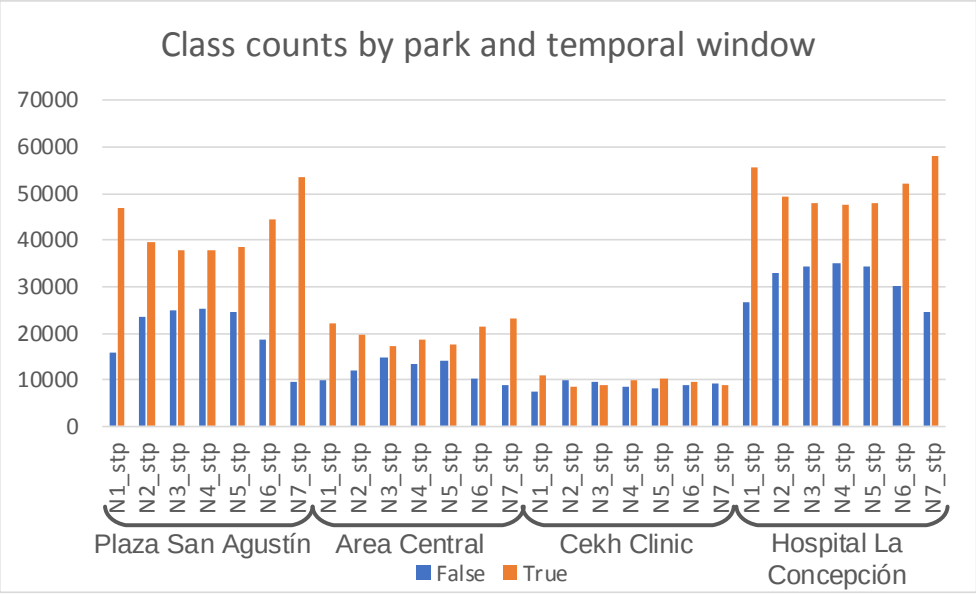

Figure 2 *- Class count by park dataset and consequent temporal window.*

When we are dealing with unbalanced datasets like these, our prediction will probably be biased toward the majority class. Therefore, in order to make our prediction unbiased, the datasets should be balanced. To cope with the unbalanced data, in terms of classification, we have agreed to treat it in two forms, treat the data as it is assuming the unbalance and treat it using balancing methods.

The balancing methods used were 'Synthetic Minority Oversampling Technique', aka SMOTE [1] and Resample. Each of these methods enable balancing a dataset using different approaches.

SMOTE generates artificial instances of the minority class using the nearest neighbors of these cases and the majority class instances are also under-sampled leading to a more balanced dataset.

In what concerns the Resample method, it produces a random subsample of the dataset using sampling with or without replacement, which implies using copies of the minority class instances to even the number of minority class or deleting instances from the majority class to achieve a balanced dataset.

2.5- Classification Methods

All classification methods used are available in the Weka java library, and all the classification methods we will refer have been integrated in RedParkMiner.

In terms of classification approaches, we have performed instance oriented classification, which comprises evaluating each temporal window set of events as one event, and multi-instance oriented classification, which consists on evaluating each temporal window set of events as sequenced events related to a specific common outcome.

Cross validation was not considered for the classification due to the various datasets large size.

2.5.1- Instance and Time-Window based Oriented

For our instance and time-window based oriented classification, we proposed to use Naïve Bayes [2], AdaBoost [3] and Random Forests [4] as classification methods.

Naïve Bayes is a popular classification method, mostly used as a baseline classifier, with a simple approach, that consists on the calculus of the associated probabilities of the values of each class attributes, using these probabilities to determine the new instance class. In a concise definition, it chooses the most probable class for an instance considering its' own attribute values.

Ada Boost, whose name is derived from the expression 'Adaptive Boost', is an adaptable classification method where multiple classifiers are employed to build a stronger classifier. It is adaptable, because the subsequent classifiers training are iteratively adjusted in favor of the instances that were misclassified by their predecessors. Equal weights are usually assigned to all training examples and then a base algorithm is chosen. On each iteration, the base algorithm is applied to the training set and the weights of the incorrectly classified instances are increased. The final model is the weighted sum of all classifiers. It is sensible in terms of data noise and outliers in the dataset.

Random Forest is a classification method that consists on ensemble learning and it is composed by decision tree predictors. These predictors are trained with different training sets and when instance classification is needed, every tree in the forest will assign a class for each instance and in the end the final classification chosen for each instance will the most assigned class.

These classifiers were chosen due to the data structure of the time-window based data and due to the fact that these are the ones who commonly present good performances in tabular classification.

2.5.2- Multi-Instance Oriented

Regarding the multi-instance oriented classification, we've used Hidden Markov Models (HMM) [5], SVMs (Support Vector Machines) [6], MDD (Modified Diverse Density) [7], MIDD (Modified Inverted Diverse Density) [7], MIEMDD (Expectation Maximization with Diverse Density) [8] and QuickDDIterative (Quick Diverse Density Iterative) [9] for classification.

Hidden Markov Models is a statistical model, whose modelled system has unobserved (hidden) states. This statistical model can be considered as a tuple of the following objects: states, observations, state transition probabilities and output probabilities. For this work, we can simple state that we fed the algorithm with the number of states (including hidden ones) and with observations to generate the statistical models for the classes present in the observations. Once the model is built, we simply retrieve the class statistical distribution for the sequences that are being used to test the model and each of these sequences will be classified with the class that has the higher distribution value.

SVMs construct a hyper plane or set of hyper planes in a n-dimensional space which will be populated with each instance that corresponds to a n-dimensional vector and the margin between the instances classes will be maximized. To classify an instance, the support vector machine verifies the location of this instance in the dimensional space of the built model and compares it to the existing hyper plane, attributing to the instance the class designated by the hyper plane. We use two variations of SVMs one that uses SMO (Sequential Minimal Optimization [10]) to solve the quadratic programming problem (MISMO) and another (MISVM) that uses Andrews (2003)[6] "maximum bag margin formulation" and heuristic optimization.

The following classifications approaches are variations of the DD algorithm (Diverse Density [7]). The DD algorithm has a similar behavior to SVM, the difference is that SVM uses hyperplanes to differentiate existing classes and DD algorithm uses a representative instance to perform that distinction, as we will detail further. MDD, based on Maron [7] DD maximization algorithm, assumes a representative instance ***T*** as the concept. This representative instance, ***T***, must be “dense” in that it is much closer to instances from positive bags than from negative bags, as well as “diverse” in that it is close to at least one instance from each positive bag. The classification of new instances is made in order of their approximation to ***T***.

MIDD has the same behavior as MDD, the only difference is that the representative instance ***T*** in terms of density is closer to the instances from the negative bags rather than the ones from the positive bags, as well as close to at least one instance from the negative bags. The classification of new instances is the same as in MDD.

MIEMDD, based on Zhang and Goldman EM-DD algorithm, like DD, in terms of classification has the same behavior, the difference resides in the calculation of the representative instance ***T***. It assumes a representative instance ***T***, just like DD does, then repeatedly performs two steps that combine Expectation Maximization with DD to search for maximum likelihood hypothesis. In the first Step (E Step), ***T*** is used to pick one instance from each bag, which is most likely, the one responsible for the label given

to the bag. In the second step (M Step), the objective is to find a new ***T'*** that maximizes DD for ***T***. Once the maximization step is complete, the proposed target is reset from ***T*** to ***T'*** and return to the first step until the algorithm converges.

In QuickDDIterative, based on JR Foulds and E Frank [9] approach, as in MIEMDD, it changes the representative instance ***T*** calculation. As referred for MDD the point of maximum diverse density is by definition close to instances from positive bags, but the best points from the different positive bags are averaged to form a hypothesis, in QuickDDIterative it has picked the best point from all positive bags as the target point.

2.6 - Classification Metrics

The chosen metrics for our analysis are sensitivity (1), specificity (2), precision (3), NPV (4) (negative predictive value) and accuracy (5).

$$Sensitivity = \frac{True\ Positive}{True\ Positives + False\ Negatives} \quad (1), \quad Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}, \quad (2)$$

$$Precision = \frac{True\ Positive}{True\ Positives + False\ Positives} \quad (3), \quad NPV = \frac{True\ Negatives}{True\ Negatives + False\ Negatives}, \quad (4)$$

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Postives + Negatives} \quad (5)$$

For all analysis that we will be performing, we have to bear in mind that when we refer:

• Accuracy, we're aiming to predict more accurately the non-occurrence and the occurrence of parking sessions in the following period compared with all the parking sessions that will or will not occur in that period.

• Sensitivity, we are aiming to predict more accurately the occurrence of parking sessions in the following period compared with all the parking sessions that will effectively occur.

• Specificity, we're aiming to predict more accurately the non-occurrence of parking sessions in the following period compared with all the parking sessions that will not effectively occur in that period.

• Precision, we're aiming to predict more accurately the occurrence of parking sessions in the following period compared with all the parking sessions that will be predicted to occur in that period.

• NPV, we're aiming to predict more accurately the non-occurrence of parking sessions in the following period compared with all the parking sessions that will be predicted to not occur in that period.

3- Requirements Analysis

This chapter is dedicated to the analysis of the needs/conditions we have taken into account in order for the development of the RedParkMiner tool. We have identified the tools Stakeholders, relevant requirements related with this tool desired behavior and the necessary use cases to understand how it should work.

Figure 3 *- Context Diagram*

In the above diagram (Figure 3), we have presented the RedParkMiner tool and the relation it possesses with its interveners.

3.1- Stakeholders

Our stakeholders can be divided into 2 categories:

➔ System users, responsible for using the tool in order to achieve their technical goals, which are Business Intelligence Developers, ETL Developers, SQL Developers, Data Integration Developers, Data Scientists.

➔ Interveners that benefit from the system from a functional, political or financial perspectives, which in this case we'll be the Project Manager, Product Owner and Team Leader.

3.2- Requirements specification

Requirement specification has been divided into two types, non-functional requirement (NFR) specification and functional requirement (FR) specification.

In the NFR specification, we will focus on how the system must operate, and so we have divided our requirements into four categories: *flexibility*, *structure* and *environment*.

In terms of *flexibility*, our requirements are as follows:

- ➔ **NFR1**: The system must allow for data importation from multiple types of data sources (databases, files and web services);
- ➔ **NFR2**: The system must allow for the use of multiple types of classification methods;
- ➔ **NFR3**: The system must be able to handle source data in order to generate convenient data sets to be used by classification methods;

In terms of *structure*, our requirements are as follows:

- ➔ **NFR4:** The system should be divided into separate modules in order to promote organization;
- ➔ **NFR5:** The system data storage should be divided into configuration storage and data handling/transformation/classification storage;

In terms of *environment*, our requirements are as follows:

- ➔ **NFR6:** The system should be independent from the operating system where it's being developed and where it's deployed;
- ➔ **NFR7:** The system installation should be simple;
- ➔ **NFR8:** The system deployment should be simple;
- ➔ **NFR9:** The system development should be clear and simple;

Regarding functional requirement specification, we must bear in mind the necessary tasks, actions or activities to be accomplished.

- ➔ **FR1:** The system must allow for multiple source type configuration.
- ➔ **FR2:** The system must allow for the imported dataset division into convenient datasets, according to the datasets attribute.
- ➔ **FR3:** The system must allow for the generation of windowed datasets/views over the imported data in order to make diverse classifications regarding the sets temporal components.
- ➔ **FR4:** The system must allow for the configuration of new classification methods, to be used on all existing datasets.
- ➔ **FR5:** The system must be able to perform classifications on all existing datasets.
- ➔ **FR6:** The system must return the best possible classification method by all predefined metrics by existing dataset.

3.3- Use Cases

Figure 4 illustrates the most relevant use cases for this system, and associate them to the enumerated requirements:

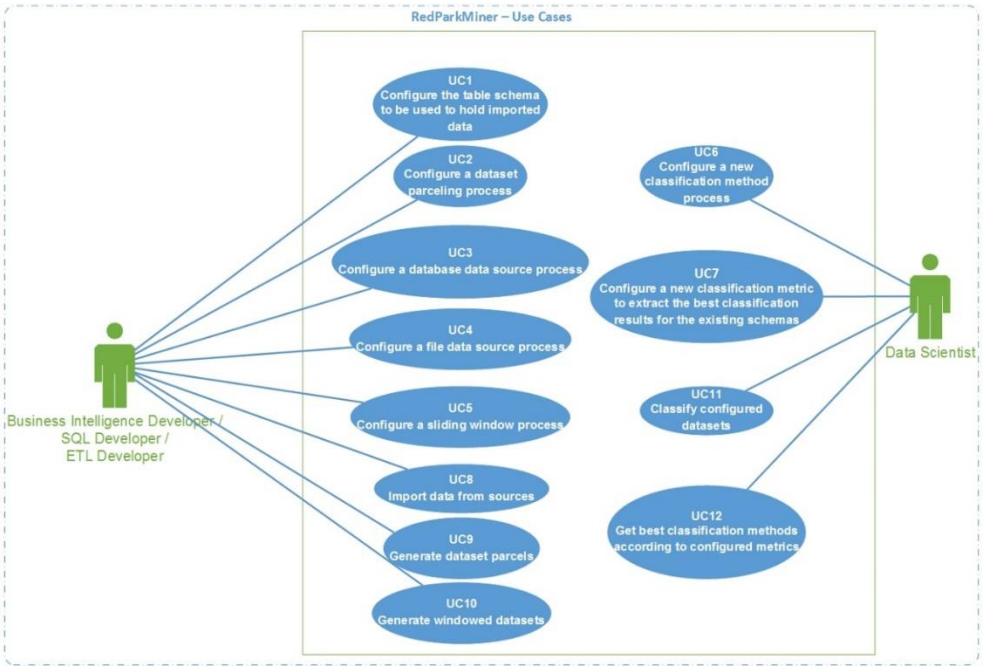

Figure 4 *- Use Case Diagram*

In the bellow tables, a detailed description of each scenario is provided in order to better understand each use case and its constraints.

Use Case	UC1 - Configure the table schema to be used to hold imported data
Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario	Precondition: A database provider instance must be installed, as well as a Database with a schema to group the imported datasets into independent tables (Whose schema name by default will be imported datasets) 1- Create a SQL script to create a table with the expected dataset schema structure. 2- Run script in database.

Table 4 *- Use Case 1 describes how to configure a table schema that will be used to import data.*

Use Case	UC2 - Configure a dataset parceling process

Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario 1	Precondition: A database provider instance must be installed, as well as a configuration database and a schema to group specific dataset configurations (Whose schema name suffix by default will be _configuration) and a database were the original dataset is stored and were the parcels will be stored as well. 1- Create a SQL script to create a table in the configuration schema, to handle the all parceling related values 2- Create a SQL script to insert into the previous created table, the values by which the parceling must be done. These values can be custom, which means that the dataset can be parceled not only explicitly but also implicitly. 3- Run scripts in the configuration database. 4- Create a new package 5- Create two new connections, using Database Connection Components, fill in the connection properties of each one with existing predefined connection values. (One of these predefined connections must be pointing to the configuration database, and the other must be pointing to the database were the original dataset is stored and were the dataset parcels will be held) 6- Create a Database Extraction Component, which uses the configuration database connection. This component shall return all the parceling values previously configured. 7- For each parceling value we will use a Database Query Component to create new schemas in the database were each parcel of the original dataset will be held. 8- Create a Database Extraction Component, which uses the previous step connection. This component shall return all dataset fields filtered by the defined parceling value. 9- Create Database Insert Component to write the previous result into a destination database table in the previously created schema. (Configure the component to drop the table if already exists and create it)
Scenario 2	Precondition: A database provider instance must be installed, as well as a database were the original dataset is stored and were the parcels will be stored as well. The previous scenario first 3 steps must be accomplished. 1- Create a new package 2- Create a new connection, using a Database Connection Component, fill in the connection properties with an existing predefined connection values. 4- Create a Database Extraction Component, which uses the previous connection. This component shall return all dataset fields filtered by the defined parceling value. 5- Create Database Insert Component to write the previous result into a destination database table in the previously created schema. (Configure the component to drop the table if already exists and create it)

Table 5 *- Use case 2 describes how to configure a process responsible for the dataset parceling.*

Use Case	UC3 - Configure a database data source process
Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario	Precondition: A database provider instance must be installed, as well as a database were the dataset will be stored. There must exist a source database from where we will extract the data. 1- Create a new package 2- Create a destination connection, using a Database Connection Component, fill in the connection properties with an existing predefined connection values. 3- Create a source connection, using a Database Connection Component, fill in the connection properties with existing source connection values. 4- Create a Database Extraction Component, fill with the query that will return the desired data. 5- Configure Database Insert Component to write the previous result into a destination database table.

Table 6 *- Use case 3 describes how to configure a process responsible for the data extraction of a database data source.*

Use Case	UC4 - Configure a file data source process
Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario	Precondition: A database provider instance must be installed, as well as a database were the dataset will be stored. 1- Create a new package 2- Create a destination connection, using a Database Connection Component, fill in the connection properties with an existing predefined connection values 3- Using a Delimited File Component, fill with necessary configurations to parse correctly the data file 4- Configure Database Insert Component to write the previous result into a destination database table.

Table 7 *- Use case 4 describes how to configure a process responsible for the data extraction of a file data source*

Use Case	UC5 - Configure a sliding window process
Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario	Precondition: A database provider instance must be installed, as well as a configuration database and a schema (Whose names suffix by default will be _configuration) and a database were the original dataset is stored and were the parcels are stored as well. 1- Create a SQL script to create table in the configuration database, to handle the windowed datasets name and their respective window size. 2- Run script 3- Create a new package 4- Create two new connections, using Database Connection Components, fill in the connection properties of each one with existing predefined connection values. (One of these predefined connections must be pointing to the configuration database, and the other must be pointing to the database were the dataset parcels are held and also were the windowed datasets will be held) 5- Create a Database Extraction Component, which uses the configuration database connection. This component shall return all parceled dataset schema names. 6- Create a Database Extraction Component, which uses the previous connection. For each previous schema name, will get each windowed dataset name and respective window size. 7- For each window size, get the parceled dataset by creating a Database Extraction Component, which uses the second configured connection. 8- For each result returned find a matching one within the window size, if there's a match return the row with a new attribute that corresponds to the next period of occurrence, which will be valued as true, otherwise return the row but with the previous attribute valued false. 9- Create a Database Insert Component to write the previous row flow into a table whose name will be the same as the window size dataset name, in the parcels schema.

***Table 8** - Use case 5 describes how to configure a process responsible the generation of temporal windows*

Use Case	UC6 - Configure a new classification method process
Actor	Data Scientist
Scenario	Precondition: A database provider instance must be installed, as well as a Database with the results schema and a table to save the classification results. 1- Create a script in the programming language more convenient and define the necessary methods to retrieve a windowed table data and be able to divide it into a training set and a test set. 2- Create a script in the previous chosen programming language more convenient and define the necessary balancing methods to balance the training set. 3- Create a script in the previous chosen programming language and define the necessary methods to apply classification methods with a previously defined training set and test set. 4- Create a script in the previous chosen programming language so that it receives as input a windowed table name, the balancing method and the classification method to apply. In the end of this script classification results must be written into a result table, along with the windowed table name, the balancing method and the classification method. 5- Create a package that iterates over all windowed tables from each dataset parcel and calls the previously defined script for the iterating dataset, with all classification methods and balancing methods.

Table 9 - Use case 6 describes how to configure a process responsible for the data classification.

Use Case	UC7 - Configure a new classification metric to extract the best classification results for the existing schemas
Actor	Data Scientist
Scenario	Precondition: A database provider instance must be installed, as well as a Database with the results schema and a table with the classification results. Classifications must already have been performed. 1- Create a new package 2- Create a new connection, using a Database Connection Component, fill in the connection properties with an existing predefined connection values. 3- Create a Database Extraction Component, which uses the previous connection. This component shall return the maximum value for a metric calculated with the classification results, grouped by the windowed table name. 4- Create a Lookup Component, which compares the maximum values of the previous component results with all the existing ones and returns the classifiers name and balancing method that match with the metrics value. 5- Create Database Insert Component to write the previous result into a destination database table related to that specific metric.

Table 10 *- Use case 7 describes how to configure a process responsible for obtaining the best classification results, according to a specific metric.*

Use Case	UC8 - Import data from sources
Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario	Precondition: All importation packages must be in the data importation module 1- Run Data Importation Package

Table 11 *- Use case 8 describes how to import data from the configured sources.*

Use Case	UC9 - Generate dataset parcels
Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario	Precondition: All parceling packages must be in the data parceling module. Data Importation must already have been run. 1- Run Data Parceling Package

Table 12 *- Use case 9 describes how to generate parcels from the configured datasets.*

Use Case	UC10 - Generate windowed datasets
Actor	Business Intelligence Developer / SQL Developer / ETL Developer
Scenario	Precondition: All windowing packages must be in the sliding window dataset module. Data Importation and Data Parceling must already have been run. 1- Run Sliding Window Dataset Package

Table 13 *- Use case 10 describes how to generate temporal windows from the datasets.*

Use Case	UC11 - Classify configured datasets
Actor	Data Scientist
Scenario	Precondition: All classification packages must be in the data classification module. Data Importation, Data Parceling and Sliding Window Data Generation must already have been run. 1- Run Data Classification Package

Table 14 *- Use case 11 describes how to classify the generated datasets.*

Use Case	UC12 - Get best classification methods according to configured metrics
Actor	Data Scientist
Scenario	Precondition: All classification packages must be in the data classification module. Data Importation, Data Parceling, Sliding Window Data Generation and Data Classification must already have been run.
	1- Run data best classification method package
	2- Access the desired metric table and retrieve best classification results for each classified dataset.

Table 15 *- Use case 12 describes how to retrieve the best classification methods for configured datasets according to the configured metrics.*

	FR1	FR2	FR3	FR4	FR5	FR6
UC1	X					
UC2		X				
UC3	X					
UC4	X					
UC5			X			
UC6				X	X	
UC7						X
UC8	X					
UC9		X				
UC10			X			
UC11					X	
UC12						X

Figure 5 *- Relation between the use cases and the functional requirements*

3.4- Structure

Bellow a block diagram (Figure 6) represents various parts of RedParkMiner components. In order to refine the specificity of each component, specific components are presented.

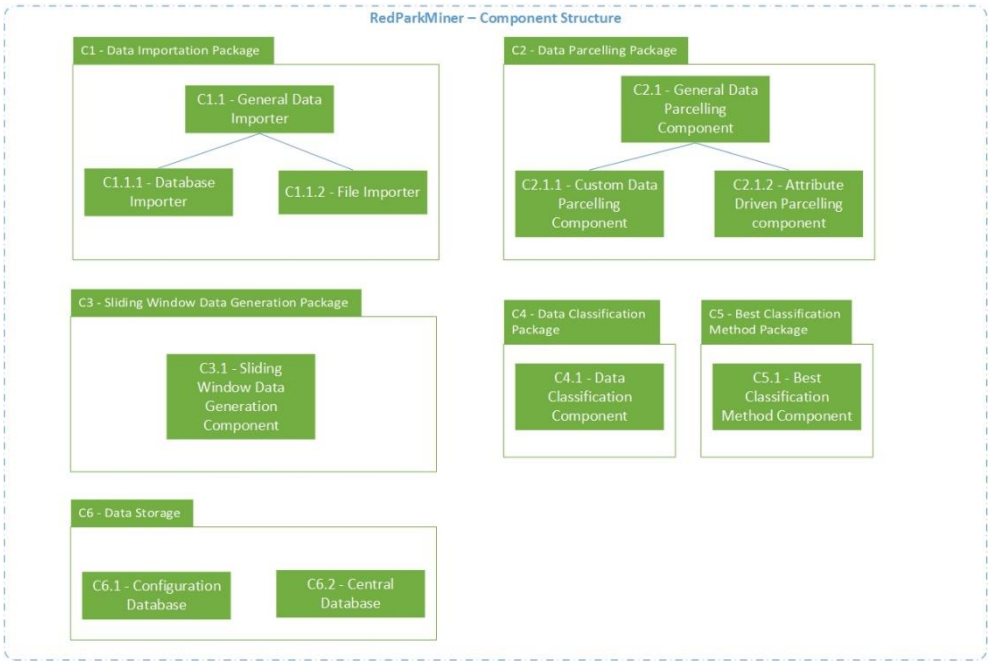

Figure 6 *- RedParkMiner block diagram*

Now that our structure is presented, we will present the relations existing between our structure, our non-functional requirements and our use cases (Figure 7 and Figure 8).

	C1	C1.1	C1.1.1	C1.1.2	C2	C2.1	C2.1.1	C2.1.2	C3	C3.1	C4	C4.1	C5	C5.1	C6	C6.1	C6.2
NFR 1	X	X	X	X													
NFR 2											X	X					
NFR 3					X	X	X	X									
NFR 4	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
NFR 5															X	X	X

Figure 7 *- Alignment between non-functional requirements and RedParkMiners structure*

	C1	C1.1	C1.1.1	C1.1.2	C2	C2.1	C2.1.1	C2.1.2	C3	C3.1	C4	C4.1	C5	C5.1	C6	C6.1	C6.2
UC1																	X
UC2					X	X	X	X							X	X	X
UC3	X	X	X	X													X
UC4	X	X	X	X													X
UC5									X	X					X	X	X
UC6											X	X					X
UC7													X	X			X
UC8	X	X	X	X													X
UC9					X	X	X	X							X	X	X
UC10									X	X					X	X	X
UC11											X	X					X
UC12													X	X			X

Figure 8 *- Relation between the use cases and the system components.*

3.5- Behavior

To represent the general behavior of RedParkMiner, two activity diagrams (Figure 9 and Figure 10) are presented bellow and are associated with the already presented use cases in order to understand where the use cases belong in terms of behavior.

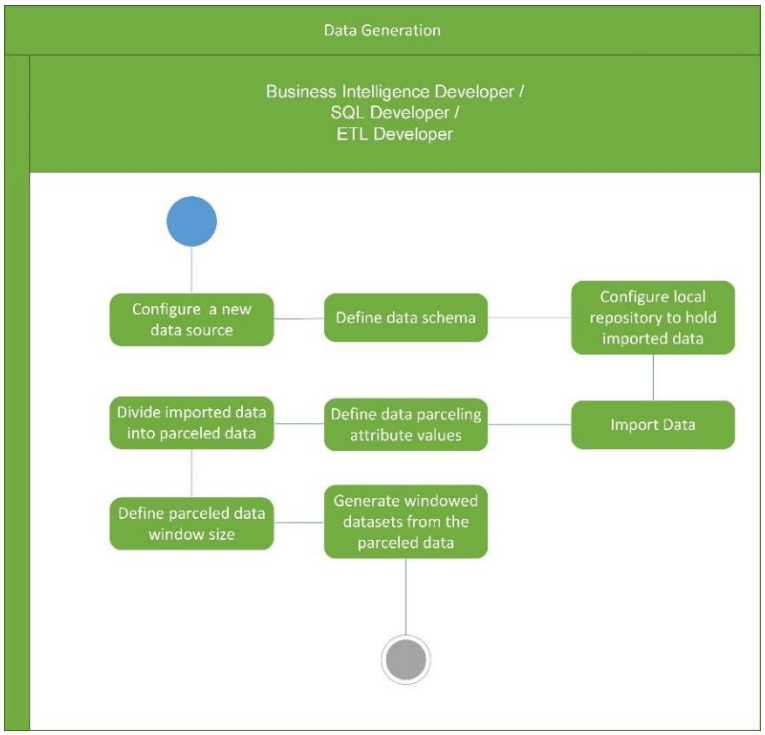

Figure 9 *- Data generation activity diagram*

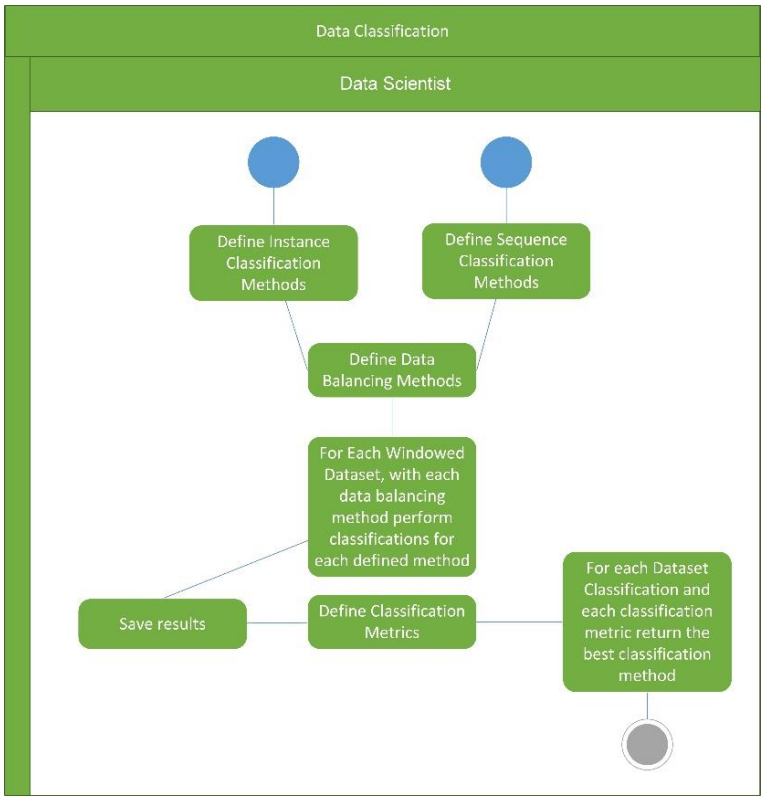

Figure 10 *- Data classification activity diagram*

	Data Generation Activity	Data Classification Activity
UC1	X	
UC2	X	
UC3	X	
UC4	X	
UC5	X	
UC6		X
UC7		X
UC8	X	
UC9	X	
UC10	X	
UC11		X
UC12		X

Figure 11 *- Relation between the use cases and the activities.*

As we can see in Figure 11, the separation is obvious because there is indeed a division between the part in RedParkMiner that consists mainly of data integration/generation, and the part that classifies the data and retrieves the best classification type.

3.6- State Machine

Finally, in the end of the requirement analysis a state machine diagram (Figure 10) is shown to share the necessary sequential flow in order for this tool to achieve its main objective, returning the best possible classification method, according to a specific metric, to mine a specific dataset.

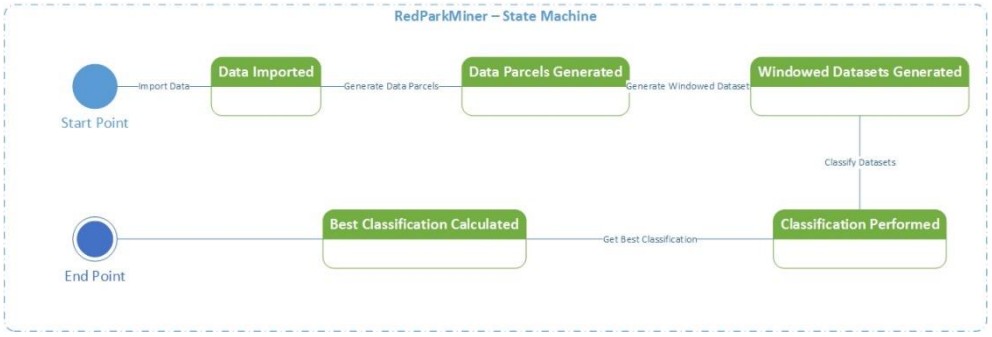

Figure 12 *- RedParkMiners state machine diagram*

4- System Architecture

In this chapter, we will approach the solution design as well as the decisions made to achieve it.

4.1- Overview

The main purpose of this tool is to provide Data Scientists, Business Intelligence Developers and Data Engineers the possibility of modular data analysis, by facilitating not only source data transformation process but also facilitating the data analysis, so we can consider this tool as being part data integration tool and part data mining tool.

This tool has been designed and created inside the car parking business scope, but it can be extended into a more general scope, so it is not dependent on the scope.

From our requirement analysis phase, we have decided to use open source cross platform technologies like Talend, Java, Weka and PostgreSQL not only to make the tool environment independent in terms of development and in terms of use but also to enable anyone to replicate this effort in order to extend and possibly enhance it. For the tools development to be aligned with the requirements, it has also been divided into various modules, to increase the tools extensibility. These modules can be grouped into two main modules, the Data Generation Process module (DGP module) and the Data Classification Process module (DCP module). In the DGP module, we will focus on the necessary data integration processes and subsequent data transformation processes in order to achieve a satisfying dataset structure to work in the DCP module. In the DCP module, the focus is to take advantage of the calculated datasets and develop/perform new/existing data mining approaches in order to understand which one will be the best to apply in a specific case.

In the bellow diagram (Figure 13), follows the technical solution, with all general components and technologies used until this point.

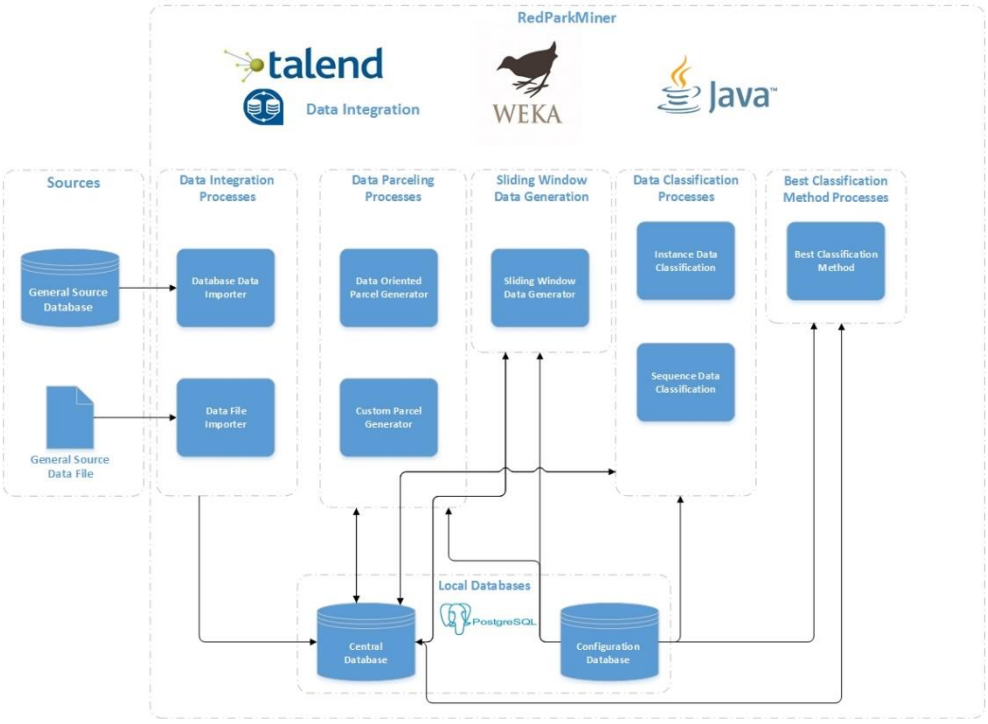

Figure 13 *- System specification*

4.2- Sources

Our technological stack decisions have been oriented to promote maximum compatibility/extensibility in what concerns our possible data origins. These data sources can have multiple representation types, even though in the diagram only the database type and file type are represented, because these are the types more commonly used.

4.3- Data Integration Processes

Depending on the source type and structure, these processes were conceived with the sole purpose of extracting the data that will be explored throughout the tools data pipeline. The development tool used (Talend) for our tools development has the necessary connectors to avoid limitations related to how the data is available, even though we depend on it to configure the referenced processes. In terms of data structure, the integration processes will convert the source data into normalized data, even if it is dealing with data warehouse data models or relational data models. In the

current development, the integration processes consist on retrieval of the normalized data related to short term parking sessions from a star schema in a Redshift Database, at Amazon Web Services.

4.4- Data Parceling Processes

Our data may sometimes be stratified, and we may want to explore each stratum separately, so in the Data Parceling Process our goal is to divide the already integrated dataset into previously configured parcels that correspond to the previous discussed stratums. The parceling process can be divided into two types, data oriented parceling and custom parceling. In data oriented parceling, we iterate over preconfigured values, associated to one or more dataset fields, and for each value, we will create a new dataset from the original dataset filtered by these values. Finally, we have custom parceling, which usually may not entirely correspond to a dataset direct stratum but corresponds to the original dataset complex filtering to create a new dataset. In the current development, the data is being parceled into six parts. Four parts are oriented to the data values and the other two parts are oriented to their occurrence occurring during the week or the weekend.

4.5- Sliding Window Data Generation

Before applying data classification methods to the calculated datasets, we need these to be labelled according to the criteria by which the classification will be performed, and considering the temporality of our datasets, we'll not only be interested in applying the chosen criteria but also being able to widen it, which is what is proposed with the sliding window data generation process.

Therefore, at this point on a global overview the goal is to generate preconfigured temporal windows for each parceled dataset. Currently the main concern of the sliding window process is to calculate for each dataset occurrence the intermediary occurrences (if applicable) that lead to the future occurrence we want to predict and that we will be labelling accordingly. This process will possibly be the most complex process to configure due to the set of attributes from the intermediary occurrences that must be present until the end of each temporal window calculation.

In the current development for each parceled dataset, we have calculated seven different temporal windows with an associated window size comprised between 1 and 7. To simplify how these temporal windows are structured we'll explain the structure for a window size equal to one and seven. The temporal window with a window size one will have information about a specific occurrence and a label describing if there is a following occurrence in the following period. The temporal window with a window size seven will have information about a specific occurrence, its consecutive six occurrences and if there's a following occurrence in the period following its last occurrence.

In the end of this process, we will have labelled datasets with multiple views over each data occurrence, which will be enough dataset material to perform our classification analysis.

4.6- Data Classification Processes

Usually, at our disposal there are classification libraries that are composed by a various set of classification methods, it's also usual that these methods are experimented separately in order to better understand their behavior with specific datasets, so this module was designed to make use of the existing libraries classification methods, applying the configured methods from these libraries to the already calculated datasets. Until this point, our focus was on the data generation/integration that we wanted to explore, now we will focus on applying configured classification methods to the previous discussed data. In this tools perspective, there are two ways of looking at data, looking at it as a set of instances, being each one an occurrence used as a whole to train the classifiers, or looking at it as a set sequences associated with a specific occurrence that dictate the outcome of the trained sequence classifiers. In these processes, classification methods will not be the only concern in terms of configuration, we will also have to deal with balancing methods in order to avoid possible misclassification due to biasing (even though this is not a sure thing). The goal of these processes will be to train, to classify all previous datasets according to the discussed perspectives and to save the respective results.

In the current development, for the instance oriented classification we have configured three classifiers and two data balancing methods. The instance oriented classification methods are Naïve Bayes, AdaBoost and Random Forest and the balancing methods we have used are Resample and SMOTE. For the sequence classification, we have used seven classification methods and two balancing methods. The sequence classification methods are HMMs, MDD, MIDD, MIEMDD, MISMO,

MISVN, QuickDDIterative and as used before the balancing methods are SMOTE and Resample.

4.7- Best Classification Method Processes

When we're experimenting classification methods with datasets our focus will always be on how well these methods behave in terms of classification, and for example, after binary classification is performed we'll have four important facts, the true positives which correspond to instances that are true and are classified as true, true false which correspond to instances that are false and are classified as false, the false positives which are false instances that are classified as true and the false negatives which are true instances classified as false. The manipulation of these facts is the key to understand how good our classification is, because they will help us answer questions like: "Overall, how often is the classifier correct?" (Accuracy)," When it's actually true, how often does it predict true?" (Sensitivity), "When it's actually false, how often does it predict false?" (Specificity), "When it predicts true, how often is it correct?" (Precision) or "When it predicts false, how often is it correct?" (Negative Predictive Value). So, in our final stage the tool will be focused on answering these questions not only with the best results obtained for each question above but also with the classification method that can achieve these results. From a global perspective, the best classification method processes will retrieve the best way to classify each existing dataset but unfortunately, in terms of what is the best classification method, it will always depend on the user's necessity. In order to give the users various perspectives best classification methods, we have predefined several metrics and for each metric and each dataset, it gets the classification maximum values and correspondent classification approach. The results are written into a table, which will then be used to enlighten the user what is the best method to apply to a specific dataset.

In the current development five metrics have been preconfigured, namely, accuracy, negative predictive value, precision, sensitivity and specificity.

4.8- Local Databases

Our local databases are a crucial component of our tool, we'll rely not only on their storage capabilities to save the results of all our processes but also on their capacity to handle extendable structures to enable dynamic configurations inherent to any of the developed processes. We have divided our repository structure into two

databases, a central database and a configuration database. Our central database will handle and save information regarding all the data integration processes, which means that this database will have to save the original data, the parceled data, the windowed parceled data and the results from the classifications performed with these datasets. Our configuration database will handle transversal configurations values that will be used by almost all our processes.

In the current development, the central database has been the divided, with the help of its schemas, into four parts. The first database part will be responsible to store a copy of all the data extracted from our sources, the second part composed by various schemas that represent the data parcels extracted from the sources copied data and their associated sliding window representations, the third part will be responsible to store our classification results and the finally the last part will hold the best classification results.

4.9- Tech Stack

As referred previously, our solution depends on open source cross platform technologies oriented to our objectives, but the already existing experience with these technologies was also an important factor. Our main development platform is the Talend Open Studio, designed essentially for data integration purposes, it provides not only a user-friendly interface, mainly because it's based on Eclipse, but it also provides the necessary components to safely extend our designs. The fact that it is open source and built on Java enables the use of other open source libraries and tools like Weka or Spark, which may provide us with a collection of machine learning algorithms for data mining tasks. In terms of data persistence, PostgreSQL was the choice even though there is no obvious reason to choose this database provider over others, as MySQL for example.

4.10- Installing and Using

To install the tool the destination must have an installed PostgreSQL instance running, with the necessary credentials for the tool to perform its actions, also Java (version 8 or greater) must be installed. It is advised to create a folder with the necessary SQL scripts to create the local databases architecture. To facilitate the database setup, it is advised to use Flyway to perform and manage the databases state in terms of

structure evolution. To perform developments on the tool, the tools project must be imported into Talend Open Studio. Finally, to use the tools capabilities, with Talend Open Studio build the project by selecting the five discussed processes (Data Integration, Data Parceling, Sliding Window Data Generation, Data Classification and Best Classification Method). The necessary assemblies and batch/shell scripts will compose the build. These batch/shell scripts can be used to execute each module separately.

4.11- Limitations

In terms of limitations, this tools major limitation concerns its high configurability, which may be complex. This can be a big issue especially if the user is not aware of the sources datasets capabilities.

Another major limitation may be the use of external libraries that are not interpreted by Java which by one hand may prove difficult to make it work and by other hand may not be even possible.

5- Case Study

In this chapter, we will analyze the results achieved with the application of the tool ***RedParkMiner*** to the car parking business scope. We will analyze the methodologies used by dataset, classification results by metric and dataset, and optimal methodology to apply by dataset.

It is important to refer that the goal of our tool is to provide information on the best methods and approaches to apply to the classification of configured datasets. According to a predefined set of metrics, we will analyze the resulting best classification approaches provided by RedParkMiner.

We will divide each of the approaches analysis into two parts, the first part we will analyze the parks datasets and in the second part we will analyze the weekdays datasets. In each of these analyses, we will start out by analyzing the set of methodologies that for each dataset maximize the predefined metrics. After understanding which methods must be applied to each dataset to maximize a specific metric, we will analyze their performance, by analyzing the results of the methods for each of the datasets. Finally, to understand which of the proposed methodologies for the underlying datasets are the optimal we will analyze ROC charts that explicitly show the relation between specificity and sensitivity. When analyzing these ROC charts we must understand that for each of the models, we only want those that are farthest from the 45º diagonal that crosses the chart (which represents a poor model of a random classifier), i.e. the optimal model will be the point closest to the point (X=0%, Y=100%).

In this chapter, we will only focus on the accuracy and specification metrics and the other already referred metrics results we will be available in the Appendix chapter. Accuracy was chosen detriment of sensitivity, because these provide approximate results due to the explored unbalanced data. Specificity was chosen due to the existent difficulty in maximizing this metric.

5.1- Instance Classification Results

In this section, due to the nature of the data explored, we will explore all the proposed metrics best results for the instance datasets.

In Figure 14 we will present the classification and data balancing methods used to maximize the proposed metrics.

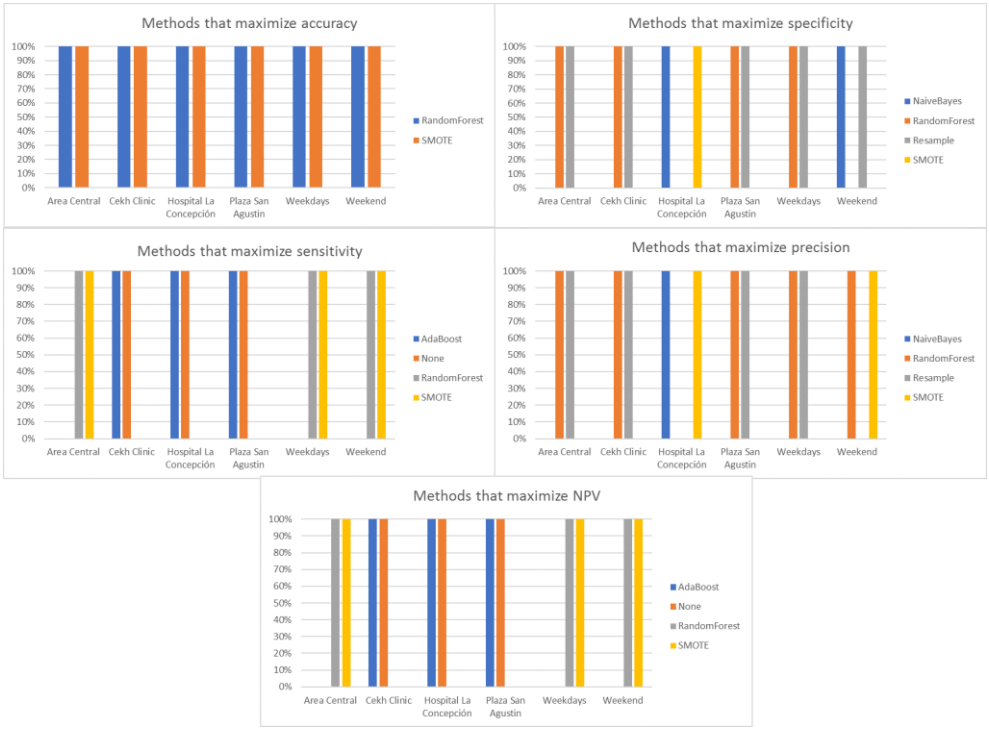

Figure 14 *- Classification and balancing methods by dataset and by maximized metric.*

As we can observe in Figure 14, only when maximizing accuracy there is a unanimous choice i.e. Random Forest as the classification method and SMOTE as a data balancing method. When maximizing specificity and precision Naïve Bayes and Random Forest are the chosen classification methods and SMOTE and Resample are the chosen data balancing methods. For sensitivity and NPV metric maximization, the chosen classification methods are AdaBoost and Random Forest, regarding the data balancing methods SMOTE and no balancing method are the choices.

In the instance classification case, we can say that when we are maximizing sensitivity we will also be maximizing NPV due to the methodology approaches chosen by both metrics maximization.

We will now focus on the metrics results for each of the maximized metric.

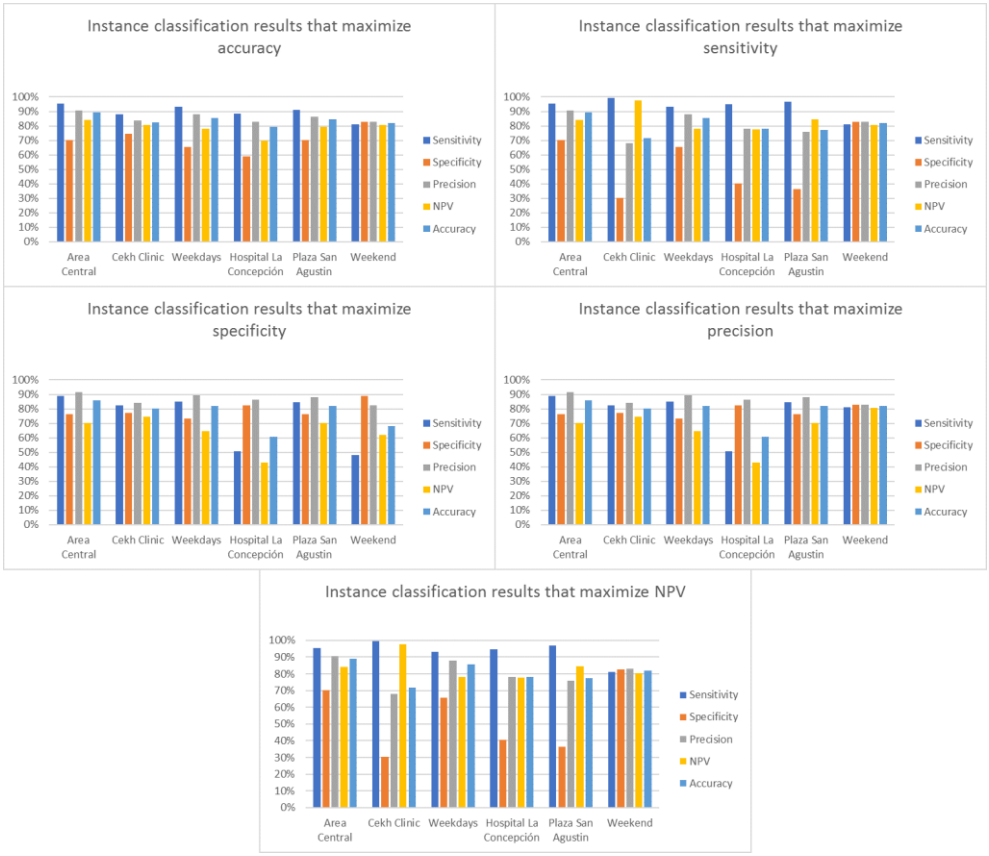

Figure 15 *- Instance classification results by dataset and maximized metric.*

By analyzing Figure 15, we can see that when accuracy is maximized, the averaged result value for this maximized metric will be approximately 85%, for sensitivity the averaged result value will be approximately 90%, for specificity the averaged result value will be approximately 70%, for precision the averaged result value will be approximately close to 85% and for NPV the averaged result value will approximately 80%.

When we consider the maximization of sensitivity and NPV we can see that sensitivity will have an averaged result value of approximately 95%, in terms of specificity the results averaged value will be approximately at 55%, in terms of precision and accuracy the averaged value will be approximately 80% and for NPV the averaged value will be approximately 85%.

For the maximization of specificity, the averaged result value for this maximized metric is approximately 80%, for sensitivity the averaged result value is approximately 75%, for precision approximate averaged result value will be close to 90%, in terms of NPV the averaged result value will be around 65% and in terms of accuracy the averaged value will be close to 75%.

Regarding the maximization of precision, the averaged result value for this maximized metric is approximately 90%, for sensitivity the averaged result value is approximately 80%, for specificity the averaged result value is approximately 80%, for NPV the averaged result value is approximately 70% and for accuracy the averaged result value is approximately 80%.

After the analysis of Figure 15, which focuses only on the maximized metrics impact on remaining metrics, we will now follow on the overall metric results for each of the datasets.

Starting with “Area Central” the metric maximization impact shows an averaged result value of approximately 90% for accuracy, precision and sensitivity, an approximate averaged result value of 70% for specificity and an 80% averaged result value for NPV.

For “Cekh Clinic” the accuracy averaged result value is of approximately 80%, for sensitivity the averaged result value is of approximately 90%, for specificity the averaged result value is of approximately 60%, for precision the averaged result value is of approximately 80% and for NPV the averaged result value is of approximately 85%.

“Hospital La Concepción” averaged result value for accuracy is at approximately 70%, as for sensitivity the averaged result value is approximately 75%, for specificity and NPV the averaged result value is approximately 60% and for precision the averaged result value is approximately 80%.

For “Plaza San Agustin”, the averaged result value for accuracy, precision and NPV is approximately 80%, for sensitivity the averaged result value is approximately 90% and for specificity the averaged result value is approximately 60%.

The weekday’s dataset averaged result value for accuracy is approximately 85%, for sensitivity and precision the averaged result value is approximately 90% and for specificity and NPV the averaged result value is approximately 70%.

For the weekend dataset, the averaged result value for accuracy is approximately 80%, as for precision and specificity the averaged result value is approximately 85% and for sensitivity and NPV the averaged result value is approximately at 75%.

After the analysis of the results provided by the maximization of the proposed metrics, for each dataset and each metric we will now analyze Figure 16, in order to understand which will be the optimal model to apply for the discussed metrics and datasets.

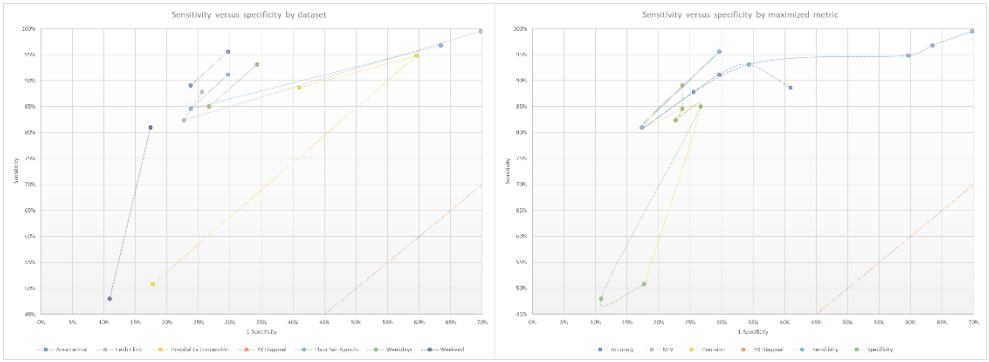

Figure 16 *- In this chart, we show the relation between the classification models sensitivity and specificity of the proposed metrics and the parks datasets.*

In terms of metrics it's difficult to understand the best model for each due to the result value overlapping, but considering that the optimal model will be the one whose values are closer to 100% in terms of sensitivity and specificity, we can see that when maximizing accuracy, NPV, precision and sensitivity the best model will be the one that uses Random Forest and SMOTE on the weekend dataset and whose specificity and sensitivity values are approximately 80%. The only maximized metric whose optimal model is different is specificity, the optimal model of this maximized metric uses Random Forest and Resample on Area Central dataset having a sensitivity value of approximately 90% and a specificity value of approximately 75%.

In terms of datasets, the optimal model for "Area Central" is the one that maximizes precision and specificity with the use of Random Forest and Resample leading to a sensitivity value of approximately 90% and a specificity value of approximately 75%. For "Cekh Clinic" the optimal model maximizes accuracy using Random Forest and SMOTE to achieve a sensitivity value of approximately 90% and a specificity value of approximately 75%. For "Hospital La Concepción" the optimal model follows the same methodology as in "Cekh Clinic", but with lower values for sensitivity and specificity i.e. sensitivity has a value of approximately 90% and specificity has a value of approximately 60%. "Plaza San Austin" and the weekdays' dataset optimal model uses the same methodology, which consists on the maximization of precision and specificity by using Random Forest with Resample achieving approximately 75% of specificity and 85% of sensitivity. For the weekend dataset, the optimal model maximizes accuracy, NPV, precision and sensitivity by using Random Forest and Resample achieving approximately a value of 80% for sensitivity and specificity.

Even though there is not a unanimous choice, in terms of data balancing methods, concerning the optimal model from the analysis of Figure 16, the clear choice for classification is Random Forest.

In Table 16 we can analyze more closely the results from the instance data classification.

Maximized Metric	Classifier	Balancing	Dataset	Sensitivity	Specificity	Precision	NPV	Accuracy
Accuracy	RandomForest	SMOTE	Area Central	95,55%	70,36%	90,51%	84,23%	89,19%
Accuracy	**RandomForest**	**SMOTE**	**Cekh Clinic**	**87,86%**	**74,42%**	**83,58%**	**80,54%**	**82,44%**
Accuracy	RandomForest	SMOTE	Weekdays	93,09%	65,69%	87,86%	78,09%	85,61%
Accuracy	**RandomForest**	**SMOTE**	**Hospital La Concepción**	**88,61%**	**59,05%**	**82,90%**	**69,83%**	**79,49%**
Accuracy	RandomForest	SMOTE	Plaza San Agustin	91,17%	70,35%	86,47%	79,31%	84,41%
Accuracy	**RandomForest**	**SMOTE**	**Weekend**	**81,00%**	**82,64%**	**83,04%**	**80,56%**	**81,80%**
Sensitivity	RandomForest	SMOTE	Area Central	95,55%	70,36%	90,51%	84,23%	89,19%
Sensitivity	**AdaBoost**	**None**	**Cekh Clinic**	**99,49%**	**30,26%**	**67,88%**	**97,55%**	**71,59%**
Sensitivity	RandomForest	SMOTE	Weekdays	93,09%	65,69%	87,86%	78,09%	85,61%
Sensitivity	**AdaBoost**	**None**	**Hospital La Concepción**	**94,84%**	**40,41%**	**78,09%**	**77,76%**	**78,04%**
Sensitivity	AdaBoost	None	Plaza San Agustin	96,78%	36,56%	76,02%	84,53%	77,22%
Sensitivity	**RandomForest**	**SMOTE**	**Weekend**	**81,00%**	**82,64%**	**83,04%**	**80,56%**	**81,80%**
Specificity	RandomForest	Resample	Area Central	89,08%	76,26%	91,74%	70,23%	85,84%
Specificity	**RandomForest**	**Resample**	**Cekh Clinic**	**82,38%**	**77,30%**	**84,32%**	**74,76%**	**80,33%**
Specificity	RandomForest	Resample	Weekdays	84,99%	73,34%	89,47%	64,70%	81,81%
Specificity	**NaiveBayes**	**SMOTE**	**Hospital La Concepción**	**50,79%**	**82,26%**	**86,51%**	**42,74%**	**60,51%**
Specificity	RandomForest	Resample	Plaza San Agustin	84,58%	76,24%	88,09%	70,41%	81,87%
Specificity	**NaiveBayes**	**Resample**	**Weekend**	**48,06%**	**89,10%**	**82,23%**	**62,05%**	**68,09%**
Precision	RandomForest	Resample	Area Central	89,08%	76,26%	91,74%	70,23%	85,84%
Precision	**RandomForest**	**Resample**	**Cekh Clinic**	**82,38%**	**77,30%**	**84,32%**	**74,76%**	**80,33%**
Precision	RandomForest	Resample	Weekdays	84,99%	73,34%	89,47%	64,70%	81,81%
Precision	**NaiveBayes**	**SMOTE**	**Hospital La Concepción**	**50,79%**	**82,26%**	**86,51%**	**42,74%**	**60,51%**
Precision	RandomForest	Resample	Plaza San Agustin	84,58%	76,24%	88,09%	70,41%	81,87%
Precision	**RandomForest**	**SMOTE**	**Weekend**	**81,00%**	**82,64%**	**83,04%**	**80,56%**	**81,80%**
NPV	RandomForest	SMOTE	Area Central	95,55%	70,36%	90,51%	84,23%	89,19%
NPV	**AdaBoost**	**None**	**Cekh Clinic**	**99,49%**	**30,26%**	**67,88%**	**97,55%**	**71,59%**
NPV	RandomForest	SMOTE	Weekdays	93,09%	65,69%	87,86%	78,09%	85,61%
NPV	**AdaBoost**	**None**	**Hospital La Concepción**	**94,84%**	**40,41%**	**78,09%**	**77,76%**	**78,04%**
NPV	AdaBoost	None	Plaza San Agustin	96,78%	36,56%	76,02%	84,53%	77,22%
NPV	**RandomForest**	**SMOTE**	**Weekend**	**81,00%**	**82,64%**	**83,04%**	**80,56%**	**81,80%**

Table 16 - *This table provides the classifiers, balancing types and respective results, which maximize each proposed metric, for each instance dataset.*

5.2- Time-Window based Classification Results

In this section, we will analyze the best results, by each defined metric, obtained through the classification of the time-window based data from the datasets.

5.2.1- Accuracy maximization results

In what concerns accuracy maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in Figure 17.

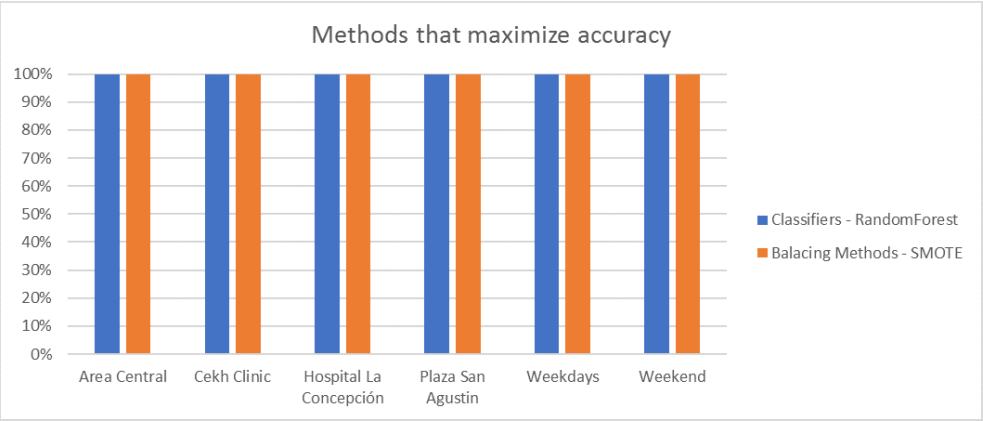

Figure 17 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize accuracy.*

The analysis to be made to Figure 17 is simple, Random Forest and SMOTE are the methods that best maximize accuracy for all the explored datasets. In future analysis we will be focused on possible methods that would optimally maximize the studied metric, in accuracy's case we will focus on the transversal metric results and the temporal window best suited to achieve the proposed maximization.

Now let us pay attention to the results from the parks datasets, shown in Figure 18, of the application of the above classification methods and balancing methods.

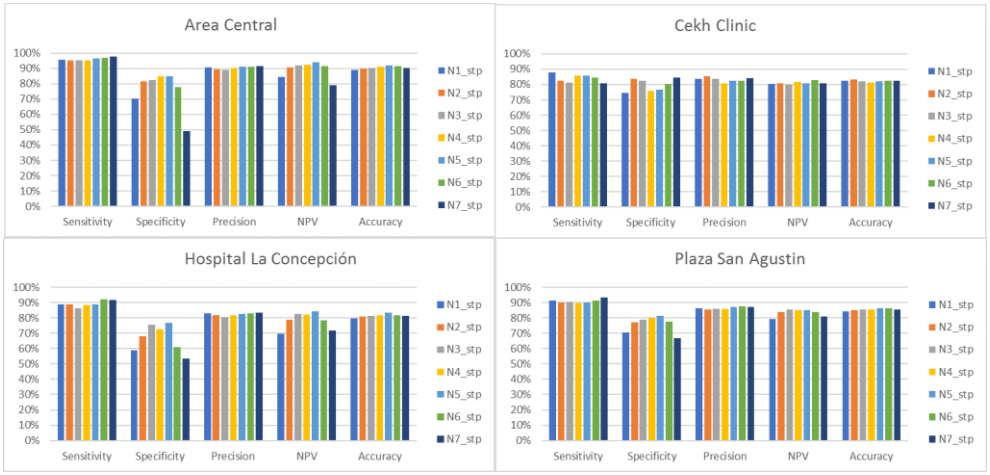

Figure 18 *- This set of charts shows the classification results, which maximize accuracy, by each defined metric, by each temporal window and by each park dataset.*

In Figure 18, with the already described maximization attempt of accuracy, we've achieved a mean value of approximately 90% for sensitivity, for specificity he mean value is approximately 75%, in terms of precision, the mean value is at approximately

at 85%, in terms of NPV the mean value is at approximately at 85% and concerning accuracy, which is the metric we are maximizing, the mean value is approximately 85%.

We will now focus on Figure 19, that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in Figure 18 provides the optimal model.

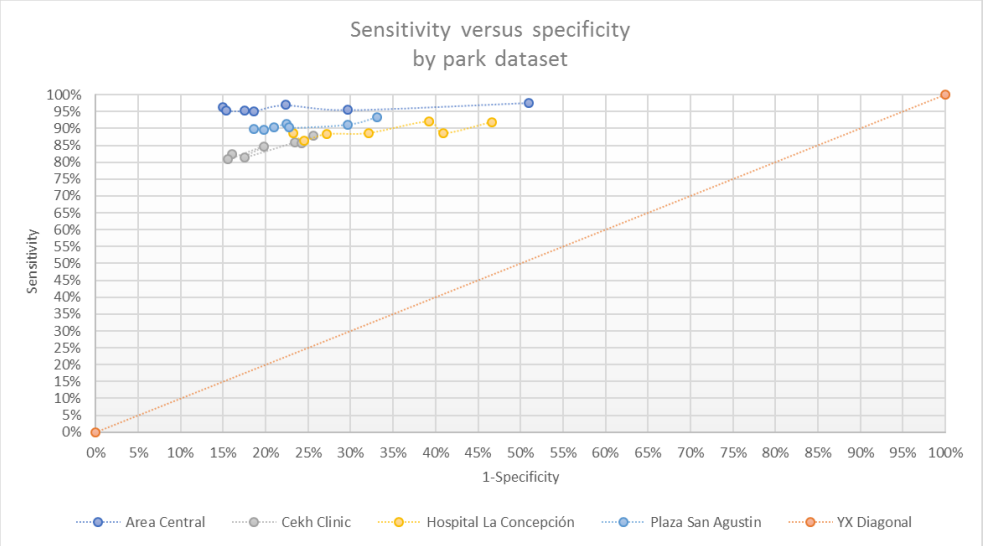

Figure 19 *- In this chart, we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing Figure 19, we can conclude that for “Area Central” the optimal model has approximately 95% sensitivity and 85% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 90% sensitivity and 80% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 90% sensitivity and 75% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 80% sensitivity and 85% specificity, it corresponds to the N2_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

For the weekday's datasets, we already know Random Forest and SMOTE will be the unanimous methods, there is only left to know if the optimal model results will be as good as the parks datasets results.

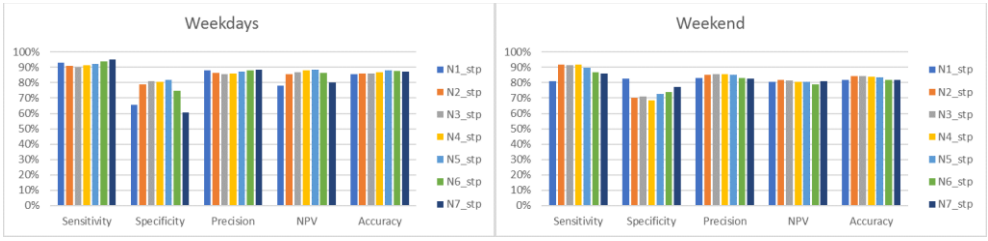

Figure 20 *- This set of charts shows the classification results, that maximize accuracy, by each defined metric, by each temporal window and by each weekday type dataset.*

In Figure 20, we explore the weekdays' datasets, also with the goal of maximizing accuracy, which leads to a mean value of approximately 90% for sensitivity, a mean value of approximately 75% for specificity, a mean value of approximately 85% for precision, approximately 85% mean value for NPV and approximately 85% of mean value for accuracy.

Compared to the parks datasets results mean values, for accuracy maximization, the results are approximately the same.

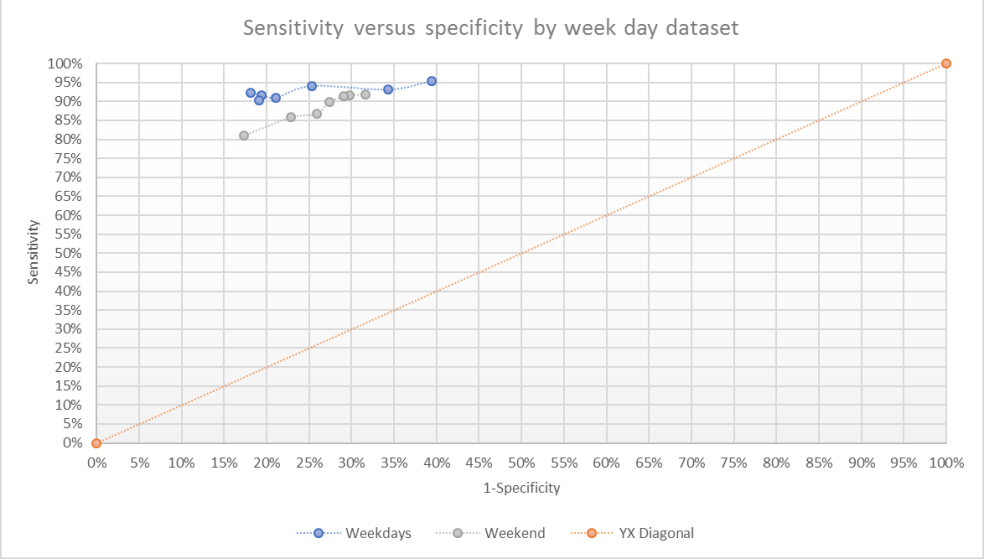

Figure 21 *- In this chart, we show the relation between the classification models sensitivity and specificity of the weekdays' datasets.*

In Figure 21, we can see that the optimal model for the weekdays dataset has approximately 95% for sensitivity and 80% for specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 80% for sensitivity and approximately 80% for specificity, it corresponds to the N1_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

In sum, for this case, we can say that when we are maximizing accuracy the optimal model will reside on the choice of Random Forest and as SMOTE for data balancing.

The explored results in accuracy maximization can be explored in Table 17, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
RandomForest	SMOTE	Area Central	N1_stp	95,55%	70,36%	90,51%	84,23%	89,19%
RandomForest	SMOTE	Area Central	N2_stp	95,14%	81,37%	89,61%	90,83%	90,01%
RandomForest	SMOTE	Area Central	N3_stp	95,20%	82,45%	89,14%	91,91%	90,13%
RandomForest	SMOTE	Area Central	N4_stp	95,40%	84,67%	90,37%	92,43%	91,12%
RandomForest	SMOTE	Area Central	N5_stp	96,43%	84,99%	90,98%	93,81%	91,98%
RandomForest	SMOTE	Area Central	N6_stp	96,99%	77,64%	91,26%	91,47%	91,31%
RandomForest	SMOTE	Area Central	N7_stp	97,66%	49,07%	91,53%	78,82%	90,34%
RandomForest	SMOTE	Cekh Clinic	N1_stp	87,86%	74,42%	83,58%	80,54%	82,44%
RandomForest	SMOTE	Cekh Clinic	N2_stp	82,44%	83,90%	85,43%	80,67%	83,12%
RandomForest	SMOTE	Cekh Clinic	N3_stp	81,36%	82,47%	83,74%	79,94%	81,88%
RandomForest	SMOTE	Cekh Clinic	N4_stp	85,71%	75,76%	80,64%	81,82%	81,14%
RandomForest	SMOTE	Cekh Clinic	N5_stp	85,98%	76,51%	82,56%	80,84%	81,85%
RandomForest	SMOTE	Cekh Clinic	N6_stp	84,71%	80,18%	82,32%	82,79%	82,54%
RandomForest	SMOTE	Cekh Clinic	N7_stp	80,80%	84,41%	84,27%	80,96%	82,57%
RandomForest	SMOTE	Hospital La Concepción	N1_stp	88,61%	59,05%	82,90%	69,83%	79,49%
RandomForest	SMOTE	Hospital La Concepción	N2_stp	88,66%	67,89%	81,79%	78,63%	80,75%
RandomForest	SMOTE	Hospital La Concepción	N3_stp	86,46%	75,51%	80,68%	82,50%	81,44%
RandomForest	SMOTE	Hospital La Concepción	N4_stp	88,39%	72,78%	81,64%	82,07%	81,80%
RandomForest	SMOTE	Hospital La Concepción	N5_stp	88,66%	76,74%	82,64%	84,42%	83,36%
RandomForest	SMOTE	Hospital La Concepción	N6_stp	92,01%	60,76%	83,01%	78,48%	81,88%
RandomForest	SMOTE	Hospital La Concepción	N7_stp	91,87%	53,32%	83,59%	71,71%	81,13%
RandomForest	SMOTE	Plaza San Agustin	N1_stp	91,17%	70,35%	86,47%	79,31%	84,41%
RandomForest	SMOTE	Plaza San Agustin	N2_stp	90,23%	77,22%	85,64%	84,01%	85,04%
RandomForest	SMOTE	Plaza San Agustin	N3_stp	90,33%	79,01%	85,74%	85,40%	85,61%
RandomForest	SMOTE	Plaza San Agustin	N4_stp	89,71%	80,18%	86,03%	85,13%	85,67%
RandomForest	SMOTE	Plaza San Agustin	N5_stp	89,87%	81,33%	86,99%	85,26%	86,29%
RandomForest	SMOTE	Plaza San Agustin	N6_stp	91,43%	77,51%	87,54%	83,95%	86,32%
RandomForest	SMOTE	Plaza San Agustin	N7_stp	93,44%	66,88%	87,02%	81,08%	85,57%
RandomForest	SMOTE	Weekdays	N1_stp	93,09%	65,69%	87,86%	78,09%	85,61%
RandomForest	SMOTE	Weekdays	N2_stp	90,96%	78,91%	86,22%	85,75%	86,04%
RandomForest	SMOTE	Weekdays	N3_stp	90,20%	80,92%	85,72%	86,67%	86,11%
RandomForest	SMOTE	Weekdays	N4_stp	91,52%	80,61%	86,08%	87,88%	86,79%
RandomForest	SMOTE	Weekdays	N5_stp	92,34%	81,86%	87,42%	88,68%	87,91%
RandomForest	SMOTE	Weekdays	N6_stp	94,04%	74,66%	87,92%	86,46%	87,50%
RandomForest	SMOTE	Weekdays	N7_stp	95,32%	60,59%	88,54%	80,22%	87,04%
RandomForest	SMOTE	Weekend	N1_stp	81,00%	82,64%	83,04%	80,56%	81,80%
RandomForest	SMOTE	Weekend	N2_stp	91,67%	70,27%	85,13%	81,96%	84,18%
RandomForest	SMOTE	Weekend	N3_stp	91,38%	70,88%	85,39%	81,53%	84,22%
RandomForest	SMOTE	Weekend	N4_stp	91,84%	68,34%	85,40%	80,59%	84,05%
RandomForest	SMOTE	Weekend	N5_stp	89,85%	72,63%	85,00%	80,57%	83,53%
RandomForest	SMOTE	Weekend	N6_stp	86,74%	74,05%	83,26%	78,95%	81,64%
RandomForest	SMOTE	Weekend	N7_stp	85,79%	77,11%	82,82%	80,83%	81,99%

Table 17 *- This table provides the classifiers, balancing types and respective results that maximize accuracy for each dataset and temporal window.*

5.2.2- Specificity maximization results

In terms of specificity maximization, the classification approaches and data balancing approaches chosen for the configured datasets are shown in Figure 22.

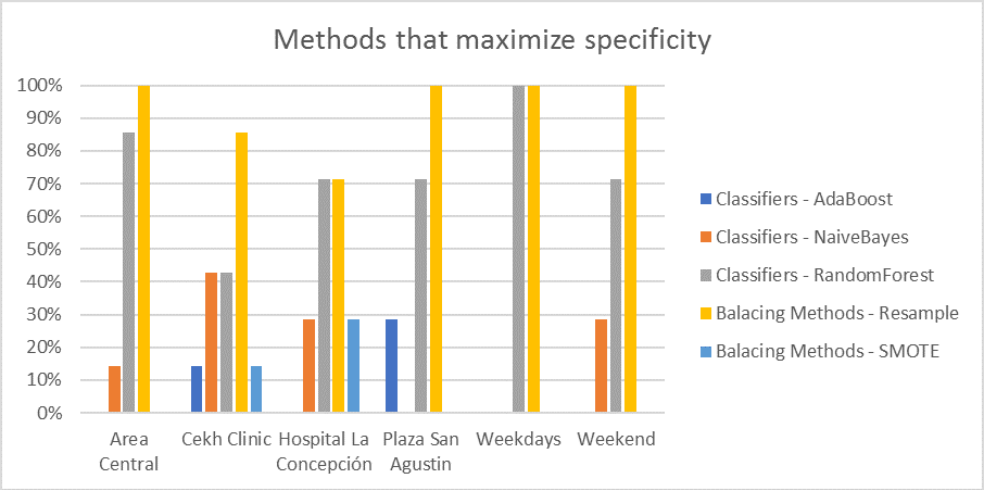

Figure 22 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize specificity.*

Starting with “Area Central” datasets, we can observe that approximately 85% of the datasets maximize specificity by using Random Forest and Resample, the other 15% gain by using Naïve Bayes and Resample.

In “Cekh Clinic” in terms of classifiers all classifiers are used, being Naïve Bayes and Random Forest the prevailing classifiers with approximately 40% use, leaving AdaBoost with approximately 15% use. Resample is the prevailing balancing method, with approximately 85% use and the other approximately 15% is the use of SMOTE for data balancing.

In “Hospital La Concepción”, in terms of balancing we can see that Resample is also the most used balancing method, with approximately 70% use, being SMOTE the other balancing method used with the other 30% of the classifiers. In terms of classifiers for “Hospital La Concepción”, Random Forest is the most used classifier, with 70% use, and Naïve Bayes the least used classifier with 30% use.

In “Plaza San Agustin” Resample is the chosen balancing method for all the classifiers, which are Random Forest and AdaBoost with approximately 70% and 30% use, respectively.

In the weekdays, the classifier and balancing method chosen for all the datasets are Random Forest and Resample. In the weekend's classification, Resample is the chosen balancing method for Random Forest, with approximately 70% use, and Naïve Bayes, with approximately 30% use.

From the observation and description of Figure 22, we can state that, in terms of specificity, there is not a transversal set of methodologies to apply, even though Resample and Random Forest appear to be the most predominant methods.

Now we will pay attention to the results from the parks datasets, shown in Figure 23, of the application of the above classification methods and balancing methods.

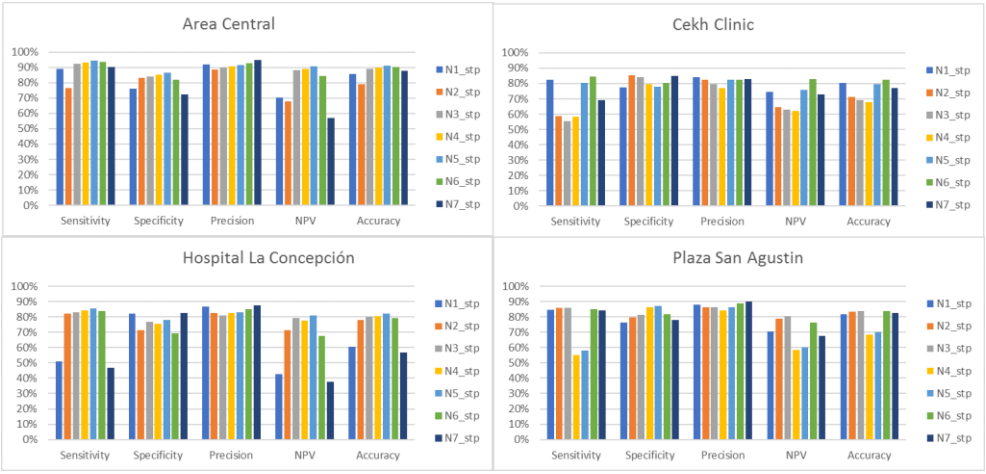

Figure 23 *- This set of charts shows the classification results that maximize specificity, by each defined metric, by each temporal window and by each park dataset.*

In Figure 23, with the already described maximization attempt of specificity, we have achieved a mean value of approximately 80% for accuracy, sensitivity and for specificity, which is the metric we are maximizing, in terms of precision the mean value is at approximately at 85% and in terms of NPV the mean value is at approximately at 70%.

We will now focus on Figure 24, which will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in Figure 23 provides the optimal model.

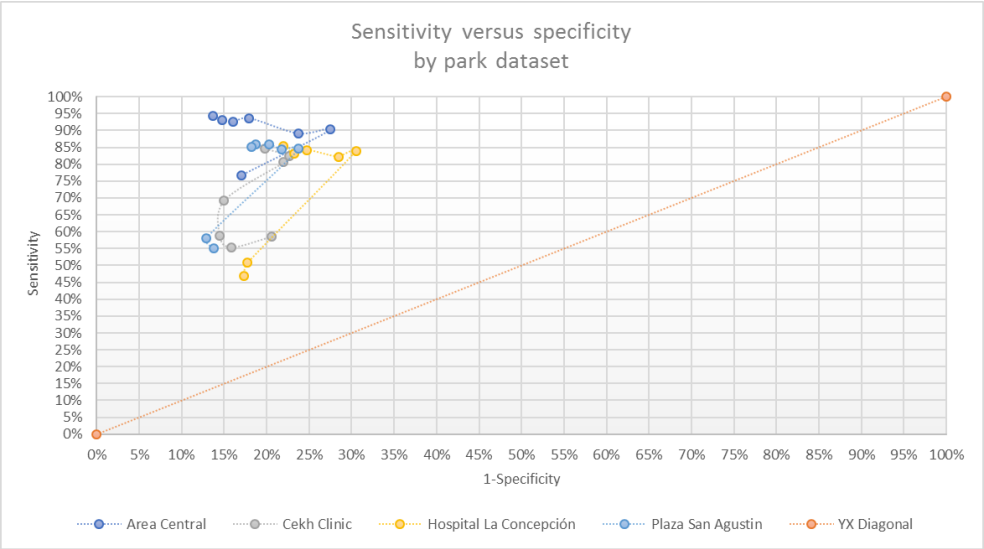

Figure 24 *- In this chart, we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing Figure 24 we can conclude that for “Area Central” the optimal model has approximately 95% sensitivity and 85% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 85% sensitivity and 80% specificity, it corresponds to the N6_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 85% sensitivity and 80% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 85% sensitivity and 80% specificity, it corresponds to the N6_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

In terms of classifiers Random Forest was the unanimous choice as the optimal model for the parks datasets classification, and even though it was expected that Resample would be the unanimous balancing method chosen, it was just the most used in the optimal models.

For the weekdays and weekends datasets, by analyzing Figure 23 we already know Resample will be the unanimous balancing method, and that at least for the weekdays datasets Random Forest will be present on the optimal model for the specified datasets.

The only aspects left to know are the temporal window of the optimal model and if the weekday's results will be as good as the parks datasets results and what will be the optimal model classifier for the weekend datasets.

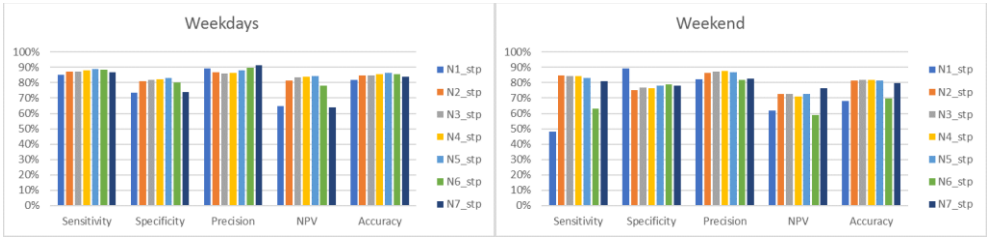

Figure 25 *- This set of charts shows the classification results, that maximize specificity, by each defined metric, by each temporal window and by each weekday type dataset.*

In Figure 25, we explore the weekdays' datasets, also with the goal of maximizing specificity, which leads to a mean value of approximately 80% for sensitivity and specificity, a mean value of approximately 85% for precision, approximately 75% mean value for NPV and approximately 80% of mean value for accuracy.

Compared to the parks datasets results mean values, for specificity maximization, the weekdays' dataset results mean values maintain approximately the same value for all metrics with the exception of NPV that shows an increase of 5% in terms of mean value.

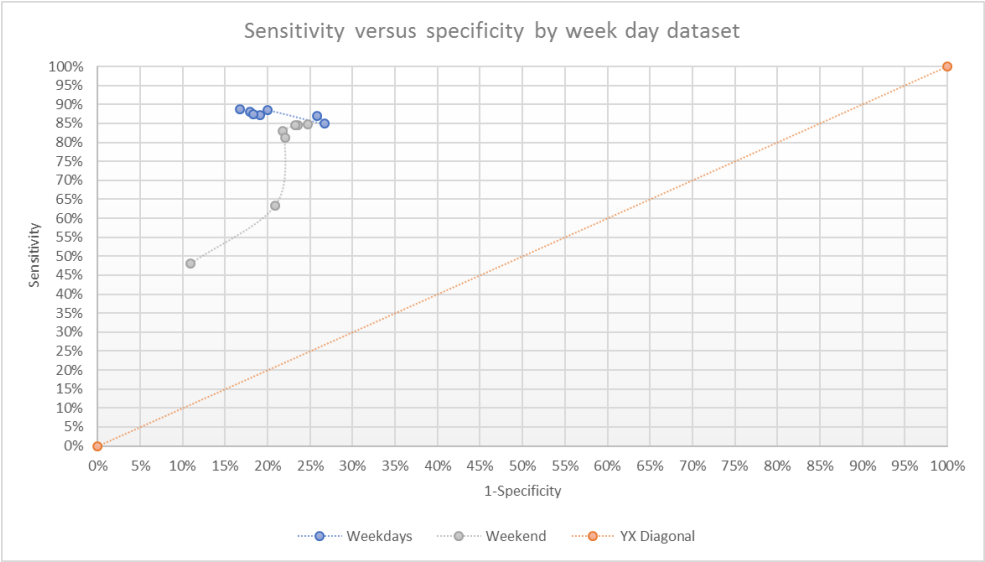

Figure 26 *- In this chart, we show the relation between the classification models sensitivity and specificity of the weekdays' datasets.*

In Figure 26, we can see that the optimal model for the weekday's dataset has approximately 90% for sensitivity and 85% for specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 85% for sensitivity and approximately 80% for specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

In sum, for this case, we can say that when we are maximizing specificity the optimal model will reside on the choice of Random Forest as classifier and Resample as the data balancing method.

The explored results in specificity maximization can be explored in Table 18, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
RandomForest	Resample	Area Central	N1_stp	89,08%	76,26%	91,74%	70,23%	85,84%
NaiveBayes	Resample	Area Central	N2_stp	76,62%	82,99%	88,38%	67,76%	78,99%
RandomForest	Resample	Area Central	N3_stp	92,49%	83,95%	89,71%	88,08%	89,09%
RandomForest	Resample	Area Central	N4_stp	93,15%	85,19%	90,46%	89,19%	89,98%
RandomForest	Resample	Area Central	N5_stp	94,30%	86,30%	91,53%	90,61%	91,19%
RandomForest	Resample	Area Central	N6_stp	93,70%	82,10%	92,64%	84,41%	90,29%
RandomForest	Resample	Area Central	N7_stp	90,29%	72,49%	94,87%	56,97%	87,61%
RandomForest	Resample	Cekh Clinic	N1_stp	82,38%	77,30%	84,32%	74,76%	80,33%
NaiveBayes	Resample	Cekh Clinic	N2_stp	58,87%	85,56%	82,36%	64,51%	71,32%
NaiveBayes	Resample	Cekh Clinic	N3_stp	55,31%	84,17%	79,50%	62,92%	68,99%
AdaBoost	Resample	Cekh Clinic	N4_stp	58,49%	79,41%	76,99%	61,89%	68,09%
RandomForest	Resample	Cekh Clinic	N5_stp	80,55%	78,02%	82,58%	75,62%	79,45%
RandomForest	SMOTE	Cekh Clinic	N6_stp	84,71%	80,18%	82,32%	82,79%	82,54%
NaiveBayes	Resample	Cekh Clinic	N7_stp	69,30%	85,00%	82,68%	72,81%	77,02%
NaiveBayes	SMOTE	Hospital La Concepción	N1_stp	50,79%	82,26%	86,51%	42,74%	60,51%
RandomForest	Resample	Hospital La Concepción	N2_stp	82,21%	71,54%	82,46%	71,20%	78,15%
RandomForest	Resample	Hospital La Concepción	N3_stp	83,02%	76,72%	80,83%	79,25%	80,13%
RandomForest	Resample	Hospital La Concepción	N4_stp	84,21%	75,29%	82,36%	77,69%	80,45%
RandomForest	Resample	Hospital La Concepción	N5_stp	85,37%	77,99%	82,88%	81,03%	82,09%
RandomForest	Resample	Hospital La Concepción	N6_stp	83,89%	69,43%	85,12%	67,40%	79,20%
NaiveBayes	SMOTE	Hospital La Concepción	N7_stp	46,84%	82,70%	87,51%	37,54%	56,84%
RandomForest	Resample	Plaza San Agustin	N1_stp	84,58%	76,24%	88,09%	70,41%	81,87%
RandomForest	Resample	Plaza San Agustin	N2_stp	85,86%	79,70%	86,43%	78,92%	83,40%
RandomForest	Resample	Plaza San Agustin	N3_stp	85,80%	81,26%	86,48%	80,38%	83,91%
AdaBoost	Resample	Plaza San Agustin	N4_stp	55,05%	86,17%	84,42%	58,49%	68,24%
AdaBoost	Resample	Plaza San Agustin	N5_stp	58,10%	87,14%	86,25%	59,96%	70,26%
RandomForest	Resample	Plaza San Agustin	N6_stp	85,25%	81,77%	88,99%	76,23%	83,97%
RandomForest	Resample	Plaza San Agustin	N7_stp	84,38%	78,19%	90,19%	67,80%	82,55%
RandomForest	Resample	Weekdays	N1_stp	84,99%	73,34%	89,47%	64,70%	81,81%
RandomForest	Resample	Weekdays	N2_stp	87,27%	80,90%	86,89%	81,41%	84,67%
RandomForest	Resample	Weekdays	N3_stp	87,39%	81,65%	85,81%	83,61%	84,86%
RandomForest	Resample	Weekdays	N4_stp	88,15%	82,11%	86,59%	84,09%	85,53%
RandomForest	Resample	Weekdays	N5_stp	88,78%	83,27%	87,87%	84,46%	86,45%
RandomForest	Resample	Weekdays	N6_stp	88,46%	79,97%	89,65%	77,93%	85,59%
RandomForest	Resample	Weekdays	N7_stp	87,01%	74,14%	91,49%	64,11%	83,94%
NaiveBayes	Resample	Weekend	N1_stp	48,06%	89,10%	82,23%	62,05%	68,09%
RandomForest	Resample	Weekend	N2_stp	84,70%	75,22%	86,39%	72,59%	81,38%
RandomForest	Resample	Weekend	N3_stp	84,47%	76,70%	87,11%	72,61%	81,76%
RandomForest	Resample	Weekend	N4_stp	84,43%	76,46%	87,85%	70,89%	81,79%
RandomForest	Resample	Weekend	N5_stp	83,02%	78,20%	86,79%	72,74%	81,25%
NaiveBayes	Resample	Weekend	N6_stp	63,23%	79,11%	81,83%	59,12%	69,61%
RandomForest	Resample	Weekend	N7_stp	81,19%	77,92%	82,55%	76,31%	79,76%

Table 18 *- This table provides the classifiers, balancing types and respective results that maximize specificity for each dataset and temporal window.*

5.3- Multi-Instance Classification Results

In this section, we will analyze the best results, by each defined metric, obtained through the classification of the multi-instance data from the datasets.

5.3.1- Accuracy maximization results

Regarding accuracy maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in Figure 27.

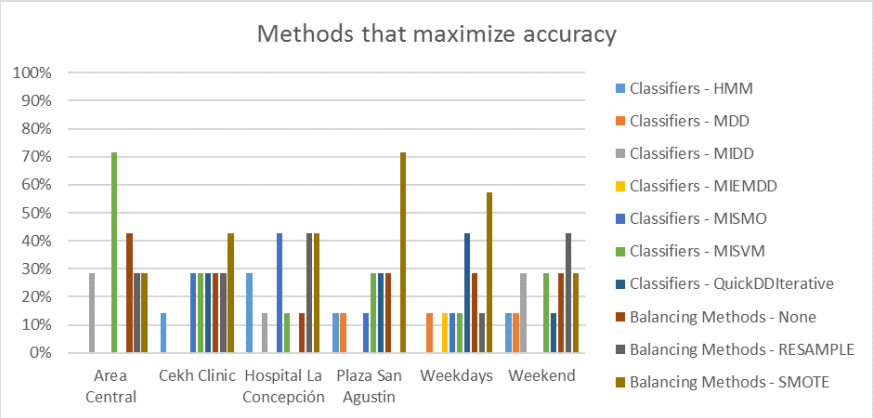

Figure 27 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize accuracy.*

The analysis to be made to Figure 27 is complex, due to the diversity of methods that maximize accuracy. We can see that MISVM has an approximate 30% overall classification use. Following MISVM in terms of use is QuickDDIterative and MISMO that even though are not present in all datasets temporal window classification their overall use is approximately 20% and 15% respectively. The least used classification methods are HMM and MIDD with approximately 10% use, MDD with approximately 8% use and MIEMDD with 2% use. In terms of balancing methods, SMOTE is the most used method with an overall use of approximately 45%, followed by no type of data balancing with approximately 30% use and Resample with approximately 25% use.

Now let us pay attention to the results from the parks datasets, shown in Figure 28, of the application of the above classification methods and balancing methods.

Figure 28 *- This set of charts shows the classification results that maximize accuracy, by each defined metric, by each temporal window and by each park dataset.*

In Figure 28, the maximization attempt of accuracy has led in most cases to the implicit maximization of sensitivity and in fewer cases the maximization specificity. In terms of overall results, we've achieved a mean value of approximately 80% for sensitivity, for specificity the mean value is approximately 25%, in terms of precision, the mean value is at approximately at 65%, in terms of NPV the mean value is at approximately at 55% and concerning accuracy, which is the metric we are maximizing, the mean value is approximately 60%.

We will now focus on Figure 29, that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in Figure 28 provides the optimal model.

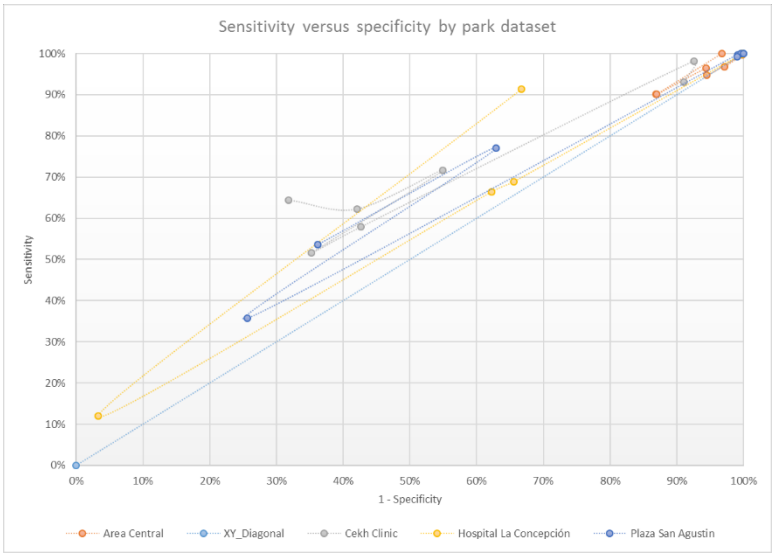

Figure 29 *- In this chart, we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing Figure 29, we can conclude that for “Area Central” the optimal model has approximately 90% sensitivity and 15% specificity, it corresponds to the N4_stp temporal window used to train and test the MIDD classifier with no data balancing.

As for “Plaza San Agustin”, the optimal model has approximately 55% sensitivity and 65% specificity, it corresponds to the N3_stp temporal window used to train and test the MDD classifier with no data balancing.

For “Hospital La Concepción”, the optimal model has approximately 90% sensitivity and 35% specificity, it corresponds to the N4_stp temporal window used to train and test the MISVM classifier with RESAMPLE as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 65% sensitivity and 70% specificity, it corresponds to the N7_stp temporal window used to train and test the QuickDDIterative classifier with RESAMPLE as a training data balancing method.

We will now explore the discussed the results as well as the optimal model for the weekdays’ datasets.

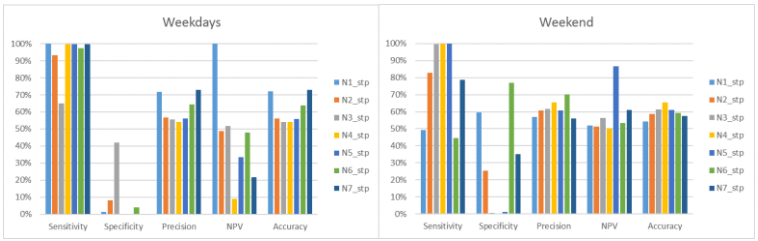

Figure 30 *- This set of charts shows the classification results, that maximize accuracy, by each defined metric, by each temporal window and by each weekday type dataset.*

In Figure 30, we explore the week days’ datasets, also with the goal of maximizing accuracy, we can observe that the behavior is similar than the one present in the parks datasets i.e. the implicit maximization of either sensitivity or specificity, which leads to a mean value of approximately 85% for sensitivity, a mean value of approximately 20% for specificity, a mean value of approximately 60% for precision, approximately 50% mean value for NPV and approximately 60% of mean value for accuracy.

Compared to the parks datasets results mean values, the results are approximately the same, except for specificity that shows an increase of 5%, as well as precision and NPV that show a decrease in 5%.

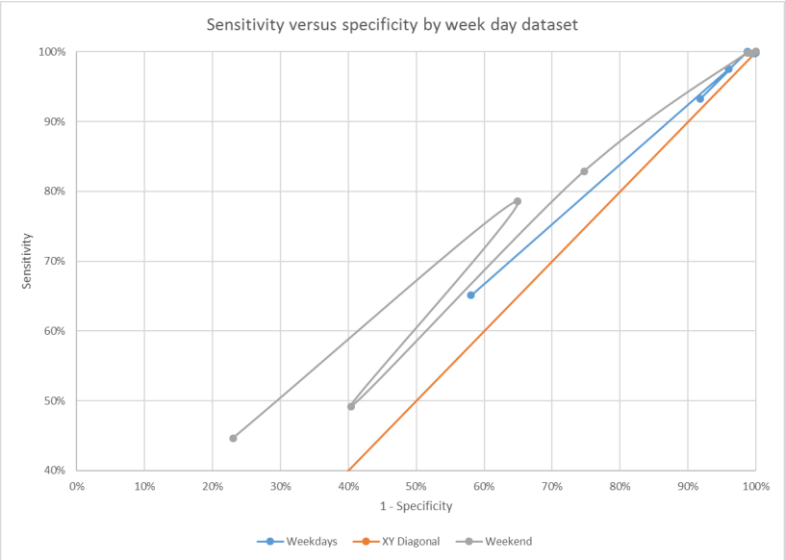

Figure 31 *- In this chart, we show the relation between the classification models sensitivity and specificity of the weekdays' datasets.*

In Figure 31, we can see that the optimal model for the weekday's dataset has approximately 65% for sensitivity and 40% for specificity, it corresponds to the N3_stp temporal window used to train and test the MISMO classifier with Resample as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 45% for sensitivity and approximately 75% for specificity, it corresponds to the N6_stp temporal window used to train and test the MDD classifier with SMOTE as a training data balancing method.

In sum, for this case, we can say that when we are maximizing accuracy the choice for optimal model is not unanimous and it leads to either maximizing sensitivity or specificity.

The explored results in accuracy maximization can be explored in Table 19**Table 17**, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
MISVM	RESAMPLE	Area Central	N1_stp	99,74%	0,21%	73,16%	23,08%	73,04%
MISVM	None	Area Central	N2_stp	100,00%	3,26%	59,03%	100,00%	59,59%
MIDD	None	Area Central	N3_stp	90,07%	13,03%	56,31%	51,32%	55,75%
MIDD	None	Area Central	N4_stp	90,08%	13,09%	56,40%	51,38%	55,83%
MISVM	SMOTE	Area Central	N5_stp	96,43%	5,58%	56,46%	55,19%	56,40%
MISVM	SMOTE	Area Central	N6_stp	94,70%	5,52%	67,89%	33,04%	66,02%
MISVM	RESAMPLE	Area Central	N7_stp	96,75%	2,84%	82,32%	15,74%	80,20%
MISVM	None	Cekh Clinic	N1_stp	71,64%	45,07%	66,62%	50,94%	61,14%
HMM	SMOTE	Cekh Clinic	N2_stp	57,96%	57,33%	52,07%	63,03%	57,61%
MISMO	None	Cekh Clinic	N3_stp	51,59%	64,79%	56,00%	60,65%	58,66%
QuickDDIterative	SMOTE	Cekh Clinic	N4_stp	93,13%	8,99%	56,84%	50,41%	56,34%
MISVM	RESAMPLE	Cekh Clinic	N5_stp	98,16%	7,39%	62,57%	71,77%	62,94%
MISMO	SMOTE	Cekh Clinic	N6_stp	62,26%	57,88%	61,53%	58,63%	60,15%
QuickDDIterative	RESAMPLE	Cekh Clinic	N7_stp	64,44%	68,21%	61,59%	70,81%	66,55%
MISMO	None	Hospital La Concepción	N1_stp	99,90%	0,16%	70,61%	38,46%	70,57%
MISMO	SMOTE	Hospital La Concepción	N2_stp	99,79%	0,29%	58,26%	50,00%	58,24%
MISMO	RESAMPLE	Hospital La Concepción	N3_stp	11,95%	96,75%	79,95%	50,35%	52,66%
MISVM	RESAMPLE	Hospital La Concepción	N4_stp	91,30%	33,33%	61,28%	76,82%	64,41%
HMM	SMOTE	Hospital La Concepción	N5_stp	68,89%	34,43%	55,11%	48,65%	53,00%
HMM	SMOTE	Hospital La Concepción	N6_stp	66,44%	37,76%	67,21%	36,95%	56,62%
MIDD	RESAMPLE	Hospital La Concepción	N7_stp	99,96%	0,17%	72,84%	62,50%	72,83%
QuickDDIterative	SMOTE	Plaza San Agustin	N1_stp	99,97%	0,02%	65,98%	25,00%	65,97%
MISMO	None	Plaza San Agustin	N2_stp	76,97%	37,06%	61,19%	55,51%	59,54%
MDD	SMOTE	Plaza San Agustin	N3_stp	53,59%	63,82%	63,20%	54,25%	58,33%
HMM	SMOTE	Plaza San Agustin	N4_stp	35,68%	74,39%	61,27%	50,46%	53,81%
QuickDDIterative	SMOTE	Plaza San Agustin	N5_stp	99,28%	0,95%	53,83%	53,02%	53,82%
MISVM	None	Plaza San Agustin	N6_stp	99,76%	0,90%	60,47%	71,43%	60,53%
MISVM	SMOTE	Plaza San Agustin	N7_stp	100,00%	0,41%	68,89%	100,00%	68,93%
MISVM	None	Weekdays	N1_stp	100,00%	1,29%	71,84%	100,00%	71,95%
MDD	SMOTE	Weekdays	N2_stp	93,28%	8,23%	56,66%	48,77%	56,08%
MISMO	RESAMPLE	Weekdays	N3_stp	65,11%	41,93%	55,68%	51,75%	54,18%
MIEMDD	None	Weekdays	N4_stp	99,87%	0,02%	54,20%	9,09%	54,16%
QuickDDIterative	SMOTE	Weekdays	N5_stp	99,73%	0,17%	56,02%	33,59%	55,97%
QuickDDIterative	SMOTE	Weekdays	N6_stp	97,53%	4,04%	64,44%	47,78%	63,93%
QuickDDIterative	SMOTE	Weekdays	N7_stp	99,83%	0,13%	72,87%	21,74%	72,79%
MIDD	RESAMPLE	Weekend	N1_stp	49,19%	59,56%	56,91%	51,92%	54,17%
HMM	SMOTE	Weekend	N2_stp	82,85%	25,27%	60,68%	51,41%	58,78%
MISVM	RESAMPLE	Weekend	N3_stp	99,70%	0,61%	61,50%	56,41%	61,47%
QuickDDIterative	None	Weekend	N4_stp	99,98%	0,03%	65,48%	50,00%	65,48%
MISVM	RESAMPLE	Weekend	N5_stp	99,88%	1,25%	60,78%	86,79%	60,93%
MDD	SMOTE	Weekend	N6_stp	44,64%	76,95%	70,04%	53,52%	59,28%
MIDD	None	Weekend	N7_stp	78,62%	35,11%	56,08%	60,91%	57,44%

Table 19 *- This table provides the classifiers, balancing types and respective results that maximize accuracy for each dataset and temporal window.*

5.3.3- Specificity maximization results

Regarding specificity maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in Figure 32.

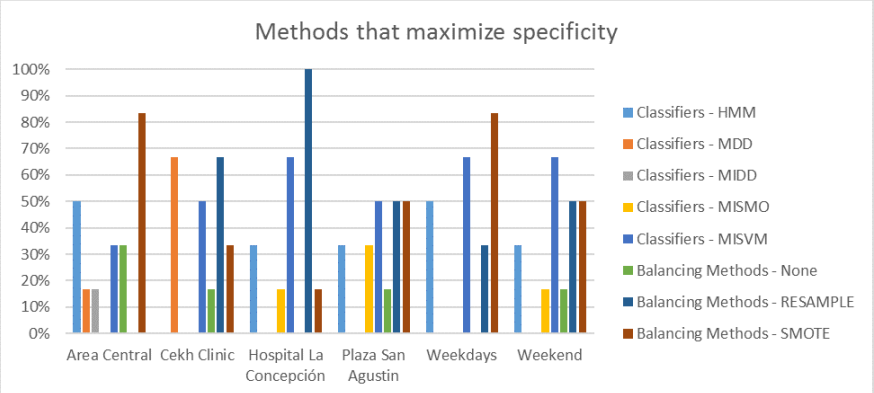

Figure 32 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize specificity.*

The analysis to be made to Figure 32 is complex, due to the diversity of methods that maximize specificity. We can see that MISVM has an approximate 45% overall classification use. Following MISVM in terms of use is HMM that is present in almost all datasets temporal window classification with an overall use of approximately 30%. The least used classification methods are MISMO and MDD with approximately 10% use and MIDD with approximately 5% use. In terms of balancing methods, Resample and SMOTE are the most used method with an overall use of approximately 45% each, followed by no type of data balancing with approximately 10% use.

Now let us pay attention to the results from the parks datasets, shown in Figure 33, of the application of the above classification methods and balancing methods.

Figure 33 *- This set of charts shows the classification results that maximize specificity, by each defined metric, by each temporal window and by each park dataset.*

In Figure 33, the maximization attempt of specificity has led to the implicit minimization of sensitivity. In terms of overall results, we have achieved a mean value of approximately 10% for sensitivity, for specificity the mean value is approximately 95% and in terms of precision the mean value is approximately 65%, as for NPV the mean value is at approximately at 40% and in terms of accuracy the mean value is at 45%.

We will now focus on Figure 34, that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in Figure 33**Figure 28** provides the optimal model.

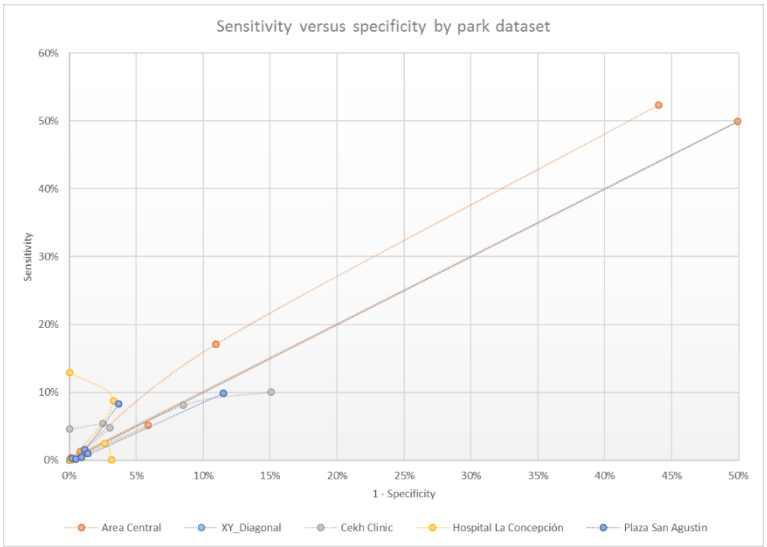

Figure 34 *- In this chart, we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing Figure 34, we can conclude that for “Area Central” the optimal model has approximately 50% sensitivity and 55% specificity, it corresponds to the N1_stp temporal window used to train and test the HMM classifier with SMOTE as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 10% sensitivity and 90% specificity, it corresponds to the N3_stp temporal window used to train and test the HMM classifier with Resample as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 15% sensitivity and 100% specificity, it corresponds to the N2_stp temporal window used to train and test the MISVM classifier with RESAMPLE as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 10% sensitivity and 85% specificity, it corresponds to the N7_stp temporal window used to train and test the MDD classifier with SMOTE as a training data balancing method.

We will now explore the discussed the results as well as the optimal model for the weekdays’ datasets.

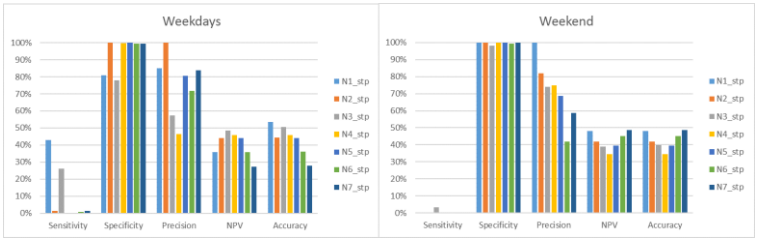

Figure 35 *- This set of charts shows the classification results, that maximize specificity, by each defined metric, by each temporal window and by each weekday type dataset.*

In Figure 35, we explore the week days' datasets, also with the goal of maximizing specificity, we can observe that the behavior is similar than the one present in the parks datasets i.e. the implicit minimization of sensitivity, which leads to a mean value of approximately 5% for sensitivity, a mean value of approximately 95% for specificity, for precision the mean value is approximately 75%, the mean value for NPV is approximately 40% and for accuracy has approximately 45% mean value.

Compared to the parks datasets results mean values, the results are approximately the same, with the exception of sensitivity and NPV precision that show a decrease of approximately 5% and an increase of approximately 10%, respectively.

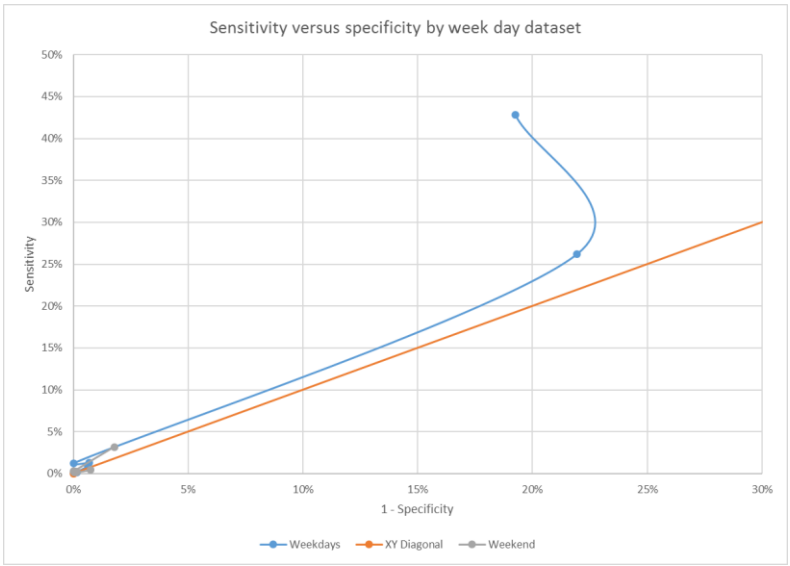

Figure 36 *- In this chart, we show the relation between the classification models sensitivity and specificity of the weekdays' datasets.*

In Figure 36 we can see that the optimal model for the weekdays dataset has approximately 45% for sensitivity and 80% for specificity, it corresponds to the N1_stp temporal window used to train and test the MISVM classifier with SMOTE as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 5% for sensitivity and approximately 100% for specificity, it corresponds to the N3_stp temporal window used to train and test the MISMO classifier with SMOTE as a training data balancing method.

In sum, for this case, we can say that when we are maximizing specificity the choice for optimal model is not unanimous and it leads to the minimization of sensitivity.

The explored results in specificity maximization can be explored in Table 20**Table 17**, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
HMM	SMOTE	Area Central	N1_stp	52,37%	55,99%	76,45%	30,11%	53,34%
HMM	SMOTE	Area Central	N2_stp	49,95%	50,06%	58,34%	41,67%	50,00%
MISVM	SMOTE	Area Central	N3_stp	5,14%	94,13%	52,14%	44,36%	44,79%
MISVM	SMOTE	Area Central	N4_stp	0,35%	99,88%	78,85%	44,54%	44,62%
MDD	SMOTE	Area Central	N5_stp	1,27%	99,19%	66,67%	44,17%	44,41%
MIDD	None	Area Central	N6_stp	1,04%	98,88%	66,23%	32,14%	32,50%
HMM	None	Area Central	N7_stp	17,08%	89,07%	88,03%	18,58%	29,70%
MISVM	SMOTE	Cekh Clinic	N1_stp	0,03%	100,00%	100,00%	39,53%	39,54%
MISVM	RESAMPLE	Cekh Clinic	N2_stp	0,04%	100,00%	100,00%	55,57%	55,58%
MISVM	None	Cekh Clinic	N3_stp	4,61%	100,00%	100,00%	54,69%	55,66%
MDD	RESAMPLE	Cekh Clinic	N4_stp	5,41%	97,50%	73,54%	44,47%	45,68%
MDD	RESAMPLE	Cekh Clinic	N5_stp	4,84%	96,97%	71,60%	39,25%	40,59%
MDD	RESAMPLE	Cekh Clinic	N6_stp	8,12%	91,48%	50,78%	47,92%	48,16%
MDD	SMOTE	Cekh Clinic	N7_stp	10,10%	84,91%	34,63%	54,43%	51,88%
MISVM	SMOTE	Hospital La Concepción	N1_stp	0,08%	96,83%	5,61%	28,76%	28,53%
MISVM	RESAMPLE	Hospital La Concepción	N2_stp	12,92%	100,00%	100,00%	45,14%	49,27%
MISVM	RESAMPLE	Hospital La Concepción	N3_stp	0,05%	99,90%	37,50%	48,01%	48,00%
MISMO	RESAMPLE	Hospital La Concepción	N4_stp	0,03%	99,88%	25,00%	46,36%	46,34%
HMM	RESAMPLE	Hospital La Concepción	N5_stp	2,48%	97,36%	52,35%	46,07%	46,23%
MISVM	RESAMPLE	Hospital La Concepción	N6_stp	8,78%	96,71%	83,69%	35,56%	38,88%
HMM	RESAMPLE	Hospital La Concepción	N7_stp	0,13%	99,86%	71,43%	27,19%	27,24%
MISMO	SMOTE	Plaza San Agustin	N1_stp	8,29%	96,32%	81,38%	35,12%	38,23%
MISVM	None	Plaza San Agustin	N2_stp	1,03%	98,63%	49,39%	43,58%	43,65%
HMM	RESAMPLE	Plaza San Agustin	N3_stp	9,86%	88,50%	49,92%	45,78%	46,22%
MISVM	RESAMPLE	Plaza San Agustin	N4_stp	0,19%	99,51%	30,43%	46,92%	46,86%
MISVM	RESAMPLE	Plaza San Agustin	N5_stp	0,27%	99,81%	62,50%	46,24%	46,28%
HMM	SMOTE	Plaza San Agustin	N6_stp	1,56%	98,84%	67,25%	39,76%	40,14%
MISMO	SMOTE	Plaza San Agustin	N7_stp	0,41%	99,11%	50,00%	31,10%	31,20%
MISVM	SMOTE	Weekdays	N1_stp	42,86%	80,76%	84,87%	35,95%	53,63%
MISVM	SMOTE	Weekdays	N2_stp	1,26%	100,00%	100,00%	44,03%	44,43%
MISVM	SMOTE	Weekdays	N3_stp	26,20%	78,07%	57,24%	48,56%	50,66%
HMM	RESAMPLE	Weekdays	N4_stp	0,12%	99,83%	46,34%	45,72%	45,72%
HMM	SMOTE	Weekdays	N5_stp	0,08%	99,98%	80,65%	43,93%	43,95%
HMM	RESAMPLE	Weekdays	N6_stp	0,89%	99,38%	71,81%	35,97%	36,25%
MISVM	SMOTE	Weekdays	N7_stp	1,33%	99,30%	83,73%	27,24%	27,90%
MISVM	SMOTE	Weekend	N1_stp	0,27%	100,00%	100,00%	48,01%	48,08%
MISVM	RESAMPLE	Weekend	N2_stp	0,17%	99,95%	81,82%	41,79%	41,84%
MISMO	SMOTE	Weekend	N3_stp	3,19%	98,21%	73,98%	38,92%	39,85%
HMM	RESAMPLE	Weekend	N4_stp	0,15%	99,91%	75,00%	34,54%	34,59%
HMM	SMOTE	Weekend	N5_stp	0,20%	99,86%	68,75%	39,50%	39,56%
MISVM	RESAMPLE	Weekend	N6_stp	0,45%	99,24%	41,82%	45,23%	45,21%
MISVM	None	Weekend	N7_stp	0,21%	99,85%	58,82%	48,70%	48,72%

Table 20 - This table provides the classifiers, balancing types and respective results that maximize specificity for each dataset and temporal window.

5.4- Critical analysis

In this section, we will discuss the tools' overall results, having in mind the information discovered from the previous analysis.

Considering that Instance Classification is implicitly contained in Time-Window based Classification, it will not be referred directly in this section.

In Time-Window based Classification, the methodology average metric result range will fluctuate approximately between 70% and 90%, so we can consider this as a stable methodology considering that the methodology average metric result range for Multi-Instance Classification can fluctuate between 1% and 100%.

In terms of optimal methodology, in Time-Window based Classification, Random Forest is the unanimous method to apply in any type of metric maximization and regarding the data balancing methodology SMOTE and Resample were the choices, being SMOTE the most chosen method. Regarding Multi-Instance Classification, optimal methodology, MISVM was the most chosen method, as for the data balancing SMOTE and Resample were the choices, being Resample the most chosen method.

In this concrete case, we can observe that at least in terms of accuracy that Multi Instance classification performs badly when compared with Time-Window based Classification. Due to the original data sequential capabilities, we expected better results in Multi Instance classification but the data lacked the necessary richness to make these performances good. We can argue that in terms of specificity and sensitivity, we had outstanding results, but the fact is that to have an excellent specificity rate it implies having a very poor sensitivity rate and vice versa, which means that our multi instance classifiers would classify all sequences either as true or as false.

In what concerns our tool purpose, it was achieved as we now know, depending on the explored metric, which are the methods that will perform better with a specific dataset.

6- Conclusion

In this work, we have proposed to address the need for a data analysis tool that would help in decision making, in the car parking business scope.

Before understanding what would be the best form of obtaining the best data analysis methodologies, to help decision-making, we analyzed the source data model in order to understand how to potentiate its use. Then we defined and applied a set of methodologies in order to potentiate the data analysis and finally we identified a set of metrics that would help us evaluate the proposed methodologies performance.

Following this analysis and experimentation phase, we began to analyze what would be important, in terms of requirements, to build a tool capable of performing the discussed actions. After performing the requirement analysis, we started to design the system architecture, where we defined, with more technical precision, each of the tools components.

After developing the tool, we tested it with the data associated with the business that we are exploring. In terms of results, we identified the best set of methodologies by every metric we have chosen and concluded that within the three configured types of classification (instance, time-window based and multi-instance oriented classification) only one proved to provide the most reliable and stable set of methodologies i.e. time-window based oriented classification.

As referred in the beginning of this work, there are many data analysis tools, but these all have a lack of data manipulation capacity, and so what we have proposed is a tool capable of performing this data manipulation. We've demonstrated, in this work, that with this tool we can take a data warehouse dimensional model, denormalize its data, transform the same set of data into three sets of data that provide different perspectives of looking at the data and use it with different sets of data analysis methodologies. As an example of this tools manipulation capabilities, we can take the time-window based structure which is a temporal window whose events are aggregated by customer and aggregate it by product type, changing our perspective to a more product oriented and not explicitly customer related. This data manipulation capability has originated from the high level of configurability the tool needs, to work properly, which may be considered a big limitation because it increases the level of complexity of simple tasks related with the tools use.

In sum, we can state that we have created a data analysis tool capable of manipulating dataset in order to explore its temporality, and to use its manipulation results with data analysis methodologies in order to obtain the best set of methodologies to apply in these manipulation results.

7- References

[1] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

[2] John, George H., and Pat Langley. "Estimating continuous distributions in Bayesian classifiers." Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1995.

[3] Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." icml. Vol. 96. 1996.

[4] Breiman, Leo. "Random Forests." Machine learning 45.1 (2001): 5-32.

[5] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

[6] Andrews, Stuart, Ioannis Tsochantaridis, and Thomas Hofmann. "Support vector machines for multiple-instance learning." Advances in neural information processing systems (2003): 577-584.

[7] Maron, Oded, and Tomás Lozano-Pérez. "A framework for multiple-instance learning." Advances in neural information processing systems (1998): 570-576.

[8] Zhang, Qi, and Sally A. Goldman. "EM-DD: An improved multiple-instance learning technique." NIPS. Vol. 1. 2001.

[9] Foulds, James R., and Eibe Frank. "Speeding up and boosting diverse density learning." International Conference on Discovery Science. Springer Berlin Heidelberg, 2010.

[10] Keerthi, S. Sathiya, et al. "Improvements to Platt's SMO algorithm for SVM classifier design." Neural computation 13.3 (2001): 637-649.

[11] Hall, Mark, et al. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11.1 (2009): 10-18.

8- Appendix

8.1- Time-Window based Classification Results

8.1.1- Sensitivity maximization results

Regarding sensitivity maximization, the classification approaches and data balancing approaches chosen for the configured datasets are shown in .

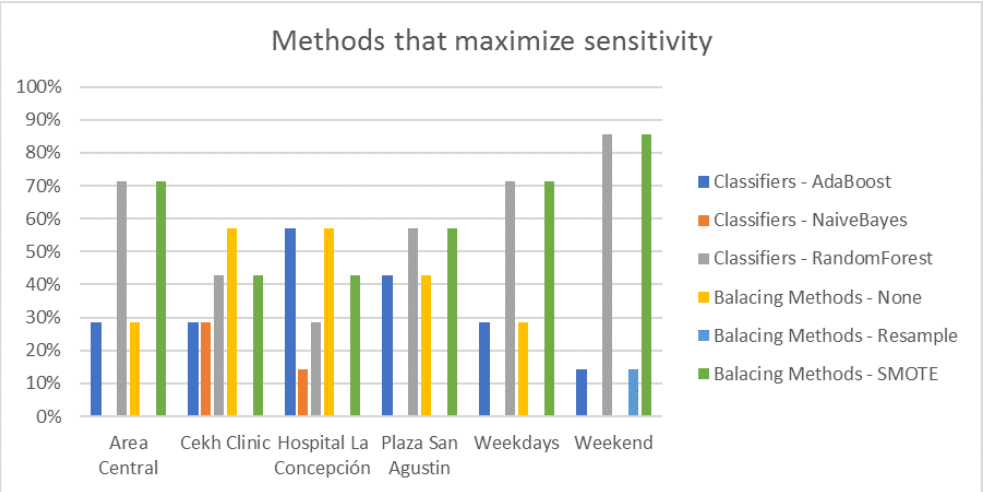

Figure 37 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize sensitivity.*

Starting with “Area Central” datasets, we can observe that even though approximately 70% of the datasets maximize sensitivity by using Random Forest and SMOTE, the other 30% do it by using AdaBoost with no type of data balancing.

In “Cekh Clinic” in terms of classifiers almost all classifiers are used, Naïve Bayes and AdaBoost with approximately 30% use and Random Forest with almost 40% use. In terms of data balancing there’s a small gap between the use of SMOTE and the use of no type of data balancing.

In “Hospital La Concepción” in terms of data balancing, we can see that the behavior is the same as in “Cekh Clinic”. Concerning the classifier use, AdaBoost is the most used classifier, with almost 60% use, leaving Naïve Bayes with approximately 15% use and Random Forest with approximately 30% use.

In “Plaza San Agustin”, the methods most used are Random Forest and SMOTE with approximately 60% use, opposed to the 40% use of Ada Boost and no type of data balancing.

In the “Weekdays” dataset classification, the behavior is the same as in “Cekh Clinic”. As for the “Weekend” dataset classification, Random Forest and SMOTE are the most used methods with approximately 85% use, leaving the other 15% use for AdaBoost and no data balancing.

Finally, in the end of the analysis of , we can state that, there is not one obvious and transversal set of methodologies to apply, in order to maximize sensitivity.

We will now focus on the results from the parks datasets, shown in , of the classification and balancing methods described above, in .

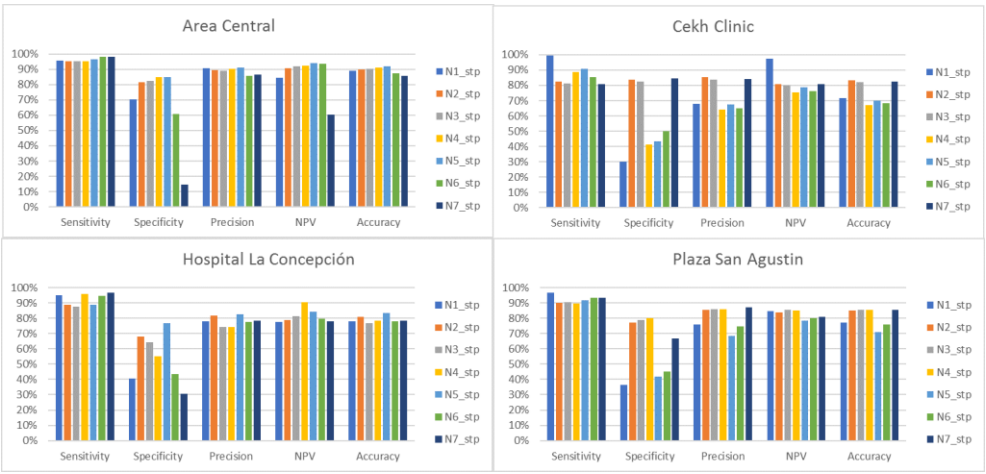

Figure 38 *- This set of charts shows the classification results, which maximize sensitivity, by each defined metric, by each temporal window and by each park dataset.*

As we can see in , with the already described maximization attempt of sensitivity we have achieved a mean value for accuracy of approximately 80%. Sensitivity results mean value is at approximately 90%, in terms of precision the mean value is approximately 80%, in terms of NPV the mean value is approximately 85% and concerning specificity the mean value is the lowest, when comparing with other discussed metrics, with 60% value.

We will now focus on , that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in provides the optimal model.

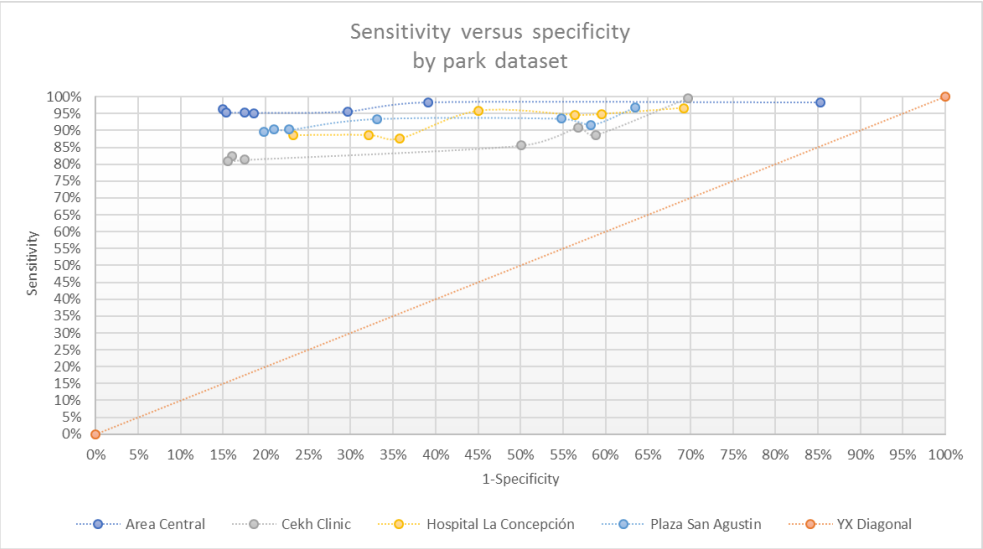

Figure 39 *- In this chart, we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing Figure 39 we can conclude that for “Area Central” the optimal model has approximately 95% sensitivity and 85% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 90% sensitivity and 80% specificity, it corresponds to the N4_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 90% sensitivity and 75% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 80% sensitivity and 85% specificity, it corresponds to the N2_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

It is unanimous that Random Forest classifier with SMOTE as balancing method has provided the optimal models for the parks datasets. We will now see if the same can be stated for the weekdays and weekends datasets, even though with the information showed in points to the same outcome.

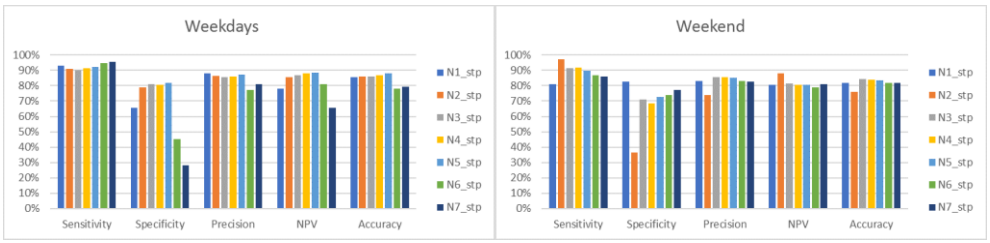

Figure 40 *- This set of charts shows the classification results, that maximize sensitivity, by each defined metric, by each temporal window and by each weekday type dataset.*

In , we explore the weekdays' datasets, also with the goal of maximizing sensitivity, which leads to a mean value of approximately 90% for sensitivity, a mean value of approximately 65% for specificity, a mean value of approximately 85% for precision, approximately 80% mean value for NPV and approximately 85% of mean value for accuracy.

Compared to the parks datasets results mean values, for sensitivity maximization, the weekdays' dataset results mean values have shown an increase of approximately 5% in all metrics with the exception of sensitivity and NPV, as sensitivity maintains approximately the same mean value and NPV has a decrease of 5% in terms of mean value.

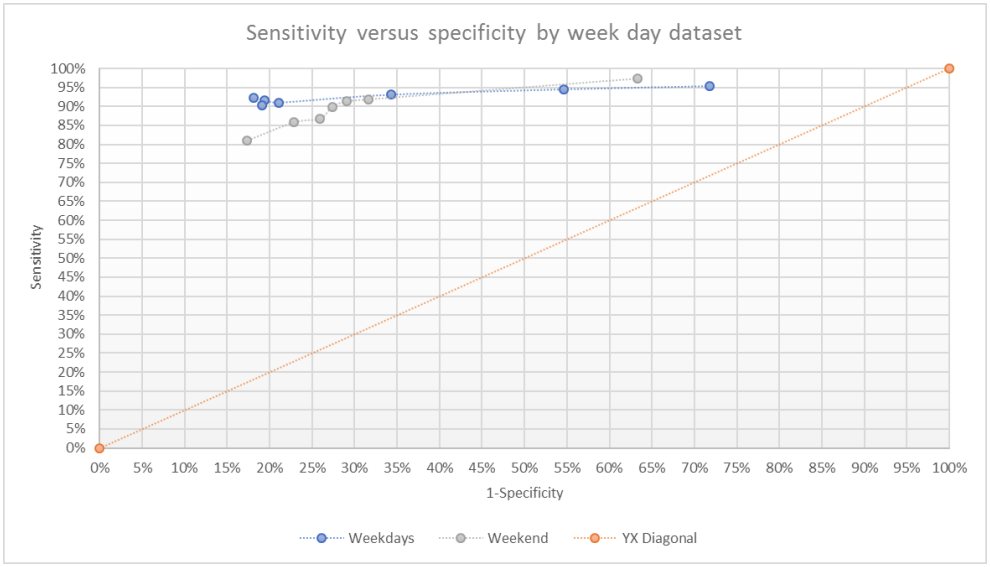

Figure 41 *- In this chart, we show the relation between the classification models sensitivity and specificity of the weekdays' datasets.*

In , we can see that the optimal model for the weekdays dataset has approximately 90% for sensitivity and 80% for specificity, it corresponds to the N5_stp

temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 80% for sensitivity and for specificity, it corresponds to the N1_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

In sum, we can say that when we are maximizing sensitivity even though we have various classifier methods and various data balancing methods the optimal methods reside unanimously in the use of Random Forest classifier and SMOTE data balancing method.

In the we can explore more closely these results.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
RandomForest	SMOTE	Area Central	N1_stp	95,55%	70,36%	90,51%	84,23%	89,19%
RandomForest	SMOTE	Area Central	N2_stp	95,14%	81,37%	89,61%	90,83%	90,01%
RandomForest	SMOTE	Area Central	N3_stp	95,20%	82,45%	89,14%	91,91%	90,13%
RandomForest	SMOTE	Area Central	N4_stp	95,40%	84,67%	90,37%	92,43%	91,12%
RandomForest	SMOTE	Area Central	N5_stp	96,43%	84,99%	90,98%	93,81%	91,98%
AdaBoost	None	Area Central	N6_stp	98,29%	60,87%	85,80%	93,66%	87,30%
AdaBoost	None	Area Central	N7_stp	98,29%	14,67%	86,66%	60,33%	85,69%
AdaBoost	None	Cekh Clinic	N1_stp	99,49%	30,26%	67,88%	97,55%	71,59%
RandomForest	SMOTE	Cekh Clinic	N2_stp	82,44%	83,90%	85,43%	80,67%	83,12%
RandomForest	SMOTE	Cekh Clinic	N3_stp	81,36%	82,47%	83,74%	79,94%	81,88%
NaiveBayes	None	Cekh Clinic	N4_stp	88,71%	41,14%	63,97%	75,58%	66,87%
AdaBoost	None	Cekh Clinic	N5_stp	90,95%	43,18%	67,44%	78,67%	70,12%
NaiveBayes	None	Cekh Clinic	N6_stp	85,55%	49,95%	65,07%	76,02%	68,51%
RandomForest	SMOTE	Cekh Clinic	N7_stp	80,80%	84,41%	84,27%	80,96%	82,57%
AdaBoost	None	Hospital La Concepción	N1_stp	94,84%	40,41%	78,09%	77,76%	78,04%
RandomForest	SMOTE	Hospital La Concepción	N2_stp	88,66%	67,89%	81,79%	78,63%	80,75%
AdaBoost	SMOTE	Hospital La Concepción	N3_stp	87,65%	64,18%	74,32%	81,46%	76,90%
NaiveBayes	None	Hospital La Concepción	N4_stp	95,79%	54,96%	74,44%	90,50%	78,56%
RandomForest	SMOTE	Hospital La Concepción	N5_stp	88,66%	76,74%	82,64%	84,42%	83,36%
AdaBoost	None	Hospital La Concepción	N6_stp	94,60%	43,65%	77,77%	79,49%	78,08%
AdaBoost	None	Hospital La Concepción	N7_stp	96,63%	30,76%	78,32%	77,89%	78,27%
AdaBoost	None	Plaza San Agustin	N1_stp	96,78%	36,56%	76,02%	84,53%	77,22%
RandomForest	SMOTE	Plaza San Agustin	N2_stp	90,23%	77,22%	85,64%	84,01%	85,04%
RandomForest	SMOTE	Plaza San Agustin	N3_stp	90,33%	79,01%	85,74%	85,40%	85,61%
RandomForest	SMOTE	Plaza San Agustin	N4_stp	89,71%	80,18%	86,03%	85,13%	85,67%
AdaBoost	None	Plaza San Agustin	N5_stp	91,66%	41,69%	68,58%	78,27%	70,74%
AdaBoost	None	Plaza San Agustin	N6_stp	93,49%	45,23%	74,69%	80,08%	75,80%
RandomForest	SMOTE	Plaza San Agustin	N7_stp	93,44%	66,88%	87,02%	81,08%	85,57%
RandomForest	SMOTE	Weekdays	N1_stp	93,09%	65,69%	87,86%	78,09%	85,61%
RandomForest	SMOTE	Weekdays	N2_stp	90,96%	78,91%	86,22%	85,75%	86,04%
RandomForest	SMOTE	Weekdays	N3_stp	90,20%	80,92%	85,72%	86,67%	86,11%
RandomForest	SMOTE	Weekdays	N4_stp	91,52%	80,61%	86,08%	87,88%	86,79%
RandomForest	SMOTE	Weekdays	N5_stp	92,34%	81,86%	87,42%	88,68%	87,91%
AdaBoost	None	Weekdays	N6_stp	94,52%	45,42%	77,26%	80,85%	77,94%
AdaBoost	None	Weekdays	N7_stp	95,40%	28,26%	80,94%	65,80%	79,39%
RandomForest	SMOTE	Weekend	N1_stp	81,00%	82,64%	83,04%	80,56%	81,80%
AdaBoost	Resample	Weekend	N2_stp	97,30%	36,70%	74,05%	87,99%	76,08%
RandomForest	SMOTE	Weekend	N3_stp	91,38%	70,88%	85,39%	81,53%	84,22%
RandomForest	SMOTE	Weekend	N4_stp	91,84%	68,34%	85,40%	80,59%	84,05%
RandomForest	SMOTE	Weekend	N5_stp	89,85%	72,63%	85,00%	80,57%	83,53%
RandomForest	SMOTE	Weekend	N6_stp	86,74%	74,05%	83,26%	78,95%	81,64%
RandomForest	SMOTE	Weekend	N7_stp	85,79%	77,11%	82,82%	80,83%	81,99%

Table 21 *- This table provides the classifiers, balancing types and respective results that maximize sensitivity, for each dataset and temporal window.*

8.1.2- Precision maximization results

In what concerns precision maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in .

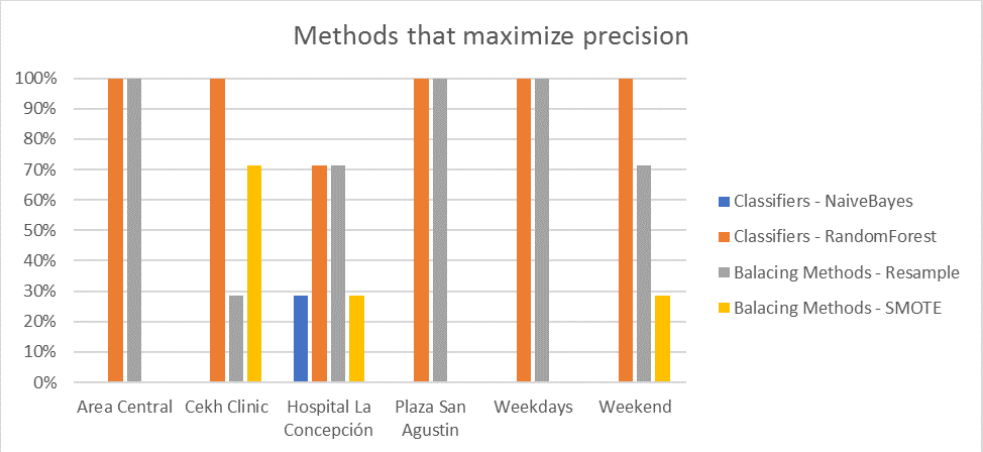

Figure 42 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize precision.*

Starting with “Area Central” datasets we can observe that all datasets maximize precision by using Random Forest and Resample.

In “Cekh Clinic”, in terms of classifiers, Random Forest is the prevailing choice being the dataset balancing divided between SMOTE with approximately 70% and Resample with approximately 30% use.

In “Hospital La Concepción” we can see that Random Forest is used in approximately 70% of the classifications being Naïve Bayes responsible for the other approximate 30%. In terms of data balancing for “Hospital La Concepción”, Resample is the most used method, with 70% use, and Naïve Bayes the least used method with 30% use.

In “Plaza San Agustin” and in the Weekdays datasets, the behavior is the same as in “Area Central”, being Random Forest and Resample the choices for classification and data balancing, respectively.

In the weekends classification, Random Forest is the chosen classifier as for data balancing method the choices are Resample, with approximately 70% use, and SMOTE, with approximately 30% use.

From the observation and description of , we can state that, in terms of precision, the set of methodologies to apply, would be Resample and Random Forest due to their predominance, in terms of use.

Now we'll pay attention to the results from the parks datasets, shown in , of the application of the above classification methods and balancing methods.

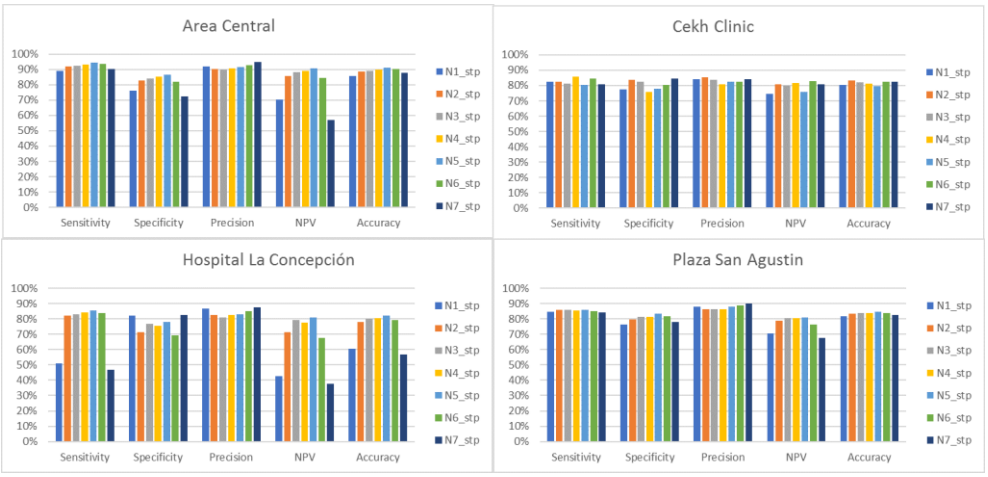

Figure 43 *- This set of charts shows the classification results, that maximize precision, by each defined metric, by each temporal window and by each park dataset.*

In , with the already described maximization attempt of precision, we've achieved a mean value of approximately 80% for accuracy and specificity, for sensitivity the mean value is approximately 85%, in terms of precision, which is the metric we are maximizing, the mean value is at approximately at 85% and in terms of NPV the mean value is at approximately at 75%.

We'll now focus on , that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows, in Figure 43, provides the optimal model.

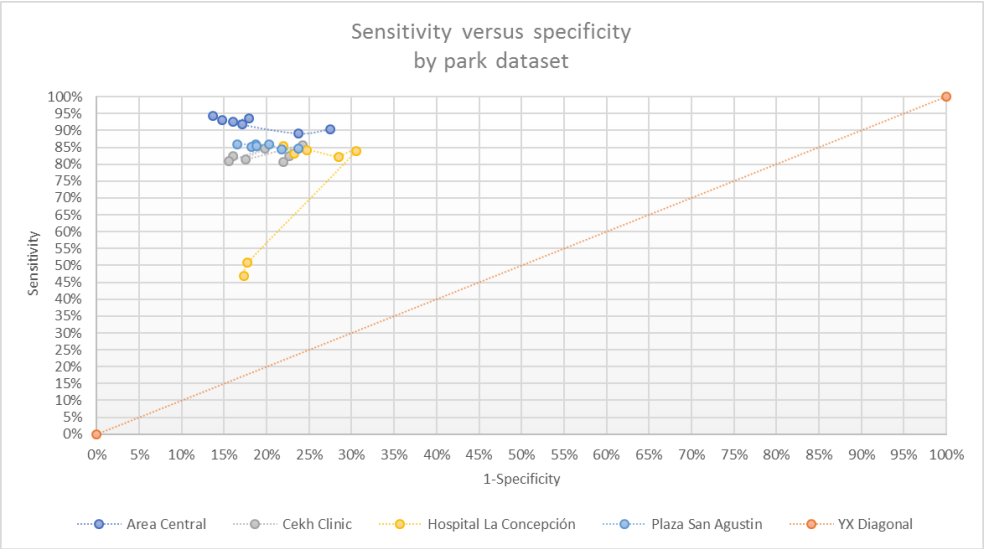

Figure 44 *- In this chart we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing we can conclude that for “Area Central” the optimal model has approximately 95% sensitivity and 85% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 85% for sensitivity and specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 85% sensitivity and 80% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 80% sensitivity and 85% specificity, it corresponds to the N2_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

In terms of classifiers, and as we predicted earlier, Random Forest was the unanimous choice as the optimal model for the parks datasets classification, and even though it was also expected that Resample would be the unanimous balancing method chosen, it was the most used in the optimal models.

For the weekdays and weekends datasets, by analyzing 2 we already know Random Forest will be the unanimous balancing method, and that at least for the

weekdays datasets Resample will also be present on the optimal model for the specified datasets.

The only aspect left to know is what will be the temporal window for the optimal model and if the weekday's results will be as good as the parks datasets results and what will be the optimal model data balancing method for the weekend datasets.

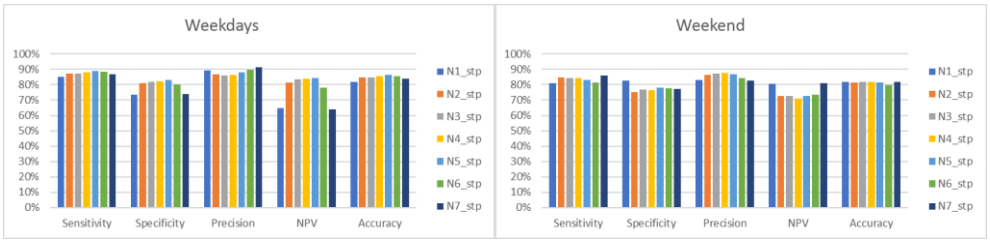

Figure 45 *- This set of charts shows the classification results, that maximize precision, by each defined metric, by each temporal window and by each week day type dataset.*

In , we explore the week days' datasets, also with the goal of maximizing precision, which leads to a mean value of approximately 85% for accuracy, sensitivity and precision, a mean value of approximately 80% for specificity and approximately 75% mean value for NPV.

Compared to the parks datasets results mean values, for precision maximization, the week days' dataset results mean values maintain approximately the same value for all metrics with the exception of accuracy that shows an increase of approximately 5% in terms of mean value.

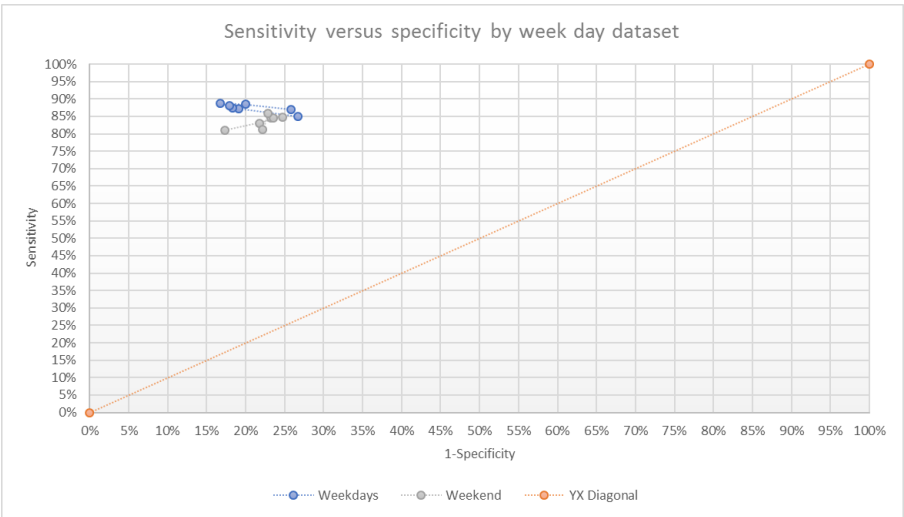

Figure 46 *- In this chart we show the relation between the classification models sensitivity and specificity of the week days' datasets.*

In , we can see that the optimal model for the weekdays dataset has approximately 90% for sensitivity and 85% for specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with Resample as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 80% for sensitivity and approximately 85% for specificity, it corresponds to the N1_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

In sum, for this case, we can say that when we're maximizing precision the optimal model will reside on the choice of Random Forest and as for the data balancing method even though it shows that Resample was the most common choice, SMOTE as also to be considered.

The explored results in precision maximization can be explored in Table 22, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
RandomForest	Resample	Area Central	N1_stp	89,08%	76,26%	91,74%	70,23%	85,84%
RandomForest	Resample	Area Central	N2_stp	91,82%	82,86%	90,05%	85,72%	88,49%
RandomForest	Resample	Area Central	N3_stp	92,49%	83,95%	89,71%	88,08%	89,09%
RandomForest	Resample	Area Central	N4_stp	93,15%	85,19%	90,46%	89,19%	89,98%
RandomForest	Resample	Area Central	N5_stp	94,30%	86,30%	91,53%	90,61%	91,19%
RandomForest	Resample	Area Central	N6_stp	93,70%	82,10%	92,64%	84,41%	90,29%
RandomForest	Resample	Area Central	N7_stp	90,29%	72,49%	94,87%	56,97%	87,61%
RandomForest	Resample	Cekh Clinic	N1_stp	82,38%	77,30%	84,32%	74,76%	80,33%
RandomForest	SMOTE	Cekh Clinic	N2_stp	82,44%	83,90%	85,43%	80,67%	83,12%
RandomForest	SMOTE	Cekh Clinic	N3_stp	81,36%	82,47%	83,74%	79,94%	81,88%
RandomForest	SMOTE	Cekh Clinic	N4_stp	85,71%	75,76%	80,64%	81,82%	81,14%
RandomForest	Resample	Cekh Clinic	N5_stp	80,55%	78,02%	82,58%	75,62%	79,45%
RandomForest	SMOTE	Cekh Clinic	N6_stp	84,71%	80,18%	82,32%	82,79%	82,54%
RandomForest	SMOTE	Cekh Clinic	N7_stp	80,80%	84,41%	84,27%	80,96%	82,57%
NaiveBayes	SMOTE	Hospital La Concepción	N1_stp	50,79%	82,26%	86,51%	42,74%	60,51%
RandomForest	Resample	Hospital La Concepción	N2_stp	82,21%	71,54%	82,46%	71,20%	78,15%
RandomForest	Resample	Hospital La Concepción	N3_stp	83,02%	76,72%	80,83%	79,25%	80,13%
RandomForest	Resample	Hospital La Concepción	N4_stp	84,21%	75,29%	82,36%	77,69%	80,45%
RandomForest	Resample	Hospital La Concepción	N5_stp	85,37%	77,99%	82,88%	81,03%	82,09%
RandomForest	Resample	Hospital La Concepción	N6_stp	83,89%	69,43%	85,12%	67,40%	79,20%
NaiveBayes	SMOTE	Hospital La Concepción	N7_stp	46,84%	82,70%	87,51%	37,54%	56,84%
RandomForest	Resample	Plaza San Agustin	N1_stp	84,58%	76,24%	88,09%	70,41%	81,87%
RandomForest	Resample	Plaza San Agustin	N2_stp	85,86%	79,70%	86,43%	78,92%	83,40%
RandomForest	Resample	Plaza San Agustin	N3_stp	85,80%	81,26%	86,48%	80,38%	83,91%
RandomForest	Resample	Plaza San Agustin	N4_stp	85,43%	81,22%	86,09%	80,38%	83,65%
RandomForest	Resample	Plaza San Agustin	N5_stp	85,78%	83,44%	87,79%	80,87%	84,80%
RandomForest	Resample	Plaza San Agustin	N6_stp	85,25%	81,77%	88,99%	76,23%	83,97%
RandomForest	Resample	Plaza San Agustin	N7_stp	84,38%	78,19%	90,19%	67,80%	82,55%
RandomForest	Resample	Weekdays	N1_stp	84,99%	73,34%	89,47%	64,70%	81,81%
RandomForest	Resample	Weekdays	N2_stp	87,27%	80,90%	86,89%	81,41%	84,67%
RandomForest	Resample	Weekdays	N3_stp	87,39%	81,65%	85,81%	83,61%	84,86%
RandomForest	Resample	Weekdays	N4_stp	88,15%	82,11%	86,59%	84,09%	85,53%
RandomForest	Resample	Weekdays	N5_stp	88,78%	83,27%	87,87%	84,46%	86,45%
RandomForest	Resample	Weekdays	N6_stp	88,46%	79,97%	89,65%	77,93%	85,59%
RandomForest	Resample	Weekdays	N7_stp	87,01%	74,14%	91,49%	64,11%	83,94%
RandomForest	SMOTE	Weekend	N1_stp	81,00%	82,64%	83,04%	80,56%	81,80%
RandomForest	Resample	Weekend	N2_stp	84,70%	75,22%	86,39%	72,59%	81,38%
RandomForest	Resample	Weekend	N3_stp	84,47%	76,70%	87,11%	72,61%	81,76%
RandomForest	Resample	Weekend	N4_stp	84,43%	76,46%	87,85%	70,89%	81,79%
RandomForest	Resample	Weekend	N5_stp	83,02%	78,20%	86,79%	72,74%	81,25%
RandomForest	Resample	Weekend	N6_stp	81,21%	77,83%	84,49%	73,57%	79,85%
RandomForest	SMOTE	Weekend	N7_stp	85,79%	77,11%	82,82%	80,83%	81,99%

Table 22 *- This table provides the classifiers, balancing types and respective results that maximize precision for each dataset and temporal window.*

8.1.3- NPV maximization results

In what concerns NPV maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in .

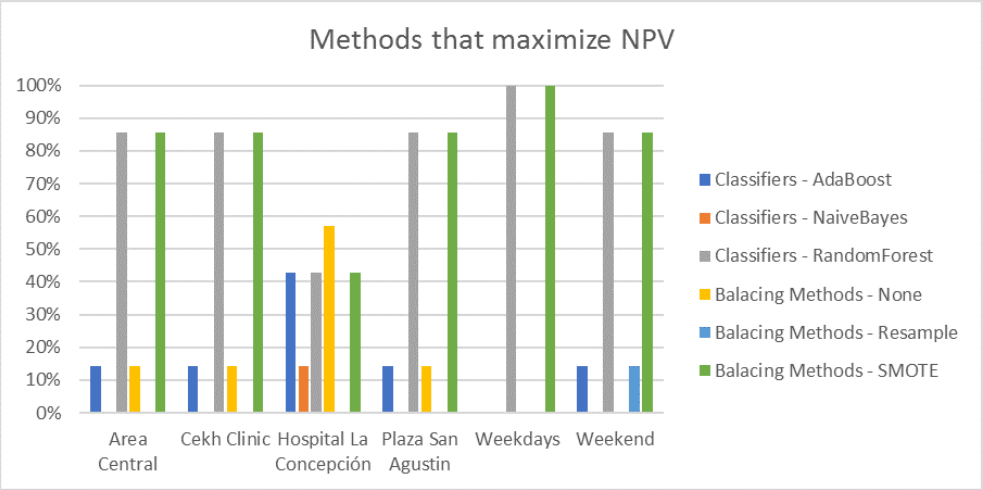

Figure 47 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize NPV.*

Starting with “Area Central” datasets, we can observe that NPV is maximized by using Random Forest and Resample in approximately 85% of the classifications and by using AdaBoost and no type of data balancing in approximately 15% of the classifications.

In “Cekh Clinic” and “Plaza San Agustin”, the behavior is the same as in “Area Central”.

In “Hospital La Concepción”, all classification methods are used, being AdaBoost and Random Forest the most used, with approximately 40% use each, and Naïve Bayes with approximately 15% use. In terms of data balancing for “Hospital La Concepción”, SMOTE is used approximately in 40% of the classifications and for the other 60% no method of balancing is used.

For the Weekdays datasets, the clear choice is Random Forest for classification and SMOTE for data balancing.

In the weekends classification, Random Forest is used approximately in 85% of the classifications and AdaBoost in the other 15%, as for data balancing method the choices are Resample, with approximately 15% use, and SMOTE, with approximately 85% use.

From the observation and description of , we can state that, in terms of NPV, the set of methodologies to apply is quite varied, so we will explore the next following results in order to achieve a conclusion.

Now we'll pay attention to the results from the parks datasets, shown in , of the application of the above classification methods and balancing methods.

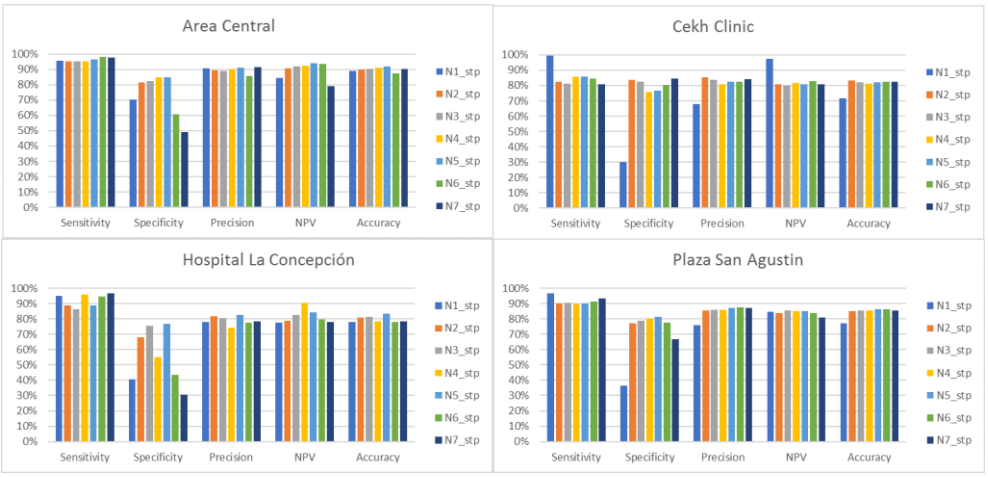

Figure 48 *- This set of charts shows the classification results, that maximize NPV, by each defined metric, by each temporal window and by each park dataset*

.

In , with the already described maximization attempt of NPV, we've achieved a mean value of approximately 85% for accuracy, precision and NPV, for sensitivity the mean value is approximately 90% and concerning specificity the mean value is approximately 70%.

We'll now focus on , that will show the relation between sensitivity and specificity, in order to understand which temporal window, in , provides the optimal model.

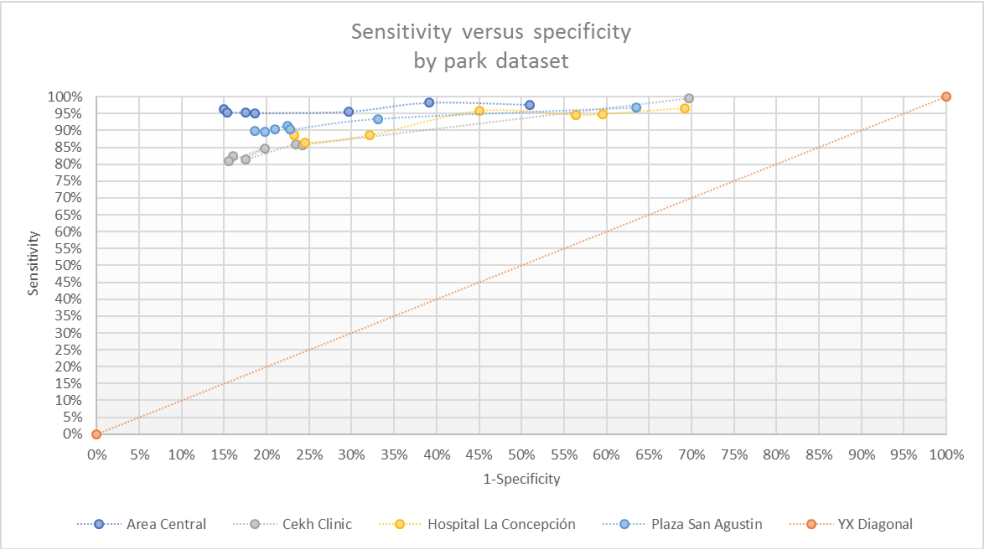

Figure 49 *- In this chart we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing we can conclude that for “Area Central” the optimal model has approximately 95% sensitivity and 85% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 90% sensitivity and 80% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 90% sensitivity and 75% specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 80% sensitivity and 85% specificity, it corresponds to the N2_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

In terms of classifiers, and as we predicted earlier, Random Forest was the unanimous choice as the optimal model for the parks datasets classification, and SMOTE was also the unanimous balancing method chosen.

For the weekdays datasets, by analyzing we already know Random Forest will be the unanimous balancing method, and that SMOTE will also be present on the optimal model for the specified datasets. There’s only left to know if the weekend optimal model will behave equally and if the optimal model results will be as good as the parks datasets results.

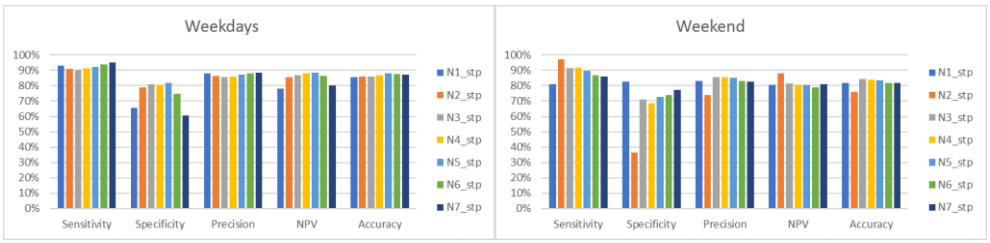

Figure 50 *- This set of charts shows the classification results, that maximize NPV, by each defined metric, by each temporal window and by each week day type dataset.*

In , we explore the week days' datasets, also with the goal of maximizing NPV, which leads to a mean value of approximately 90% for sensitivity, a mean value of approximately 70% for specificity, a mean value of approximately 85% for precision, approximately 85% mean value for NPV and approximately 85% of mean value for accuracy.

Compared to the parks datasets results mean values, for NPV maximization, the results are approximately the same.

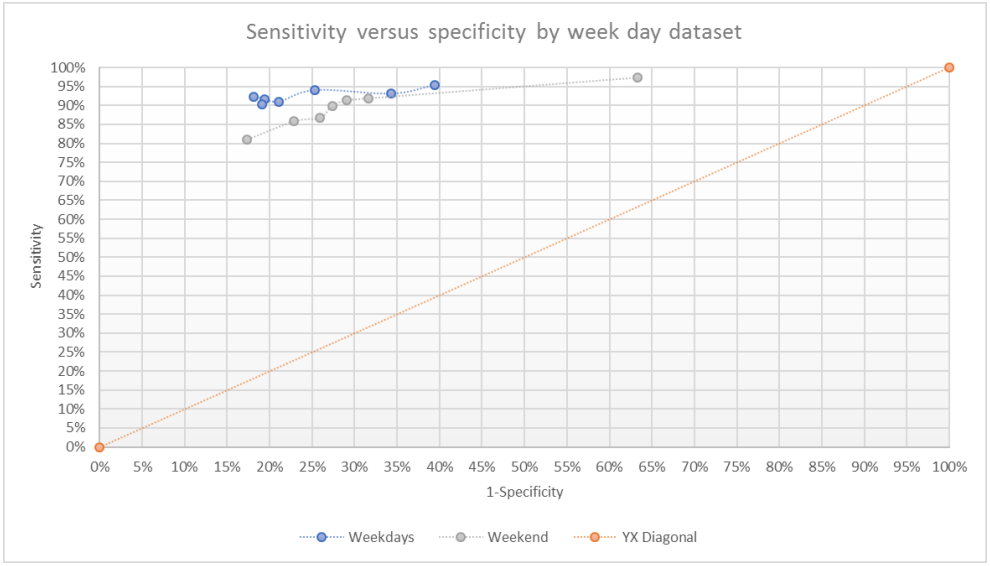

Figure 51 *- In this chart we show the relation between the classification models sensitivity and specificity of the week days' datasets.*

In , we can see that the optimal model for the weekdays dataset has approximately 90% for sensitivity and 80% for specificity, it corresponds to the N5_stp temporal window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 80% for sensitivity and approximately 80% for specificity, it corresponds to the N1_stp temporal

window used to train and test the Random Forest classifier with SMOTE as a training data balancing method.

In sum, for this case, we can say that when we're maximizing NPV the optimal model will reside on the choice of Random Forest and as SMOTE for data balancing.

The explored results in NPV maximization can be explored, in more detail, in .

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
RandomForest	SMOTE	Area Central	N1_stp	95,55%	70,36%	90,51%	84,23%	89,19%
RandomForest	SMOTE	Area Central	N2_stp	95,14%	81,37%	89,61%	90,83%	90,01%
RandomForest	SMOTE	Area Central	N3_stp	95,20%	82,45%	89,14%	91,91%	90,13%
RandomForest	SMOTE	Area Central	N4_stp	95,40%	84,67%	90,37%	92,43%	91,12%
RandomForest	SMOTE	Area Central	N5_stp	96,43%	84,99%	90,98%	93,81%	91,98%
AdaBoost	None	Area Central	N6_stp	98,29%	60,87%	85,80%	93,66%	87,30%
RandomForest	SMOTE	Area Central	N7_stp	97,66%	49,07%	91,53%	78,82%	90,34%
AdaBoost	None	Cekh Clinic	N1_stp	99,49%	30,26%	67,88%	97,55%	71,59%
RandomForest	SMOTE	Cekh Clinic	N2_stp	82,44%	83,90%	85,43%	80,67%	83,12%
RandomForest	SMOTE	Cekh Clinic	N3_stp	81,36%	82,47%	83,74%	79,94%	81,88%
RandomForest	SMOTE	Cekh Clinic	N4_stp	85,71%	75,76%	80,64%	81,82%	81,14%
RandomForest	SMOTE	Cekh Clinic	N5_stp	85,98%	76,51%	82,56%	80,84%	81,85%
RandomForest	SMOTE	Cekh Clinic	N6_stp	84,71%	80,18%	82,32%	82,79%	82,54%
RandomForest	SMOTE	Cekh Clinic	N7_stp	80,80%	84,41%	84,27%	80,96%	82,57%
AdaBoost	None	Hospital La Concepción	N1_stp	94,84%	40,41%	78,09%	77,76%	78,04%
RandomForest	SMOTE	Hospital La Concepción	N2_stp	88,66%	67,89%	81,79%	78,63%	80,75%
RandomForest	SMOTE	Hospital La Concepción	N3_stp	86,46%	75,51%	80,68%	82,50%	81,44%
NaiveBayes	None	Hospital La Concepción	N4_stp	95,79%	54,96%	74,44%	90,50%	78,56%
RandomForest	SMOTE	Hospital La Concepción	N5_stp	88,66%	76,74%	82,64%	84,42%	83,36%
AdaBoost	None	Hospital La Concepción	N6_stp	94,60%	43,65%	77,77%	79,49%	78,08%
AdaBoost	None	Hospital La Concepción	N7_stp	96,63%	30,76%	78,32%	77,89%	78,27%
AdaBoost	None	Plaza San Agustin	N1_stp	96,78%	36,56%	76,02%	84,53%	77,22%
RandomForest	SMOTE	Plaza San Agustin	N2_stp	90,23%	77,22%	85,64%	84,01%	85,04%
RandomForest	SMOTE	Plaza San Agustin	N3_stp	90,33%	79,01%	85,74%	85,40%	85,61%
RandomForest	SMOTE	Plaza San Agustin	N4_stp	89,71%	80,18%	86,03%	85,13%	85,67%
RandomForest	SMOTE	Plaza San Agustin	N5_stp	89,87%	81,33%	86,99%	85,26%	86,29%
RandomForest	SMOTE	Plaza San Agustin	N6_stp	91,43%	77,51%	87,54%	83,95%	86,32%
RandomForest	SMOTE	Plaza San Agustin	N7_stp	93,44%	66,88%	87,02%	81,08%	85,57%
RandomForest	SMOTE	Weekdays	N1_stp	93,09%	65,69%	87,86%	78,09%	85,61%
RandomForest	SMOTE	Weekdays	N2_stp	90,96%	78,91%	86,22%	85,75%	86,04%
RandomForest	SMOTE	Weekdays	N3_stp	90,20%	80,92%	85,72%	86,67%	86,11%
RandomForest	SMOTE	Weekdays	N4_stp	91,52%	80,61%	86,08%	87,88%	86,79%
RandomForest	SMOTE	Weekdays	N5_stp	92,34%	81,86%	87,42%	88,68%	87,91%
RandomForest	SMOTE	Weekdays	N6_stp	94,04%	74,66%	87,92%	86,46%	87,50%
RandomForest	SMOTE	Weekdays	N7_stp	95,32%	60,59%	88,54%	80,22%	87,04%
RandomForest	SMOTE	Weekend	N1_stp	81,00%	82,64%	83,04%	80,56%	81,80%
AdaBoost	Resample	Weekend	N2_stp	97,30%	36,70%	74,05%	87,99%	76,08%
RandomForest	SMOTE	Weekend	N3_stp	91,38%	70,88%	85,39%	81,53%	84,22%
RandomForest	SMOTE	Weekend	N4_stp	91,84%	68,34%	85,40%	80,59%	84,05%
RandomForest	SMOTE	Weekend	N5_stp	89,85%	72,63%	85,00%	80,57%	83,53%
RandomForest	SMOTE	Weekend	N6_stp	86,74%	74,05%	83,26%	78,95%	81,64%
RandomForest	SMOTE	Weekend	N7_stp	85,79%	77,11%	82,82%	80,83%	81,99%

Table 23 *- This table provides the classifiers, balancing types and respective results that maximize NPV, for each dataset and temporal window.*

8.2- Multi-Instance Classification Results

8.2.1- Sensitivity maximization results

Regarding sensitivity maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in .

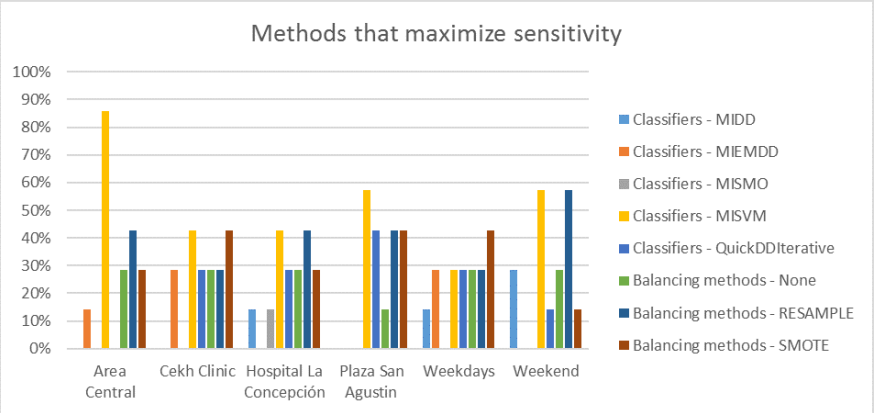

Figure 52 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize sensitivity.*

The analysis to be made to , is complex, due to the diversity of methods that maximize sensitivity. We can see that MISVM has an approximate 50% overall classification use. Following MISVM in terms of use is QuickDDIterative that is present in almost all datasets temporal window classification with an overall use of approximately 25%. The least used classification methods are MIEMDD and MIDD with approximately 10% use and MISMO with approximately 5% use. In terms of balancing methods, Resample is the most used method with an overall use of approximately 40%, followed by SMOTE with approximately 35% use and no type of data balancing with approximately 25% use.

Now let's pay attention to the results from the parks datasets, shown in , of the application of the above classification methods and balancing methods.

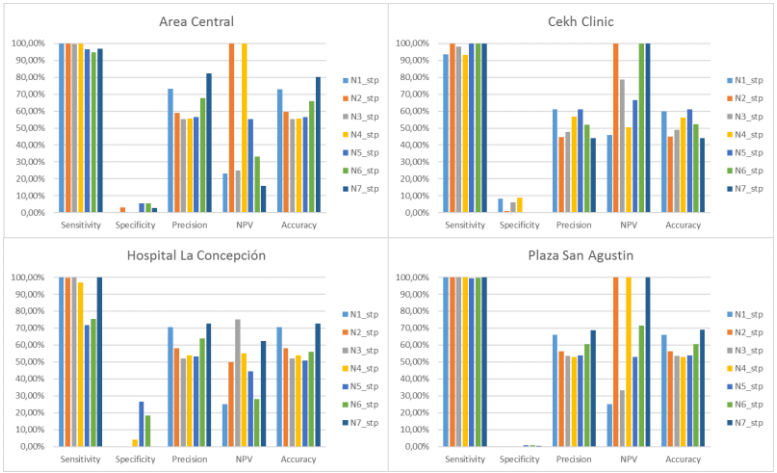

Figure 53 *- This set of charts shows the classification results, that maximize sensitivity, by each defined metric, by each temporal window and by each park dataset*

.

In , the maximization attempt of sensitivity has led to the implicit minimization of specificity. In terms of overall results, we've achieved a mean value of approximately 95% for sensitivity, for specificity the mean value is approximately 5% and in terms of precision, accuracy and NPV, the mean value is at approximately at 60%.

We'll now focus on , that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in **Figure 28** provides the optimal model.

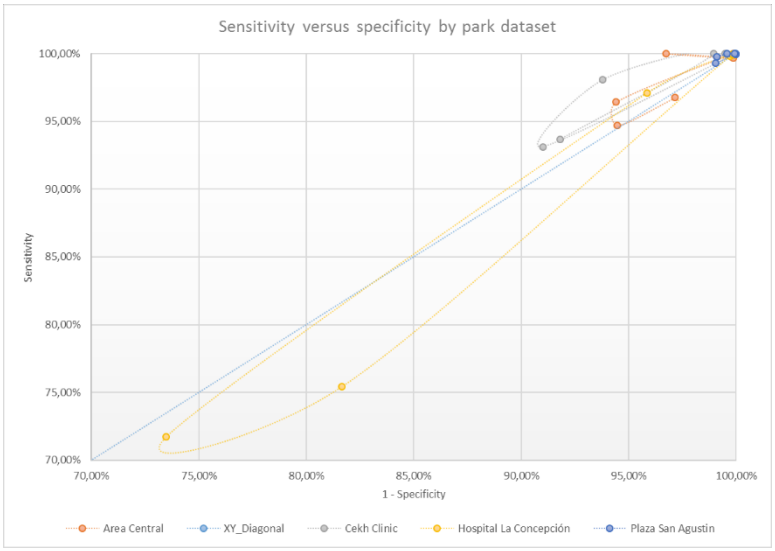

Figure 54 *- In this chart we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing , we can conclude that for "Area Central" the optimal model has approximately 95% sensitivity and 5% specificity, it corresponds to the N5_stp temporal window used to train and test the MISVM classifier with SMOTE as a training data balancing method.

As for "Plaza San Agustin", the optimal model has approximately 100% sensitivity and 1% specificity, it corresponds to the N5_stp temporal window used to train and test the QuickDDIterative classifier with SMOTE as a training data balancing method.

For "Hospital La Concepción", the optimal model has approximately 70% sensitivity and 25% specificity, it corresponds to the N5_stp temporal window used to train and test the QuickDDIterative classifier with RESAMPLE as a training data balancing method.

Finally, for "Cekh Clinic" the optimal model has approximately 95% sensitivity and 10% specificity, it corresponds to the N4_stp temporal window used to train and test the QuickDDIterative classifier with RESAMPLE as a training data balancing method.

We will now explore the discussed the results as well as the optimal model for the week days' datasets.

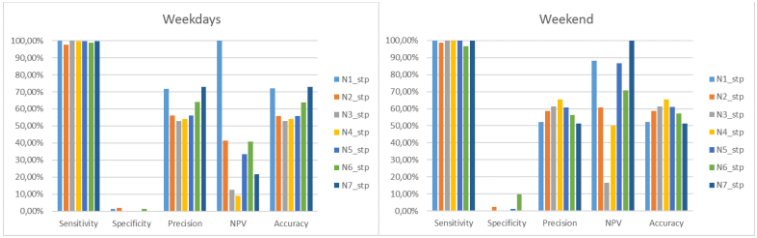

Figure 55 *- This set of charts shows the classification results, that maximize sensitivity, by each defined metric, by each temporal window and by each week day type dataset.*

In , we explore the week days' datasets, also with the goal of maximizing sensitivity, we can observe that the behavior is similar than the one present in the parks datasets i.e. the implicit minimization of specificity, which leads to a mean value of approximately 100% for sensitivity, a mean value of approximately 1% for specificity, for precision and accuracy a mean value of approximately 60% and approximately 50% mean value for NPV.

Compared to the parks datasets results mean values, the results are approximately the same, with the exception of specificity and NPV that show a decrease of 4% and 10% approximately.

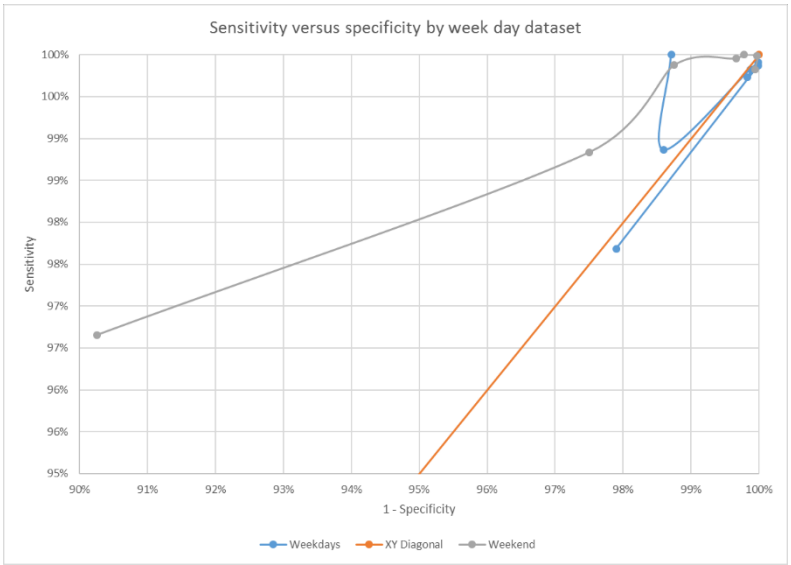

Figure 56 *- In this chart we show the relation between the classification models sensitivity and specificity of the week days' datasets.*

In , we can see that the optimal model for the weekdays dataset has approximately 95% for sensitivity and 2% for specificity, it corresponds to the N2_stp

temporal window used to train and test the MISVM classifier with Resample as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 95% for sensitivity and approximately 10% for specificity, it corresponds to the N6_stp temporal window used to train and test the MIDD classifier with RESAMPLE as a training data balancing method.

In sum, for this case, we can say that when we're maximizing sensitivity the choice for optimal model isn't unanimous and it leads to the minimization of specificity.

The explored results in sensitivity maximization can be explored in **Table 17**, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
MISVM	RESAMPLE	Area Central	N1_stp	99,74%	0,21%	73,16%	23,08%	73,04%
MISVM	None	Area Central	N2_stp	100,00%	3,26%	59,03%	100,00%	59,59%
MIEMDD	RESAMPLE	Area Central	N3_stp	99,67%	0,14%	55,40%	25,00%	55,32%
MISVM	None	Area Central	N4_stp	100,00%	0,12%	55,55%	100,00%	55,57%
MISVM	SMOTE	Area Central	N5_stp	96,43%	5,58%	56,46%	55,19%	56,40%
MISVM	SMOTE	Area Central	N6_stp	94,70%	5,52%	67,89%	33,04%	66,02%
MISVM	RESAMPLE	Area Central	N7_stp	96,75%	2,84%	82,32%	15,74%	80,20%
QuickDDIterative	SMOTE	Cekh Clinic	N1_stp	93,66%	8,19%	60,95%	45,79%	59,88%
MISVM	None	Cekh Clinic	N2_stp	100,00%	1,04%	44,69%	100,00%	45,02%
MISVM	SMOTE	Cekh Clinic	N3_stp	98,06%	6,20%	47,59%	78,63%	48,90%
QuickDDIterative	SMOTE	Cekh Clinic	N4_stp	93,13%	8,99%	56,84%	50,41%	56,34%
MIEMDD	None	Cekh Clinic	N5_stp	99,97%	0,08%	61,21%	66,67%	61,22%
MISVM	RESAMPLE	Cekh Clinic	N6_stp	100,00%	0,50%	52,10%	100,00%	52,21%
MIEMDD	RESAMPLE	Cekh Clinic	N7_stp	100,00%	0,09%	44,18%	100,00%	44,21%
MISVM	RESAMPLE	Hospital La Concepción	N1_stp	99,92%	0,06%	70,59%	25,00%	70,56%
MISMO	SMOTE	Hospital La Concepción	N2_stp	99,79%	0,29%	58,26%	50,00%	58,24%
MISVM	None	Hospital La Concepción	N3_stp	99,95%	0,17%	52,01%	75,00%	52,04%
MISVM	None	Hospital La Concepción	N4_stp	97,08%	4,15%	53,93%	55,15%	53,97%
QuickDDIterative	RESAMPLE	Hospital La Concepción	N5_stp	71,72%	26,52%	53,30%	44,51%	50,89%
QuickDDIterative	SMOTE	Hospital La Concepción	N6_stp	75,43%	18,33%	63,94%	27,99%	55,87%
MIDD	RESAMPLE	Hospital La Concepción	N7_stp	99,96%	0,17%	72,84%	62,50%	72,83%
QuickDDIterative	SMOTE	Plaza San Agustin	N1_stp	99,97%	0,02%	65,98%	25,00%	65,97%
MISVM	RESAMPLE	Plaza San Agustin	N2_stp	100,00%	0,02%	56,35%	100,00%	56,35%
MISVM	RESAMPLE	Plaza San Agustin	N3_stp	99,96%	0,02%	53,65%	33,33%	53,64%
QuickDDIterative	RESAMPLE	Plaza San Agustin	N4_stp	100,00%	0,06%	53,04%	100,00%	53,06%
QuickDDIterative	SMOTE	Plaza San Agustin	N5_stp	99,28%	0,95%	53,83%	53,02%	53,82%
MISVM	None	Plaza San Agustin	N6_stp	99,76%	0,90%	60,47%	71,43%	60,53%
MISVM	SMOTE	Plaza San Agustin	N7_stp	100,00%	0,41%	68,89%	100,00%	68,93%
MISVM	None	Weekdays	N1_stp	100,00%	1,29%	71,84%	100,00%	71,95%
MISVM	RESAMPLE	Weekdays	N2_stp	97,68%	2,10%	56,22%	41,34%	55,89%
MIEMDD	RESAMPLE	Weekdays	N3_stp	99,91%	0,01%	52,82%	12,50%	52,80%
MIEMDD	None	Weekdays	N4_stp	99,87%	0,02%	54,20%	9,09%	54,16%
QuickDDIterative	SMOTE	Weekdays	N5_stp	99,73%	0,17%	56,02%	33,59%	55,97%
MIDD	SMOTE	Weekdays	N6_stp	98,86%	1,40%	64,12%	40,91%	63,83%
QuickDDIterative	SMOTE	Weekdays	N7_stp	99,83%	0,13%	72,87%	21,74%	72,79%
MISVM	RESAMPLE	Weekend	N1_stp	99,96%	0,34%	52,13%	88,24%	52,20%
MIDD	SMOTE	Weekend	N2_stp	98,84%	2,50%	58,57%	60,63%	58,60%
MISVM	None	Weekend	N3_stp	99,82%	0,06%	61,39%	16,67%	61,33%
QuickDDIterative	None	Weekend	N4_stp	99,98%	0,03%	65,48%	50,00%	65,48%
MISVM	RESAMPLE	Weekend	N5_stp	99,88%	1,25%	60,78%	86,79%	60,93%
MIDD	RESAMPLE	Weekend	N6_stp	96,65%	9,74%	56,38%	70,69%	57,28%
MISVM	RESAMPLE	Weekend	N7_stp	100,00%	0,22%	51,37%	100,00%	51,42%

Table 24 - This table provides the classifiers, balancing types and respective results that maximize sensitivity for each dataset and temporal window.

8.2.2- Precision maximization results

Concerning precision maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in .

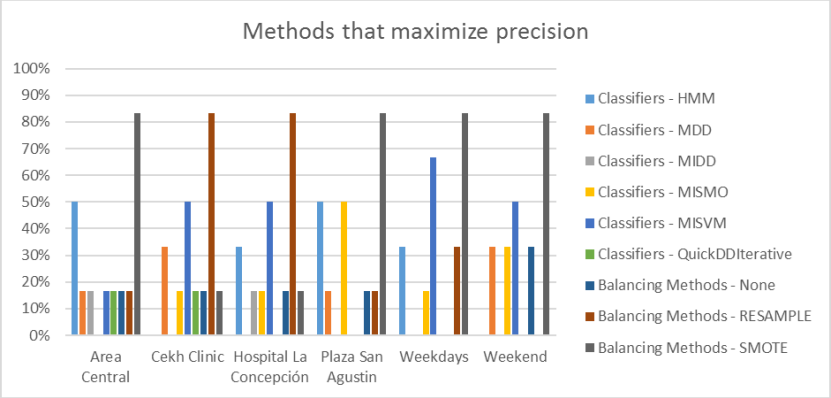

Figure 57 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize precision.*

The analysis to be made to , is complex, due to the diversity of methods that maximize precision. We can see that MISVM has an approximate 30% overall classification use. Following MISVM in terms of use are HMM and MISMO that are present in almost all datasets temporal window classification with an overall use of approximately 25% and 20% respectively. The least used classification methods are MDD with approximately 15% use and MIDD with approximately 5% use. In terms of balancing methods, SMOTE is the most used method with an overall use of approximately 50% each, followed by Resample with approximately 35% use and no type of data balancing with approximately 15% use.

Now let's pay attention to the results from the parks datasets, shown in , of the application of the above classification methods and balancing methods.

Figure 58 *- This set of charts shows the classification results, that maximize precision, by each defined metric, by each temporal window and by each park dataset.*

In , the maximization attempt of precision we've achieved a mean value of approximately 25% for sensitivity, for specificity the mean value is approximately 80% and in terms of precision the mean value is approximately 75%, as for NPV the mean value is at approximately at 45% and in terms of accuracy the mean value is at 50%.

We'll now focus on , that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in **Figure 28** provides the optimal model.

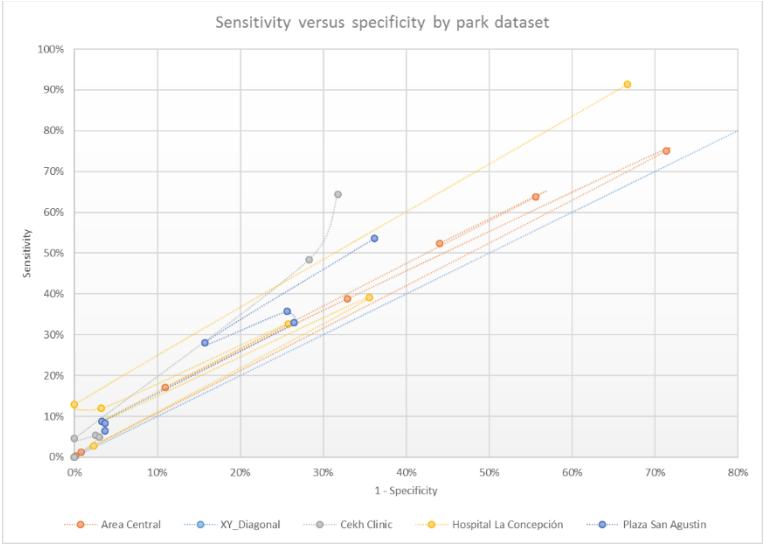

Figure 59 *- In this chart we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing , we can conclude that for “Area Central” the optimal model has approximately 50% sensitivity and 55% specificity, it corresponds to the N1_stp temporal window used to train and test the HMM classifier with SMOTE as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 55% sensitivity and 65% specificity, it corresponds to the N3_stp temporal window used to train and test the MDD classifier with SMOTE as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 90% sensitivity and 35% specificity, it corresponds to the N4_stp temporal window used to train and test the MISVM classifier with RESAMPLE as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 65% sensitivity and 70% specificity, it corresponds to the N7_stp temporal window used to train and test the QuickDDIterative classifier with SMOTE as a training data balancing method.

We will now explore the discussed the results as well as the optimal model for the week days’ datasets.

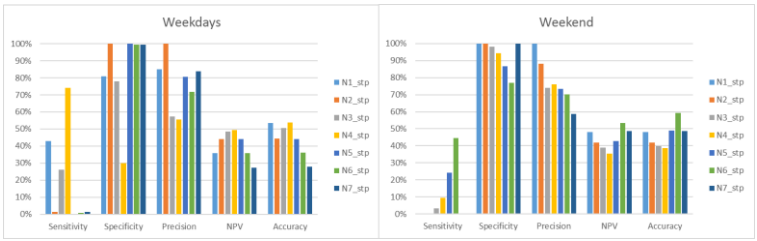

Figure 60 *- This set of charts shows the classification results, that maximize precision, by each defined metric, by each temporal window and by each week day type dataset.*

In , we explore the week days’ datasets, also with the goal of maximizing specificity, we can observe that the behavior is similar than the one present in the parks datasets i.e. the implicit minimization of sensitivity, which leads to a mean value of approximately 15% for sensitivity, a mean value of approximately 90% for specificity, for precision the mean value is approximately 75%, the mean value for NPV is approximately 40% and for accuracy has approximately 45% mean value.

Compared to the parks datasets results mean values, the results are approximately the same in terms of precision but in terms of specificity there has been an increase of approximately 10%, sensitivity there has been a decrease of approximately 10% and for NPV and accuracy there has been a decrease of approximately 5% for both.

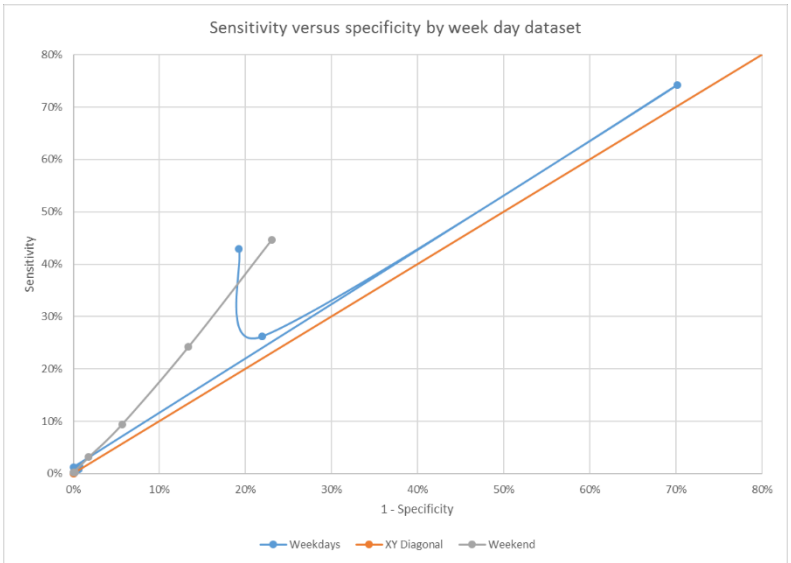

Figure 61 *- In this chart we show the relation between the classification models sensitivity and specificity of the week days' datasets.*

In we can see that the optimal model for the weekdays dataset has approximately 45% for sensitivity and 80% for specificity, it corresponds to the N1_stp temporal window used to train and test the MISVM classifier with SMOTE as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 45% for sensitivity and approximately 80% for specificity, it corresponds to the N6_stp temporal window used to train and test the MDD classifier with SMOTE as a training data balancing method.

The explored results in precision maximization can be explored in **Table 17**, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
HMM	SMOTE	Area Central	N1_stp	52,37%	55,99%	76,45%	30,11%	53,34%
HMM	RESAMPLE	Area Central	N2_stp	75,07%	28,65%	59,57%	45,07%	55,73%
QuickDDIterative	SMOTE	Area Central	N3_stp	63,83%	44,34%	58,80%	49,62%	55,15%
MISVM	SMOTE	Area Central	N4_stp	0,35%	99,88%	78,85%	44,54%	44,62%
MDD	SMOTE	Area Central	N5_stp	1,27%	99,19%	66,67%	44,17%	44,41%
MIDD	SMOTE	Area Central	N6_stp	38,80%	67,09%	71,33%	34,19%	47,90%
HMM	None	Area Central	N7_stp	17,08%	89,07%	88,03%	18,58%	29,70%
MISVM	SMOTE	Cekh Clinic	N1_stp	0,03%	100,00%	100,00%	39,53%	39,54%
MISVM	RESAMPLE	Cekh Clinic	N2_stp	0,04%	100,00%	100,00%	55,57%	55,58%
MISVM	None	Cekh Clinic	N3_stp	4,61%	100,00%	100,00%	54,69%	55,66%
MDD	RESAMPLE	Cekh Clinic	N4_stp	5,41%	97,50%	73,54%	44,47%	45,68%
MDD	RESAMPLE	Cekh Clinic	N5_stp	4,84%	96,97%	71,60%	39,25%	40,59%
MISMO	RESAMPLE	Cekh Clinic	N6_stp	48,44%	71,73%	64,96%	56,25%	59,62%
QuickDDIterative	RESAMPLE	Cekh Clinic	N7_stp	64,44%	68,21%	61,59%	70,81%	66,55%
HMM	RESAMPLE	Hospital La Concepción	N1_stp	32,72%	74,25%	75,32%	31,48%	44,93%
MISVM	RESAMPLE	Hospital La Concepción	N2_stp	12,92%	100,00%	100,00%	45,14%	49,27%
MISMO	RESAMPLE	Hospital La Concepción	N3_stp	11,95%	96,75%	79,95%	50,35%	52,66%
MISVM	RESAMPLE	Hospital La Concepción	N4_stp	91,30%	33,33%	61,28%	76,82%	64,41%
MIDD	SMOTE	Hospital La Concepción	N5_stp	39,17%	64,42%	56,28%	47,53%	50,81%
MISVM	RESAMPLE	Hospital La Concepción	N6_stp	8,78%	96,71%	83,69%	35,56%	38,88%
HMM	None	Hospital La Concepción	N7_stp	2,82%	97,70%	76,63%	27,29%	28,61%
MISMO	SMOTE	Plaza San Agustin	N1_stp	8,29%	96,32%	81,38%	35,12%	38,23%
HMM	None	Plaza San Agustin	N2_stp	28,04%	84,30%	69,79%	47,52%	52,56%
MDD	SMOTE	Plaza San Agustin	N3_stp	53,59%	63,82%	63,20%	54,25%	58,33%
HMM	SMOTE	Plaza San Agustin	N4_stp	35,68%	74,39%	61,27%	50,46%	53,81%
MISMO	SMOTE	Plaza San Agustin	N5_stp	8,80%	96,69%	75,57%	47,67%	49,42%
MISMO	SMOTE	Plaza San Agustin	N6_stp	6,52%	96,34%	73,04%	40,41%	42,17%
HMM	RESAMPLE	Plaza San Agustin	N7_stp	33,01%	73,50%	73,28%	33,26%	45,66%
MISVM	SMOTE	Weekdays	N1_stp	42,86%	80,76%	84,87%	35,95%	53,63%
MISVM	SMOTE	Weekdays	N2_stp	1,26%	100,00%	100,00%	44,03%	44,43%
MISVM	SMOTE	Weekdays	N3_stp	26,20%	78,07%	57,24%	48,56%	50,66%
MISMO	RESAMPLE	Weekdays	N4_stp	74,20%	29,82%	55,61%	49,38%	53,89%
HMM	SMOTE	Weekdays	N5_stp	0,08%	99,98%	80,65%	43,93%	43,95%
HMM	RESAMPLE	Weekdays	N6_stp	0,89%	99,38%	71,81%	35,97%	36,25%
MISVM	SMOTE	Weekdays	N7_stp	1,33%	99,30%	83,73%	27,24%	27,90%
MISVM	SMOTE	Weekend	N1_stp	0,27%	100,00%	100,00%	48,01%	48,08%
MISVM	None	Weekend	N2_stp	0,28%	99,95%	88,24%	41,82%	41,91%
MISMO	SMOTE	Weekend	N3_stp	3,19%	98,21%	73,98%	38,92%	39,85%
MISMO	SMOTE	Weekend	N4_stp	9,42%	94,36%	75,99%	35,45%	38,74%
MDD	SMOTE	Weekend	N5_stp	24,19%	86,64%	73,51%	42,73%	48,86%
MDD	SMOTE	Weekend	N6_stp	44,64%	76,95%	70,04%	53,52%	59,28%
MISVM	None	Weekend	N7_stp	0,21%	99,85%	58,82%	48,70%	48,72%

Table 25 - This table provides the classifiers, balancing types and respective results that maximize precision for each dataset and temporal window.

8.2.3- NPV maximization results

Concerning NPV maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in .

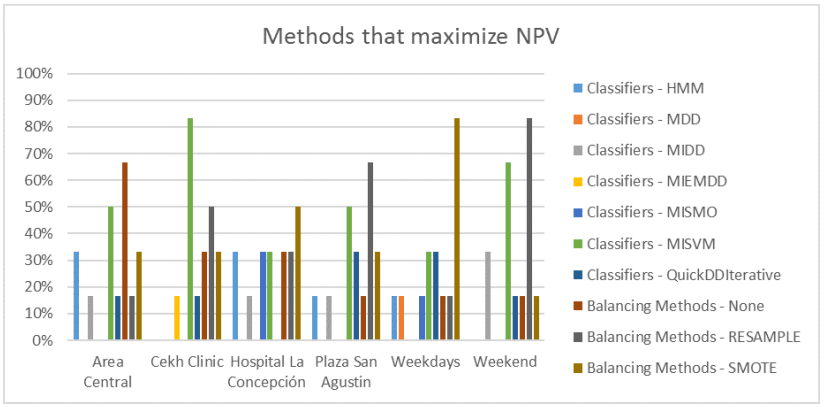

Figure 62 *- In this chart is shown the classification methods and the balancing methods to be used, to maximize NPV.*

The analysis to be made to , is complex, due to the diversity of methods that maximize NPV. We can see that MISVM has an approximate 45% overall classification use. Following MISVM in terms of use are QuickDDIterative and MISMO that are present in almost all datasets temporal window classification with an overall use of approximately 15% each. The least used classification methods are MIDD with approximately 10% use and MDD, MIEMDD and MISMO with approximately 5% use. In terms of balancing methods, Resample is the most used method with an overall use of approximately 40% each, followed by SMOTE with approximately 35% use and no type of data balancing with approximately 25% use.

Now let's pay attention to the results from the parks datasets, shown in , of the application of the above classification methods and balancing methods.

Figure 63 *- This set of charts shows the classification results, that maximize NPV, by each defined metric, by each temporal window and by each park dataset.*

In , the maximization attempt of precision we've achieved a mean value of approximately 90% for sensitivity, for specificity the mean value is approximately 10% and in terms of precision the mean value is approximately 60%, as for NPV the mean value is at approximately at 65% and in terms of accuracy the mean value is at 60%.

We'll now focus on , that will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in **Figure 28** provides the optimal model.

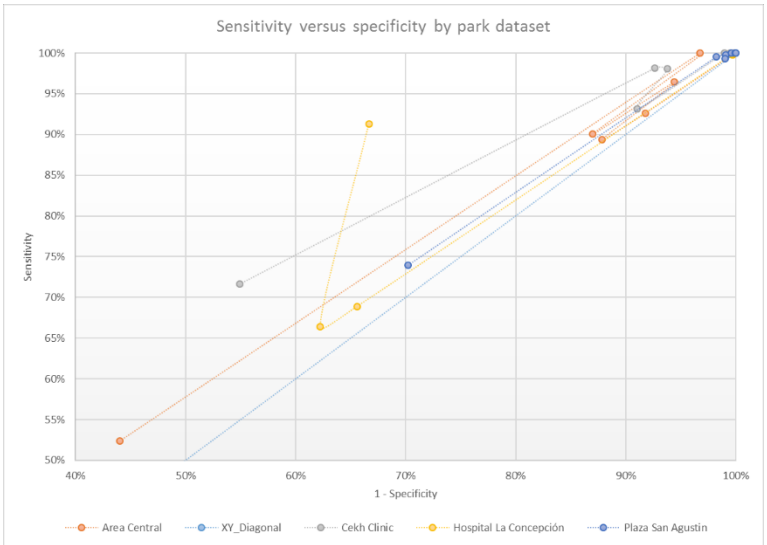

Figure 64 *- In this chart we show the relation between the classification models sensitivity and specificity of the parks datasets.*

By analyzing , we can conclude that for “Area Central” the optimal model has approximately 50% sensitivity and 55% specificity, it corresponds to the N1_stp temporal window used to train and test the HMM classifier with SMOTE as a training data balancing method.

As for “Plaza San Agustin”, the optimal model has approximately 75% sensitivity and 30% specificity, it corresponds to the N1_stp temporal window used to train and test the HMM classifier with Resample as a training data balancing method.

For “Hospital La Concepción”, the optimal model has approximately 90% sensitivity and 35% specificity, it corresponds to the N4_stp temporal window used to train and test the MISVM classifier with Resample as a training data balancing method.

Finally, for “Cekh Clinic” the optimal model has approximately 70% sensitivity and 45% specificity, it corresponds to the N1_stp temporal window used to train and test the MISVM classifier with no data balancing.

We will now explore the discussed the results as well as the optimal model for the week days’ datasets.

Figure 65 *- This set of charts shows the classification results, that maximize NPV, by each defined metric, by each temporal window and by each week day type dataset.*

In , we explore the week days' datasets, also with the goal of maximizing NPV, which leads to a mean value of approximately 95% for sensitivity, a mean value of approximately 10% for specificity, for precision the mean value is approximately 60%, the mean value for NPV is approximately 65% and for accuracy has approximately 60% mean value.

Compared to the parks datasets results mean values, the results are approximately the same in terms of specificity, precision, accuracy and NPV but in terms of sensitivity there has been an increase of approximately 5%.

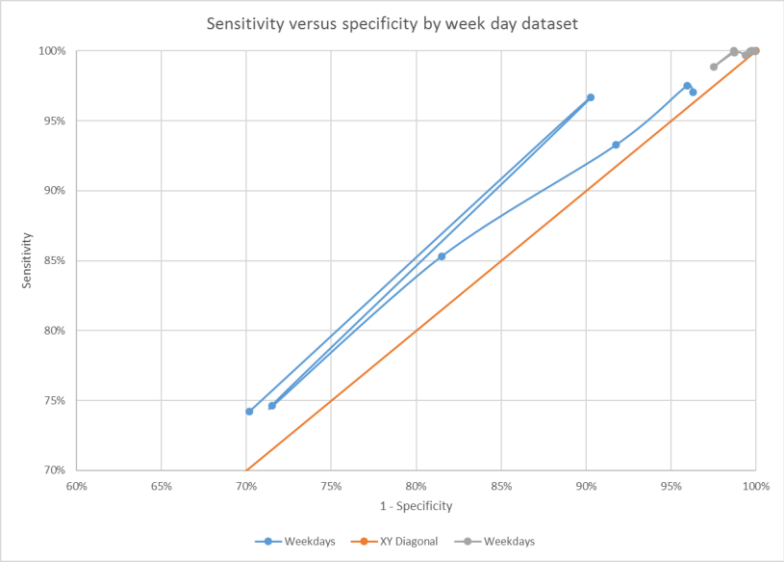

Figure 66 *- In this chart we show the relation between the classification models sensitivity and specificity of the week days' datasets.*

In we can see that the optimal model for the weekdays dataset has approximately 75% for sensitivity and 30% for specificity, it corresponds to the N4_stp temporal window used to train and test the MISMO classifier with Resample as a training data balancing method.

Regarding the weekend dataset, the optimal model has approximately 95% for sensitivity and approximately 10% for specificity, it corresponds to the N6_stp temporal

window used to train and test the MIDD classifier with Resample as a training data balancing method.

The explored results in NPV maximization can be explored in **Table 17**, with more detail.

Classifier	Balancing	Dataset	Temporal Window	Sensitivity	Specificity	Precision	NPV	Accuracy
HMM	SMOTE	Area Central	N1_stp	52,37%	55,99%	76,45%	30,11%	53,34%
MISVM	None	Area Central	N2_stp	100,00%	3,26%	59,03%	100,00%	59,59%
MIDD	None	Area Central	N3_stp	90,07%	13,03%	56,31%	51,32%	55,75%
MISVM	None	Area Central	N4_stp	100,00%	0,12%	55,55%	100,00%	55,57%
MISVM	SMOTE	Area Central	N5_stp	96,43%	5,58%	56,46%	55,19%	56,40%
HMM	RESAMPLE	Area Central	N6_stp	89,40%	12,17%	68,31%	35,16%	64,63%
QuickDDIterative	None	Area Central	N7_stp	92,63%	8,25%	82,52%	19,30%	77,76%
MISVM	None	Cekh Clinic	N1_stp	71,64%	45,07%	66,62%	50,94%	61,14%
MISVM	None	Cekh Clinic	N2_stp	100,00%	1,04%	44,69%	100,00%	45,02%
MISVM	SMOTE	Cekh Clinic	N3_stp	98,06%	6,20%	47,59%	78,63%	48,90%
QuickDDIterative	SMOTE	Cekh Clinic	N4_stp	93,13%	8,99%	56,84%	50,41%	56,34%
MISVM	RESAMPLE	Cekh Clinic	N5_stp	98,16%	7,39%	62,57%	71,77%	62,94%
MISVM	RESAMPLE	Cekh Clinic	N6_stp	100,00%	0,50%	52,10%	100,00%	52,21%
MIEMDD	RESAMPLE	Cekh Clinic	N7_stp	100,00%	0,09%	44,18%	100,00%	44,21%
MISMO	None	Hospital La Concepción	N1_stp	99,90%	0,16%	70,61%	38,46%	70,57%
MISMO	SMOTE	Hospital La Concepción	N2_stp	99,79%	0,29%	58,26%	50,00%	58,24%
MISVM	None	Hospital La Concepción	N3_stp	99,95%	0,17%	52,01%	75,00%	52,04%
MISVM	RESAMPLE	Hospital La Concepción	N4_stp	91,30%	33,33%	61,28%	76,82%	64,41%
HMM	SMOTE	Hospital La Concepción	N5_stp	68,89%	34,43%	55,11%	48,65%	53,00%
HMM	SMOTE	Hospital La Concepción	N6_stp	66,44%	37,76%	67,21%	36,95%	56,62%
MIDD	RESAMPLE	Hospital La Concepción	N7_stp	99,96%	0,17%	72,84%	62,50%	72,83%
HMM	RESAMPLE	Plaza San Agustin	N1_stp	73,98%	29,80%	67,19%	37,09%	58,97%
MISVM	RESAMPLE	Plaza San Agustin	N2_stp	100,00%	0,02%	56,35%	100,00%	56,35%
MIDD	RESAMPLE	Plaza San Agustin	N3_stp	99,52%	1,79%	54,03%	76,32%	54,27%
QuickDDIterative	RESAMPLE	Plaza San Agustin	N4_stp	100,00%	0,06%	53,04%	100,00%	53,06%
QuickDDIterative	SMOTE	Plaza San Agustin	N5_stp	99,28%	0,95%	53,83%	53,02%	53,82%
MISVM	None	Plaza San Agustin	N6_stp	99,76%	0,90%	60,47%	71,43%	60,53%
MISVM	SMOTE	Plaza San Agustin	N7_stp	100,00%	0,41%	68,89%	100,00%	68,93%
MISVM	None	Weekdays	N1_stp	100,00%	1,29%	71,84%	100,00%	71,95%
MDD	SMOTE	Weekdays	N2_stp	93,28%	8,23%	56,66%	48,77%	56,08%
QuickDDIterative	SMOTE	Weekdays	N3_stp	97,04%	3,70%	52,96%	52,78%	52,96%
MISMO	RESAMPLE	Weekdays	N4_stp	74,20%	29,82%	55,61%	49,38%	53,89%
MISVM	SMOTE	Weekdays	N5_stp	85,29%	18,51%	57,17%	49,68%	55,94%
QuickDDIterative	SMOTE	Weekdays	N6_stp	97,53%	4,04%	64,44%	47,78%	63,93%
HMM	SMOTE	Weekdays	N7_stp	74,64%	28,48%	73,73%	29,46%	62,13%
MISVM	RESAMPLE	Weekend	N1_stp	99,96%	0,34%	52,13%	88,24%	52,20%
MIDD	SMOTE	Weekend	N2_stp	98,84%	2,50%	58,57%	60,63%	58,60%
MISVM	RESAMPLE	Weekend	N3_stp	99,70%	0,61%	61,50%	56,41%	61,47%
QuickDDIterative	None	Weekend	N4_stp	99,98%	0,03%	65,48%	50,00%	65,48%
MISVM	RESAMPLE	Weekend	N5_stp	99,88%	1,25%	60,78%	86,79%	60,93%
MIDD	RESAMPLE	Weekend	N6_stp	96,65%	9,74%	56,38%	70,69%	57,28%
MISVM	RESAMPLE	Weekend	N7_stp	100,00%	0,22%	51,37%	100,00%	51,42%

Table 26 - This table provides the classifiers, balancing types and respective results that maximize NPV for each dataset and temporal window.