RedParkMiner
A tool for data analysis in car parking business

Nuno Miguel Sousa da Silva Pinto
nuno.s.s.pinto@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2017

1. 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.

2. 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.

3. Motivation

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 mining component to completely achieve its purpose.

After many iterations on the manipulation of the data warehouses' model into more interesting datasets and the consequent classification method experimentation on these datasets, we have come to realize that the effort made along these developments lead to the definition and creation of a data science tool. This data science tool can be independent of a targeted domain and a great help to various data analysis general stakeholders.

3.1 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 methodology 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.

3.2 Source Data Model

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. The park dimension describes the location where the parking session occurs.

![Figure 1 - Data warehouses’ star schema from which the data was extracted.](image)

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 a specific park customer were ignored as well as parking sessions from Portugal parks due to privacy issues.

3.3 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 poteniate 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.

3.3.1 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.

3.3.2 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.

3.3.3 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.

3.4 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 counts by park dataset and consequent temporal window.](image)

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 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.

3.5 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.

3.5.1 Instance and Time-Window based Oriented
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.

3.5.2 Multi-Instance Oriented

Regarding the multi-instance oriented classification, we’ve used Hidden Markov Models (HMM) [5], two variations of Support Vector Machines (MISVM) following Andrews ”maximum bag margin formulation” [6] and MISM following Platt’s SMO algorithm to solve the quadratic programming problem [10], 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.

3.6 Classification Metrics

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

\[
\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (1) \\
\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \quad (2) \\
\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (3) \\
\text{NPV} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}} \quad (4) \\
\text{Accuracy} = \frac{\text{Positives} + \text{Negatives}}{\text{Positives} + \text{Negatives}} \quad (5)
\]

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.

4. Requirement Analysis

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.

In the diagram bellow (Figure 3), we have presented the relation that RedParkMiner tool has with its interveners.

![Context Diagram](image)

**Figure 3 - Context Diagram**

4.1 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

4.2 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 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 attributes.
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.

4.3 Use Cases

Figure 4 illustrates the most relevant use cases for this system and in Figure 5 the association between these and the already enumerated requirements.

5. System Architecture

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 6), the technical solution is presented, with all general components and technologies used until this point.
5.1.1 Sources

Our technological stack decisions have been oriented to promote maximum compatibility/extendability 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.

5.1.2 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.

5.1.3 Data Parcelling Processes

Our data may sometimes be stratified, and we may want to explore each stratum separately, so in the Data Parcelling Process our goal is to divide the already integrated dataset into previously configured parcels that correspond to the previous discussed strata. The parcelling process can be divided into two types: data oriented parcelling and custom parcelling. In data oriented parcelling, 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 parcelling, 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 each of the park data values and the other two parts are oriented to the parking occurrences occurring during the week or the weekend.

5.1.4 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 parcelled 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 parcelled 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 will explain the structure for a window size equal to 1 and 7. The temporal window with a window size 1 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 7 will have information about a specific occurrence, its consecutive 6 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.

5.1.5 Data Classification Processes

Usually, at our disposal there are classification libraries that are composed by a various set of classification methods, it is also usual that these methods are experimented separately in order to better understand their behavior with specific datasets. 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 HMM, MDD, MIDD, MIE MDD, MISMO, MISVN, QuickDDIterative and as used before the balancing methods are SMOTE and Resample.

5.1.6 Best Classification Method Process

When we are experimenting classification methods with datasets, our focus will always be on how well these methods behave in terms of classification. For example, after binary classification is performed we will 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 about 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.

5.1.7 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 parcelled data, the windowed parcelled 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.

6. Case Study

In this chapter, we will discuss the application of the tool RedParkMiner to the car parking business scope.

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. Even though we have proposed a predefined set of metrics used to analyze the resulting best classification approaches provided by RedParkMiner, for resume purposes, we will only consider the accuracy metric maximization perspective.

For these analyses, we have to bear in mind that when we’re maximizing 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.

In these analyses, we will use ROC charts because we will be comparing models’ optimality, in binary classification, being common the use of these types of charts. For each of the models, we will only focus on those that are farthest from the 45º diagonal that crosses the graph (which represents a poor model of a random classifier), the optimal model will be the point closest to the point (X=0%, Y=100%).

6.1 Instance and Time-Window based Classification Results

In this section, we will analyze the best results, obtained through the maximization of accuracy and obtained through the classification of the time-window based data from the datasets.

Instance classifications are implicit in this analysis due to the fact they are particular cases of the time-window based classification, as described in the time-window based data description section.

In what concerns accuracy maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in Figure 7.
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 7 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 8, of the application of the above classification methods and balancing methods.

In Figure 8, 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 9, which will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in Figure 8 provides the optimal model.

By analyzing Figure 9, 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.

In Figure 10, 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.

In Figure 11, 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.

6.2 Multi-Instance Classification Results

In this section, we will analyze the best results, obtained through the maximization of accuracy and obtained through the classification of the multi-instance data from the datasets.

In what concerns accuracy maximization, the classification approaches and data balancing approaches chosen for the configured datasets are available in Figure 12.

The analysis to be made to Figure 12 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 10, of the application of the above classification methods and balancing methods.

In Figure 13, 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 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 14, which will show the relation between sensitivity and specificity, in order to understand which of the temporal windows in Figure 13 provides the optimal model.

By analyzing Figure 14, 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 Agustín”, 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_temp window used to train and test the QuickDDIterative classifier with RESAMPLE as a training data balancing method.

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

![Figure 15](image1.png) - This set of charts shows the classification results, which maximize accuracy, by each defined metric, by each temporal window and by each weekday type dataset.

In Figure 15, 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 16](image2.png) - In this chart, we show the relation between the classification models sensitivity and specificity of the weekdays’ datasets.

In Figure 16, 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_temp 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_temp 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.

6.3 Critical Analysis

In this section, we will discuss the tools’ overall results, having in mind the information retrieved 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.

7. 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 potentiates its use. Then we defined and applied a set of methodologies in order to potentiating 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.

8. References


