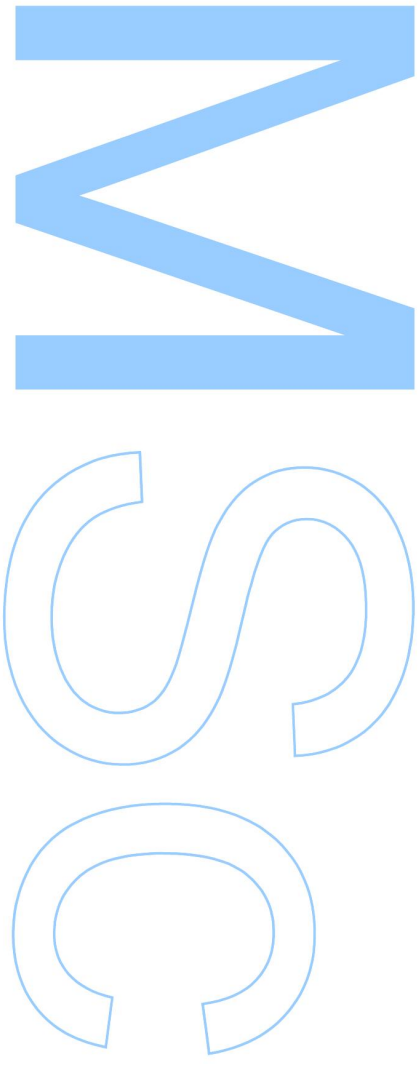
Payment Default Prediction in Telco Services

Ricardo Dias Azevedo
Mestrado em Ciência de Computadores
Departamento de Ciência de Computadores
2019

Orientador
Nuno Miguel Pereira Moniz, Professor Auxiliar Convidado,
Faculdade de Ciências da Universidade do Porto

Coorientador
Rita Paula Almeida Ribeiro, Professor Auxiliar,
Faculdade de Ciências da Universidade do Porto

Orientador Externo
Nuno Filipe Paiva



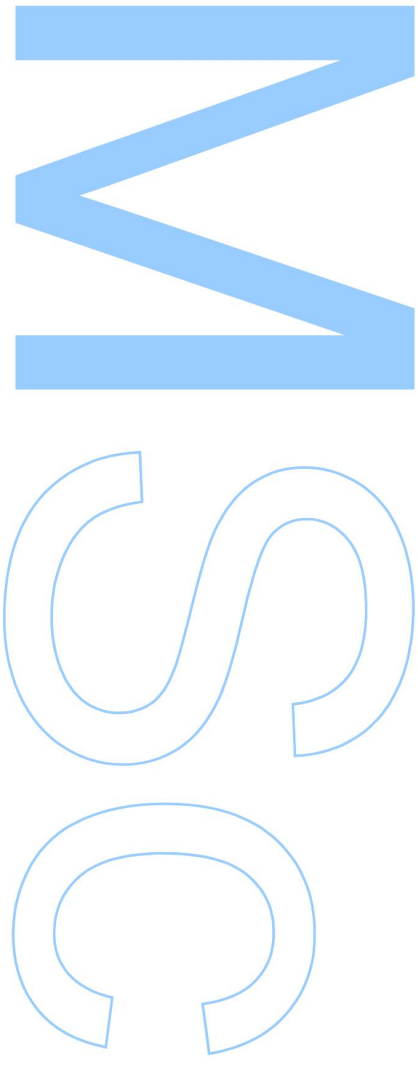
U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____



Abstract

Subscription-based services are a widely adopted business model by digital companies, presenting a constant and significant growth. Typically, clients are charged on a monthly basis, granting access to the service even before the due date, leading to a considerable risk of clients that churn due to lack of payment. Therefore, in order to avoid the loss of customers, it is fundamental to develop and implement several effective strategies.

This thesis tackles the problem of identifying clients at risk of not paying and, therefore, leaving the service. Frequently, the subset of non-paying clients is under-represented in the available dataset, introducing, therefore, a specific and challenging problem of imbalanced domain. Since the identification of non-paying clients is a non-trivial task for humans, involving a high-dimensional number of behavioural patterns, we propose a Machine Learning approach, based on binary classification of data concerning the customer-company interaction that identifies such customers, allowing companies to redirect efforts towards vulnerable segments of the customer base.

We validate the performance of several classifiers applying different strategies for imbalance domains and obtain encouraging predictive performance. We also test the performance of the models on the subdivision of customers in several payment behaviour groups. In terms of results, we verified that the use of re-sampling and threshold methods allowed to increase the predictive performance, with the best overall score accomplished with the implementation of XGBoost. Contrary to expectations, there was no improvement in results after clustering was applied.

Considering the satisfactory results obtained, we find in this thesis a valid answer to the problem stated, always taking in mind the existence of limitations and the need for potential future improvements.

Key-Words: Imbalanced Classification, Involuntary Churn Prediction, Data Mining, Machine Learning, Industry, Subscription-Based Services, Telco

Resumo

Atualmente, os serviços baseados em subscrição constituem um importante modelo de negócios amplamente adotado por empresas digitais, encontrando-se em constante crescimento e evolução. Por norma, os clientes que contratam tais serviços são cobrados mensalmente, o que lhes concede acesso ao serviço mesmo antes da data de vencimento. Tal contribui para um risco acrescido de clientes que evoluem para *churn* devido à falta de pagamento. Deste modo, para evitar a perda de clientes é fundamental desenvolver e implementar estratégias eficazes de minimização de danos.

A presente tese aborda o problema da identificação de clientes em risco de não pagamento e, portanto, abandono do serviço. Frequentemente, o subconjunto de clientes "não pagadores" encontra-se subrepresentado no conjunto de dados disponível, introduzindo, portanto, um problema específico e desafiador do domínio desbalanceado. Dado que a identificação deste tipo de clientes não é uma tarefa trivial para o ser humano, envolvendo um vasto número de padrões comportamentais, propomos uma abordagem de *Machine Learning*, baseada na classificação binária de dados relativos à interação cliente-empresa. Através desta abordagem, permite-se a correta identificação da população de clientes em questão, auxiliando as empresas no redirecionamento de esforços para os segmentos mais vulneráveis.

Para tal, validámos o desempenho de vários classificadores, aplicando diferentes estratégias específicas para domínios desbalanceados,com obtenção de resultados encorajadores. Foi igualmente testado o desempenho dos modelos de *clustering* de clientes, após a sua divisão em vários grupos tendo em conta o comportamento de pagamento. No que respeita os resultados obtidos, verificámos que o uso de métodos de *re-sampling* e *threshold* permitiu incrementar o desempenho preditivo, com a melhor pontuação global obtida aquando da implementação do método de *ensemble XGBoost*. Contrariamente ao esperado, não se objetivou melhoria nos resultados após a aplicação dos métodos de *clustering*.

Em suma, considerando os resultados satisfatórios obtidos, encontramos nesta tese uma resposta válida para o problema enunciado, tendo sempre em mente a existência de limitações e a necessidade de possíveis melhorias futuras.

Palavras-Chave: Classificação Desbalanceada, Previsão de *Churn* Involuntário, *Data Mining*, *Machine Learning*, Indústria, Serviços de Assinatura, Telecomunicações

Acknowledgements

Firstly, a special thank you to my parents and my sister for always supporting my choices and showing care and interest in my professional performance and achievements.

Next, to my girlfriend Sara, not only for the understanding and dedication but also for the help and affection at this crucial stage and for being, always and under all circumstances, my rock.

To my mentors, Rita Ribeiro and Nuno Moniz, for their help and support in finding solutions to the difficulties sometimes experienced.

Finally, to all my co-workers and my external advisor Nuno Paiva, not only for the excellent welcome but also for the motivation and learning moments provided.

To Sara

Contents

List of Tables

List of Figures

Acronyms

AUC Area Under Curve

CART Classification and Regression Trees

CLARA Clustering LARge Applications

DD Direct Debit

DDL Due Date Limit

EDA Exploratory Data Analysis

FN False Negative

FP False Positive

FPR False Positive Rate

GLM Generalised Linear Models

KDD Knowledge Discovery in Databases

PAM Partitioning Around Medoids

PP Payment Plan

RF Random Forest

ROC Receiver Operating Characteristic

ROS Random Over-sampling

RUS Random Under-sampling

SMOTE Synthetic Minority Over-sampling Technique

SVM Support Vector Machine

TN True Negative

TP True Positive

TPR True Positive Rate

XGBoost Extreme Gradient Boosting

Chapter 1

Introduction

In this chapter, the problem is presented along with related motivation and objectives.

1.1 Context and Problem Definition

Within every billing cycle, companies that provide subscription-based services face an important problem - the non-paying customer, that is, the one who will not pay by the stipulated deadline for a variety of reasons. Whenever the customer does not pay in a predetermined date, the company will cancel the provided service, a response usually referred to as involuntary churn.

In order to avoid loss or disconnection of the customer, such companies may adopt preventive strategies such as contacting either by phone or email and creating counter-proposals or new offers for the already provided services. It should be noted that the adequacy of the strategies to the client's condition and context is fundamental to their success. Therefore, recognising why the customer is not paying will not only allow the creation of problem-solving strategies, but also the assessment of the cost-effectiveness of their application. Given the fact that these approaches consume a considerable amount of time and monetary resources, client selection should be highly accurate.

One of the main approaches for solving this type of problems consists of the implementation of a dunning process. Through this process, the targeted clients will receive one notification in one day, a text message and, eventually, an email or legal message in another day. Overall, this technique ensures the payment by the majority of the customers who were postponing it for no apparent reason, due to forgetfulness. Following the application of the dunning rules - acting as a funnel - we are left with considerably fewer but more complex customers. These include *i)* those who are unsatisfied with problems related to the service provided and who refuse to pay until their resolution, *ii)* those who have already migrated to another service and also *iii)* new clients who subscribed the service with the objective of enjoying free services for as long as possible without payment. Contacting all customers who are still in the process of non-payment is one of the best and most effective measures to be implemented by many companies. Despite

the apparent simplicity, this strategy proves to be one of the most costly for the company, as it involves the allocation of a large number of human resources.

Considering all the aforementioned information, we propose and implement a solution that aims to increase the effectiveness of this contact, where the main problem concerns the appropriate selection of customers to contact each day. Due to the fact that such communication will be carried out by a team of professionals with limited time, only a certain percentage of these clients will be contacted. As such, our goal is to maximise the number of clients who return to a regularised situation. For such purpose, we consider that non-paying clients who are not contacted will, probably, become more problematic and complex over time. For the prediction task, we are faced with a low number of customers who will not pay when compared to those that will actually do it, thus introducing an imbalanced domain problem.

1.2 Motivation

Today, the Portuguese reality faces a saturated and highly competitive telecommunications market, offering consumers a wide range of options. However, the quality of the services provided, the trust conveyed by the company and, ultimately, the price-quality ratio itself, seem to be the major factors taken into consideration when choosing a service. In general, in addition to excellent marketing and promotional campaigns, such companies should be able to devise effective strategies to avoid non-payment, cancellation of services and consequent abandonment by their customers.

According to Lu (2002) [31], we find that the annual churn rate still shows significant values, in the order of 30-35%, being the cost of acquiring a new customer considerably higher than maintaining an already established one. Thus, the importance of non-payment and abandonment minimisation strategies is highlighted [31].

During the last few years, we have witnessed a significant change in corporate mentality, which has encouraged the collection of different types of data and information by telecom companies. One way of preserving costumers may consist of using the information collected about them in order to increase their satisfaction with the services provided (i.e.: predicting whether a specific client will need a different type of service approach or if he is planning to switch to a different company). The data collected may also allow the prediction on whether the customer will incur in the non-payment of an invoice. Through data mining, it is possible to transform big sets of data into useful and representative information, allowing, in the context of telecommunication companies, to optimise the services provided and ensure the satisfaction of the main link of the chain - the consumer.

One of our goals is to use this data to get a lower rate of non-paying customers. To reach this goal, some data mining techniques were applied. At an early stage, customers should be differentiated into distinct groups according to their payment profile. Based on the information previously obtained, it is possible to predict whether or not the payment will occur.

With the present work, we intend to contribute in a pertinent way to the area of the payment default prediction models. We intend not only to determine the best groups of variables to adopt, but also to understand the benefits of applying strategies designed for addressing imbalanced domain problems. Therefore, we consider that the methodology considered in this case study can be extrapolated to several contexts that deal with the non-paying customer on a routine basis.

1.3 Organisation

This thesis is organised in five chapters as follows.

- Chapter 2 presents the main theoretical concepts fundamental to the project elaboration, namely regarding the scope of Data Mining, Machine Learning, and Imbalanced Domains, as well as a brief review about similar work already performed by other authors.
- Chapter 3 focuses on the problem itself, namely the methodology used for its resolution, the datasets obtained and the pre-processing methods applied.
- Chapter 4 highlights the experimental setup, the applied learning algorithms, the main results obtained and their discussion.
- Chapter 5 presents the major conclusions drawn from the executed work, as well as its contributions to the telecommunications sector and potential future work to be developed.

Chapter 2

Literature Review

In this chapter, a systematic review of the main theoretical concepts and definitions necessary for a proper understanding of the subject is presented, such as data mining, machine learning, and imbalanced domains, as well as an overview of similar and related work already performed by other authors in the field of payment default.

2.1 Data Mining

Data collection has been increasing exponentially at a global level. One of the major problems related to this practice is the constant need to store and process information. In order to process these data into a useful subject, it is possible to adopt an approach similar to the "Knowledge Discovery in Databases" (KDD) process, described by Fayyad et al. (1996) [19]. According to this author, knowledge seems to be the end product of a phenomenon of discovery motivated by concrete, useful and valid data. Data Mining, i.e. the application of algorithms for the extraction of patterns from a set of data, is a particular step in this process. Thus, in this process, the data constitute a set of facts, whereas the patterns consist of expressions that allow describing the subsets of data.

Being a process, KDD involves several interactive and iterative steps to get better results, so the need to iterate will almost always be imposed (cf. Figure 2.1). These steps are responsible for the selection, preprocessing, subsampling and transformation of a database to apply data mining methods afterwards. A brief and synthetic review of the various steps inherent in KDD follows.

1. Establish the application domain and identify the KDD goal according to the customer's point of view (prior knowledge).
2. Definition of a data set (or a subset of variables) to analyse.
3. Data filtering and preprocessing, in order to remove confounding factors, determining necessary but missing information.

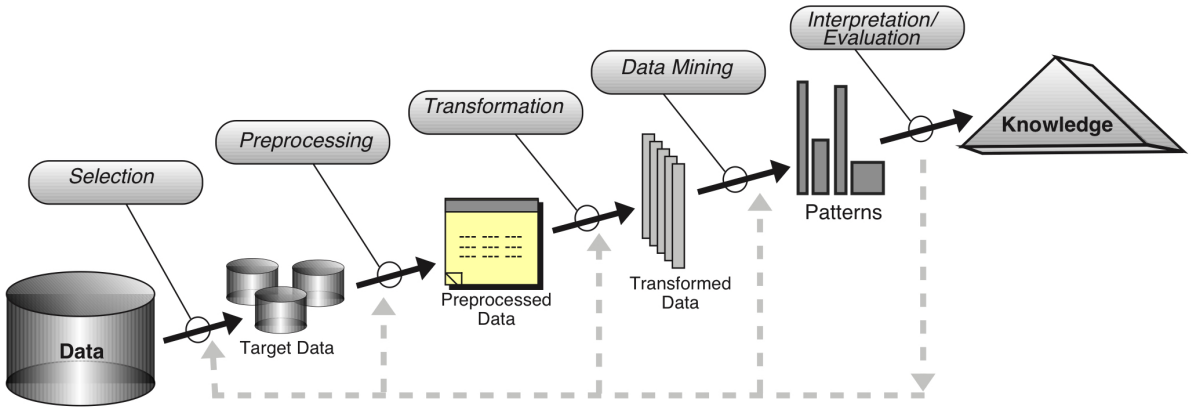


Figure 2.1: An overview of the steps comprising the KDD process [19]

4. Reduction and projection of data through the definition of features for data representation (it is particularly useful to define methods to reduce the effective number of variables associated with the data).

5. Relating the objectives inherent to the KDD process with a specific data mining method (e.g.: summarisation, classification, regression, clustering, etc).

6. Selecting the most appropriate data mining algorithm.

7. Apply learning methods, in order to search for accurate patterns.

8. Visualisation and evaluation of the obtained patterns.

9. Application of the obtained knowledge in order to improve the system in question.

While taking into consideration that we are focusing on getting important important knowledge from big (real-world) data, it seems to be important to clarify the Data Mining step. The process of knowledge acquisition can be carried out with the two following major objectives [19].

1. **Verification**: where the system is responsible for verifying the validity of the assumption-s/hypotheses previously created by the user.

2. **Discovery**: based on the system's capability to identify new and previously unknown patterns. It should be noted that this objective can be further subdivided into: (i) prediction (detection of patterns capable of predicting future behaviour) and (ii) description (discovery of patterns in order to make the system relatively intuitive from the user's perspective).

The main data mining methods that can be applied to fulfil the prediction and/or the description objectives are: **classification**, **regression** and **clustering** [19]. The following section will present some of them.

2.2 Machine Learning

Whilst data mining mainly focuses on the extraction of knowledge from information via statistical methods, machine learning commonly uses statistical methods in predictive tasks.

For the present work, the domain of greatest interest seems to be the area of **machine learning**, responsible for enabling computers to learn using sets of data.

In practical terms, machine learning falls under the scope of artificial intelligence and considers that systems and machines are capable of learning, recognising patterns, perceiving environments and making decisions based on a data set (i.e., training data). According to Gama et al. (2015) [22], machine learning algorithms used in this type of tasks aims to learn models or hypotheses through a training data set.

The most common learning tasks in Machine Learning are:

- **Supervised Learning**: This type of algorithm is responsible for creating reasonable generalisations for new situations, based on the previous analysis of the training data. In this way, the ideal result is one that allows to correctly determine the target variable for new, unseen and unlabelled examples. The goal is to approximately the unknown function $Y = f(X_1, X_2, \ldots, X_p)$, where Y is the target variable, $X_1, X_2, \ldots, X_p$ are features describing the attributes of each example and f() is the unknown function we want to approximate. The target variable is nominal for classification and continuous for regression. In order to obtain an approximation of this unknown function we use a data set with examples of the function mapping (known as a training set), i.e. $D = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n}$, where $\mathbf{x}_i$ is the features vector that describes the example i [19, 24].
- **Unsupervised Learning**: its main objective is to group similar situations, even if its representation is not known (by the absence of a target value on the training data), requiring only input attributes [19, 24].
- **Semi-supervised Learning**: machine learning task where both labelled and unlabelled data is used. This task emerged after the realisation that the combined use of these two types of data allowed to improve the accuracy of learning. So, we find that the labelled data is used to learn class models, while the unlabelled data allows the improvement of boundaries between the various classes (inevitably, we find out the correct labels for the previously unclassified data) [19, 24].

In supervised learning, we are faced with the following learning methods:

Classification consists of learning a model that allows to describe and distinguish different data classes. The model used (classifier) helps to predict categorical labels (e.g.: "safe" or "risky") in a two-step process. The first phase concerns the learning step, where a classifier is defined based on a training set. During the second step (classification step), the model is used to classify

new data. However, for this to be applied to the unclassified data, the predictive accuracy (percentage of correct test set examples that are correctly classified by the model developed) of the model must be previously estimated, using a set of data different from that of the training phase (the so-called test set). In the case of acceptable accuracy, the classifier may be used to classify future unlabelled data. One of the basic techniques of data classification is the decision tree induction, that allows labelling previously unlabelled examples by testing it against the tree [19, 24].

Regression is the statistical methodology most commonly used for numerical value prediction. Regression begins with a dataset in which target values are known. The regression algorithm estimates the target value as a function of the various predictors for each case. After summarising such relationships into a final model, it can be applied to unseen cases [19, 24].

It should be noted that any learning task should be preceded by a pipeline of data cleaning and pre processing.

One of the most commonly used methods in unsupervised learning is **Clustering** which consists in grouping/partitioning of sets of data in the same clusters (i.e., groups), so that the objects within a cluster have high levels of similarity but, on the other hand, are very dissimilar when compared with objects among other clusters. Similarity levels can be determined by taking into account the value of the attributes that describe the objects, often involving distance measurements within the cluster itself, these levels can be asserted with the silhouette coefficient. In business intelligence, clustering analysis allows the organisation of a large number of clients into several groups. For this purpose it takes into consideration common characteristics to implement more effective business strategies [27].

2.2.1 Generalised Linear Models

Generalised linear models can be represented by the function $Y = X\beta + \epsilon$, where X corresponds to the matrix of predictor/explanatory variables, β to a vector of parameters and ϵ to the residuals of the model. Thus, it can be noted that the present model combines both systematic and random components [35], [42]. Through the presented mathematical expression, it is verified that such models are characterised by the following structure [35], [42]:

1. random component - considering the vector of explanatory variables, the response variables are conditionally independent and their distribution belongs to the exponential family;
2. structural or systematic component - referring to a linear combination of explanatory variables;
3. linking function - function responsible for relating the expected value to the vector of explanatory variables.

Also according to Turkman and Silva (2000) [42], when modelling data through Generalised

Linear Models (GLM) we must fulfil three fundamental steps: (i) formulation of the models (i.e., choice of the distribution for the response variable, choice of covariates and appropriate formulation of the specification matrix and choice of the linking function); (ii) model adjustment (i.e., estimation of model parameters) and (iii) selection and validation of the models (i.e., find submodels with a moderate number of parameters that are still appropriate to the data, detect discrepancies between data and predicted values and check for significant outliers) [42].

In the following subsection, the logistic regression model is discussed in a more detailed manner, given its importance to the development of this project.

2.2.2 Logistic Regression

The logistic model emerged, among many other models, to allow the analysis of problems in which there are several independent variables that determine a binary/dichotomous outcome variables (i.e., there are only two possible outcomes). According to Dobson (1990) [17], this model is used whenever the outcome variable is measured on a binary scale (i.e., two categories, which we may refer generically to as "positive" and "negative") [1].

Thus, in such models, the dependent variable (Y) will acquire the value 1 if its result is "success" or, in turn, the value 0 if the outcome is "failed". However, in order to force the value of Y to vary between 0 and 1, it is necessary to resort to the logit transformation. Graphically, logistic regression appears to consist of a nonlinear transformation of linear regression itself, and is, therefore, an S-shape distribution function (cf. Figure 2.2).

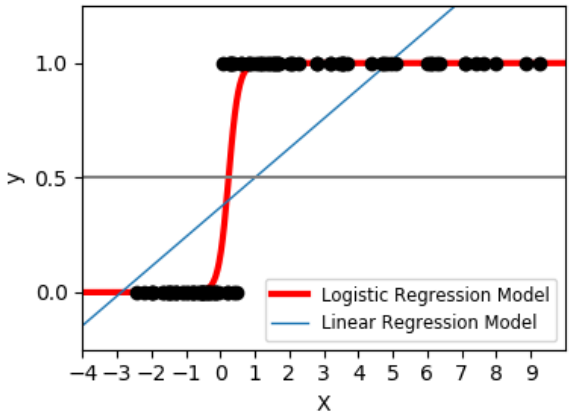


Figure 2.2: Logistic Regression example [39]

2.2.3 Decision Trees

Conceptually, decision trees consist of learning algorithms capable of addressing both classification and regression problems. Its hierarchical structure follows a strategy commonly referred to as "divide and conquer" by making recursive partitioning of the data guided by a preference criterion.

Through this process an heterogeneous and usually large population is divided into smaller and more homogeneous groups against a given target variable [20].

In general, decision trees originate from a root node and are partitioned, in a recursive manner, into several other nodes that correspond to conditions or predictions, and imply testing a given attribute (i.e., comparing the value of the attribute with a constant). Each decision tree will culminate in terminal nodes (i.e., leaf nodes) consisting of data labels (cf. Figure 2.3). Such leaf nodes provide a value response that is associated to all instances that reach the leaf.

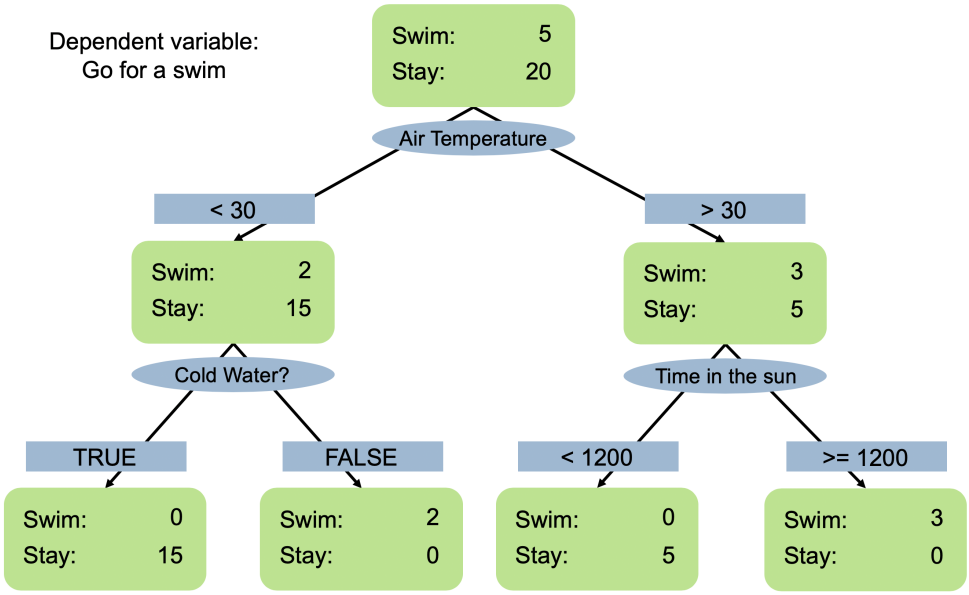


Figure 2.3: Decision Tree Example

The prediction ability of decision trees depends on the training data used by the algorithms that grow the tree. The training algorithm involves two fundamental steps: (i) growing phase and (ii) pruning phase (optional). In the **growing phase** the decision tree is built from top to bottom, with the root node corresponding to the entire database.

According to the target variable to be predicted, decision trees can be differentiated into: (i) classification trees and (ii) regression trees. Whereas in classification trees the response variable is categorical in nature (i.e., the main goal is to assign a particular class to each instance), in regression trees the response variable is continuous (i.e., the main objective is to assign a numeric value to each instance).

One of the main and most popular decision tree algorithm concerns the **CART (*Classification and Regression Trees*) algorithm**. Fundamentally, this algorithm generates binary trees where the splitting criteria is used to partition the dataset into two subsets. Posteriorly, this procedure is recursively applied to each subset until some condition determines its termination. As the trees obtained after the growing phase can be excessively complex, the **pruning phase** may be of utmost importance as it minimises overfitting issues. In this situation, each node produced by the tree is evaluated by the pruning criteria, and whenever each node meets the defined criteria, the associated sub-tree is deleted (i.e., the referred node becomes a leaf node) [10].

This algorithm emphasises successive splitting/pruning in order to increase the purity of the nodes. Two indices (Gini and Cross-Entropy) are used as purity criteria in classification problems and, according to Breiman et al. (1984) [10], a pure node is one in which all its instances belong to the same class (i.e., coherent subsets). From a practical point of view, the repeated pruning process allows the identification of the ideal subtrees, excluding branches that confer a low predictive power to each leaf. Validating the best subtree is a critical step, as we intend to choose the one that performs classification tasks with the lowest error rate possible [5].

2.2.4 Ensemble Methods

According to Bühlmann (2012) [12], ensemble methods are intended to improve the predictive performance of a given statistical learning or model fitting technique. In order to improve predictability, the general principle of this method is to draw up a predictive model based on the linear combination of several other submodels. In general this culminates with the generation of a model with more accurate and reliable estimates or results.

We are currently faced with two major ensemble methods - Bagging and Boosting - each with its own particularities and points of interest. In general, it should be noted that bagging procedures have as their main objective the reduction of variance while boosting methods are intended to reduce the bias of the procedure [12].

2.2.4.1 Bagging

Also known as *Bootstrap Aggregating*, the bagging technique was first introduced by Breiman (1996) [7] as an ensemble method that aims to improve the predictive performance of models mainly via the reduction of bias [12].

Conceptually, it should be noted that bootstrap ia a widely used tool. It is typically used to estimate the uncertainty associated with a learning method (e.g., estimation of standard errors of the coefficients from a linear regression fit) [28]. The power of this tool lies in the fact that it can easily be applied to a wide range of statistical learning methods, including those where a proper measure of variability seems to be extremely difficult [28].

According to Bühlmann et al.(2002) [11], this method is one of the most effective procedures for improving unstable estimators/classifiers, with a particular interest in problems with high dimension datasets [11]. In practice, a simple and natural way to reduce bias is to define multiple training samples from the dataset, build a prediction model using each of these samples, and average the resulting predictions [28].

Although bagging has a positive impact on various regression methods, it is particularly useful for decision trees, as they suffer from high variance (i.e., even small perturbations on the training set may induce considerable variations in predictions) [28], [20].

2.2.4.2 Random Forest

The Random Forest [8] corresponds to an ensemble method based on bagging techniques, presenting some improvements when compared to bagged trees. According to Fratello and Tagliaferri (2018) [20], in this method, a large set of independent and unstable classifiers, each created using only a sub-sample of variables and cases, are aggregated to produce a more accurate classification against a single model [20].

As with the bagging method, random forest also produces a number of decision trees, although some differences are evident, namely in the way the node division is performed. Thus, whenever a division is considered, the algorithm is not allowed to use the majority of the predictors (p) available, and a random sample of m predictors is extracted from the complete set as split candidates. At each split, only one sample of m predictors is used, indicating that the number of predictors considered in each split will be approximately equal to the square root of all predictors [28].

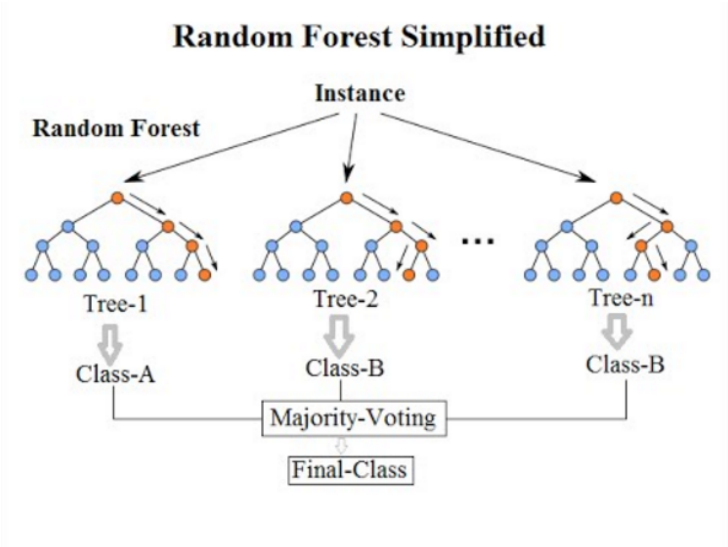


Figure 2.4: Random Forest example, a Bagging learning algorithm example where all the trees vote, with the same weight, on the final prediction [26]

When compared to the bagging technique, the random forest decreases the variance more sharply due to a decorrelation process that, by decreasing the correlation of the generated trees, increases the reliability of the learning method [28].

2.2.4.3 Boosting

Boosting [21] refers to an ensemble method that aims to improve the predictive capabilities of learning algorithms by minimising biases associated with models. Like bagging, boosting can also be applied to a wide variety of learning methods, whether classification or regression.

According to Kumar et al. (2013) [30], this algorithm develops robust learning process by

combining weaker algorithms (i.e., generates a strong classifier based on the combination of several weaker classifiers) (cf. Figure 2.5) [30].

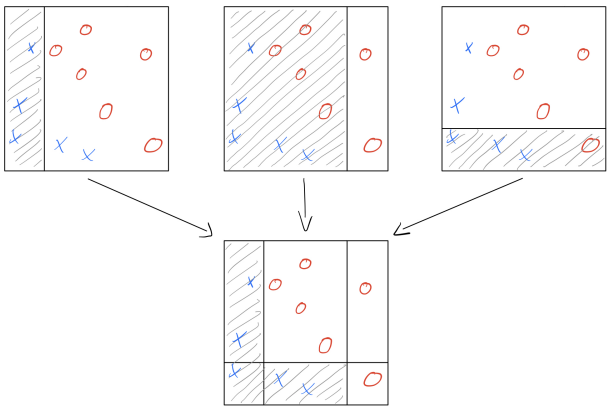


Figure 2.5: Example of the final output of boosting using weak learners (Decision Stumps)

Despite the similarity mentioned above, it should be noted that, contrary to the bagging method (where the construction of decision trees is an independent process), in the boosting technique tree generation is a sequential work (i.e., the growth of each tree depends on information obtained from previous trees) [28]. In practical terms, we find that in this method different weights are assigned to each of the training tuples. In this way, a series of k classifiers is learned iteratively [24]. Once a given classifier has been learned, the assigned weights are then updated to allow the next classifier to focus its attention on previously misclassified tuples [24]. Note that whenever a training tuple is incorrectly classified, its weight is increased, and the opposite is true with tuples that are correctly classified (i.e., the weight assigned to each tuple reflects the difficulty of its classification process). These weights will later be used to generate the training samples for the next classifier. The final output is the weighted (measure of their accuracy w.r.t. the training data) average of each of the previous classifiers' predictions [24].

Extreme Gradient Boosting (XGBoost):

Gradient Boosting is commonly applied to tree-based learning algorithms. The idea is based on an iterative optimisation approach. The objective at each round is to improve the prediction of the sub-models by applying a gradient-based update of case selection probabilities.

XGBoost is an improvement of the Gradient Boosting algorithm in terms of performance making it faster by implementing ideas such as parallelisation and cache optimisation, making it a great algorithm for scalability [14].

2.2.5 Model Evaluation

The evaluation of predictive models is a fundamental step for their subsequent implementation, as it allows the minimisation of future errors. This assessment task can be performed using several metrics as described below.

2.2.5.1 Classification Metrics

The definition of the various metrics for the model evaluation can be done through a confusion matrix (i.e., a tool that summarises the various measures utilised to evaluate the created models). Therefore, it can be considered a very useful tool for analysing the predictive performance of classifiers. Practically, considering a scenario with m classes (where $m \geq 2$), the confusion matrix consists of a table with a size of at least m per m [24].

- **True Positives (*TP*)** Correspond to a "positive" class that was correctly classified as such by the classifier model [24].
- **True Negatives (*TN*)**: Correspond to a "negative" class that was correctly classified as such by the classifier model [24].
- **False Positives (*FP*)**: Correspond to "negative" classes that were incorrectly classified as "positive" [24].
- **False Negatives (*FN*)**: Correspond to "positive" classes that were incorrectly classified as "negative" [24].
- **Accuracy**: Corresponds to the percentage of observations, in the entire test set, that are correctly classified by the model. It may also be referred to, in other literary areas, as the overall *recognition rate*, as it reflects the classifier's ability to recognise the various classes. From a mathematical point of view, this metric can be represented as follows: $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ [24].
- **Precision**: As a measure of exactness, precision gives us the percentage of "positive" observations truly labelled as such. From a mathematical point of view, this metric can be represented as follows: $precision = \frac{TP}{TP+FP}$ [24].
- **Sensitivity, Recall or True Positive Rate (TPR)**: As a measure of completeness, recall tells us the percentage of "positive" observations that are classified as such within all the existing classes. That is, sensitivity assesses the ability of a model to properly identify a given condition when that condition is indeed present (*true positive rate*). From a mathematical point of view, this metric can be represented as follows: $recall = \frac{TP}{TP+FN}$ [24].
- **Specificity**: Corresponds to the *true negative rate* (i.e., the percentage of "negative" observations that are correctly identified as such). From a mathematical point of view, this metric can be represented as follows: $specificity = \frac{TN}{TN+FP}$ [24].
- **False Positive Rate (*FPR*)**: Corresponds to the ratio between the false positives and the total of negative cases. From a mathematical point of view, this metric can be represented as follows: $FPR = \frac{FP}{TN+FP}$ [24].
- **F-Score**: This metric takes both precision and recall into account to better evaluate the model. Accordingly, the use of both metrics is based on a trade-off: the value of one may

increase at the expense of reducing the other. In general, the F-score can be defined as the harmonic mean between precision and recall with a β factor, which can give a distinct weight to precision or recall. From a mathematical point of view, the F-score can be represented as follows: $F - score = \frac{(1+\beta^2)\times precision\times recall}{\beta^2\times precision+recall}$ [24]. It should be noted that in the present work we have essentially used the F1-score this is, $\beta = 1$ giving a similar weight to each of these metrics. From a mathematical point of view, this metric can be represented as follows: $F1 - score = \frac{2\times precision\times recall}{precision+recall}$ [24], [38]. Whenever we mention the F-score metric throughout this thesis, we are referring to F1-score.

- **Receiver operating characteristic curves (ROC curves)**: The ROC curve plots the values of TPR on the y-axis and FPR on the x-axis over different threshold levels for the class probability. This curve allows us to compare the ability of a model to distinguish positive cases from negative cases, at the rate of negative cases being misclassified as positive.

- **AUC**: The AUC is the area under the Receiver Operating Characteristic curve. The AUC helps us to compare the performance of the models by measuring the area under its ROC curves. This metric tells us how much better a model is capable of separating the two classes [24].

- **First Decile LIFT**: it allows us to verify how many times our first decile of positive cases are superior to a random guess based on the class priors of the entire sample. The LIFT on the first decile is calculated by means of the following steps: (i) descending ordering of cases according to the predicted probability outputted by the model, (ii) distribution of cases by 10 group, with the first one representing the cases with the highest probability, (iii) calculation of the ratio between cases with positive target class and the total number of cases within the first group and (iv) calculation of base hit rate, that is the ratio between cases with positive target class and the total number of cases within the entire base. Therefore, the LIFT charts plot the TP against the subset required to obtain the number of true positives.

2.2.5.2 Variable Importance

When constructing a predictive model, we often come across the fact that not all variables have the same weight and degree of influence on the end result. Thus, we find that some explanatory variables will have a very significant influence on the response variable, while others will have a practically irrelevant contribution.

According to Zheng et al. (2017) [45] one of the most relevant metrics that allows us to evaluate the importance of several variables in the construction of the model is **Gain**. This metric translate as the increase in accuracy that a variable gives to each branch (i.e., improvement in the loss function).

Intuitively, it is found that the more a variable is used to create the model, the greater its

importance and, consequently its the gain, cover and frequency values [45].

2.2.6 Clustering Methods

As previously mentioned, clustering is one of the most popular unsupervised learning methods. The current literature presents us with several clustering methods that, although having some particular characteristics, may overlap. Overall, we come across four fundamental clustering methods, as follows: (i) partitioning methods, (ii) hierarchical methods, (iii) density-based methods and (iv) grid-based methods [24]. According to Han et al. (2011) [24], the main characteristics of such methods are summarised in Figure 2.6.

Method	General Characteristics
Partitioning methods	– Find mutually exclusive clusters of spherical shape – Distance-based – May use mean or medoid (etc.) to represent cluster center – Effective for small- to medium-size data sets
Hierarchical methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object "linkages"
Density-based methods	– Can find arbitrarily shaped clusters – Clusters are dense regions of objects in space that are separated by low-density regions – Cluster density: Each point must have a minimum number of points within its "neighborhood" – May filter out outliers
Grid-based methods	– Use a multiresolution grid data structure – Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

Figure 2.6: Main methods of clustering [24]

In the context of this thesis, the partitioning algorithms have been of major interest. Therefore they will be discussed in more detail below.

2.2.6.1 Partitional Clustering Algorithms

Considering a data set (D) consisting of n elements, when applying partitional methods we aim to divide this dataset into k partitions, so that each partition represents a cluster defined by $k \leq n$ (i.e., division of the dataset by k groups consisting of at least one element) [24].

According to Han et al. (2011) [24], the majority of the partitioning methods are based on distance, suggesting that after creating an initial partition, it relies on *iterative relocation techniques* to optimise the division task [24]. These authors defined as good partition techniques those in which closely related objects are gathered in the same cluster [24].

In the context of the partitioning methods, it is important to emphasise the role of the heuristics *K-Means* and *K-Medoids* as a way of optimising the clustering task, allowing us to

obtain a local optimum.

K-Means:

Considering, once again, a dataset D consisting of n objects and partitioned into k distinct groups, it can then be defined as objective function for partition quality evaluation one that seeks high intracluster similarity and significant intercluster dissimilarity [24].

In order to optimise the partition task, the *K-Means* heuristic uses a centroid-based strategy. Conceptually, the centroid corresponds to the centre point of the cluster and can be defined by the mean of the various elements that compose the cluster in question [24].

In functional terms, the *K-Means* algorithm starts by defining the centroid as the mean of the various elements of the cluster and proceeds through the following steps: (i.) random selection of k objects in D, initially representing the centre of the cluster, (ii.) distribution of the remaining objects among the various clusters according to their similarity (inferred by the distance between the object and the cluster mean), (iii.) computation, for each cluster, of the new mean considering the objects assigned in the previous iteration, (iv.) redistribution of objects according to the new calculated means (i.e., definition of new centroids) and (v.) repetition of the iterations until clusters stabilisation (i.e., unchanged clusters through the various rounds) [24].

When using this heuristic it is important to take into consideration that the results obtained depend largely on the initial random selection of the clusters. Thus, to obtain satisfactory results, it seems convenient to run this algorithm several times using different centroids [24].

K-Medoids:

The *K-Medoids* heuristic emerged to address the high sensitivity of the *K-Means* algorithm for the detection of outliers (i.e., this affects the mean value, making inadvertent object assignment to the various clusters). Therefore, the modification to the *K-means* algorithm involves selecting a particular object to represent each cluster (i.e., medoid), instead of determining its mean value. In turn, the further distribution of the remaining objects will be based on their degree of similarity to the representative object [24].

The sphere of the *K-Medoids* technique comprises two major algorithms: (i.) Partitioning Around Medoids (PAM) and (ii.) Clustering LARge Applications (CLARA).

Regarding the **PAM** algorithm, we verify that it acts in an iterative and very similar manner to the *K-Means*. Initially, the representative objects of each cluster are randomly selected. However, task optimisation requires the repeated substitution of representative objects by others, until the quality of the clustering cannot be improved by any substitution [24]. In general, the overall quality of this algorithm can be measured by a cost function of the degree of dissimilarity between an object and the representative element of each cluster [24].

When compared to *K-Means*, we find that *K-Medoid* is a more robust method as its results appear to be less influenced by outliers. However, when applied to large datasets its computation

cost increases dramatically [24]. Therefore, to address the reduced scalability and utility of the PMA algorithm in large datasets, the **CLARA** method (i.e., sampling-based method) can be used [24]. In practical terms, this method starts by selecting a random sample from the dataset and then applying the PAM algorithm to determine the best medoid [24]. In this context it is also worth noting that, since CLARA builds clusters from random samples, its effectiveness will depend directly on the sample size (i.e., if an object that is one of the best k-medoids is not selected, then the CLARA method will not be capable of outputting the best clustering) [24].

2.2.6.2 Similarity Measures and Clustering Evaluation

After performing the clustering task, it is essential to evaluate the quality of the results generated by the method used. This assessment comprises two main assumptions: (i.) determine the optimal number of clusters, and (ii.) measure cluster quality. Given the most commonly used metrics in this paper, we focus our attention on steps (ii.) and (iii.) of the evaluation.

Determining the most appropriate number of clusters is both a crucial and complex task because it is necessary information to various algorithms (e.g., *K-Means*) [24]. On the **elbow method**, the assumption is that having more clusters reduces the variance within each cluster. This is because if we increase the number of clusters, we will have much more similar data objects in each group. However, the effect on variance reduction may be minimal if too many clusters are created. Therefore, an heuristic to determine the proper number of clusters can be implemented, by selecting the turning point in the curve [24].

The second fundamental step of clustering evaluation is to answer the following questions: "How good is the clustering generated by the method used and how can I compare these results with the clustering obtained by different methods?". Currently, we have at our disposal several evaluation methods, which can be categorised into two major groups: (i.) extrinsic methods and (ii.) intrinsic methods [24]. This differentiation was idealised according to the presence or absence of *ground truth* (i.e., ideal clustering built on human knowledge), respectively [24].

In this thesis, we focus predominantly on the **silhouette coefficient**, explained below. In general, this coefficient makes its evaluation by determining the separation and cohesion of clusters. Thus, for each object o belonging to a given cluster, it is calculated, not only the average distance between o and all the other elements of the cluster (defined by $a(o)$), but also the minimum distance from all other clusters (defined by $b(o)$) to which the object does not belong [24]. Then, it is verified that $a(o)$) represents the cluster's compactness (i.e., the smaller its value, the greater the compression), and $b(o)$) the degree of separation of this object from the other clusters, being the ideal result a cluster with high compactness and with o far from neighbouring clusters [24].

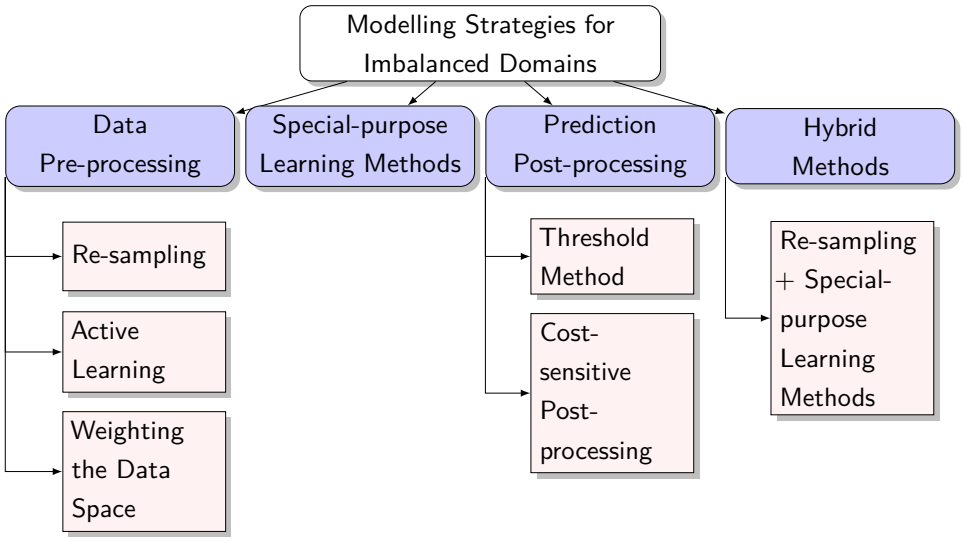


Figure 2.7: Main modelling strategies for imbalanced domain learning [6].

2.3 Imbalanced Domain Learning

The main goal of this thesis is to predict a set of a usually scarce and underrepresented class of cases within a domain of the target variable. Namely, we want to accurately predict the small percentage of customers that are not going to pay for a long period of time and, eventually, may churn. These less frequent cases in the target variable are those of greater importance to us. As previously mentioned, this constitutes a problem of imbalanced domain learning. Thus, it is crucial to identify the modelling strategies that exist for dealing with this type of problems [6].

Considering the previous assumption, several authors have verified that the application of both standard classification learning algorithms and standard evaluation metrics in imbalanced domains may compromise an optimal classification of the cases, leading to unsatisfactory results on the most important cases. The failure derives from the assumption that all the cases have equal importance. In such context, the performance on a set of rare cases will have a reduced impact on the overall performance, when compared with common cases [25, 43]. Thus, the selection of suitable learning strategies and evaluation metrics is fundamental for the success of the approach when learning with imbalanced data.

To date, four major learning strategies and categories have been defined in the context of imbalanced data distributions: (i) data pre-processing, (ii) special-purpose learning methods, (iii) prediction post-processing and (iv) hybrid methods. Figure 2.7 schematically represents the main modelling strategies and the different approaches within each category [6].

In this thesis, we focus on the use of pre-processing approaches, namely re-sampling strategies, and on the threshold-based post-processing approach.

In a simple way, pre-processing consists of adapting the data set available according to the

user's preferences and goals [6]. Within the re-sampling techniques, we can reduce data imbalance through two simple techniques: under and over-sampling. In under-sampling, only a selected set of cases from the majority class is used. In over-sampling, copies of cases from the minority class are added to the total set of data. Whenever the case selection occurs in a random fashion, these techniques are defined as Random Under-Sampling (RUS) and Random Over-Sampling (ROS). Nevertheless, it should be noted that these two techniques can lead to the elimination of useful cases and data overfitting, respectively. Still, in the scope of re-sampling, the generation of new synthetic data also allows dealing with imbalance problems with the added bonus of improving the generalisation capacity and reducing the risk of overfitting. These new cases can be obtained through two strategies: (i) interpolation of existing cases and (ii) introduction of perturbations [6]. The Synthetic Minority Over-sampling Technique (SMOTE) method proposed in [13] is a very interesting approach that uses interpolation of cases to generate new synthetic ones and that can be complemented with the RUS technique.

At the post-processing level, there are two main solutions: (i) threshold method and (ii) cost-sensitive post-processing. The threshold method consists of changing the class output of the models to the likelihood of belonging to that class. Having this value it is then possible to set a threshold that defines from which value an observation will belong to a class or another [6].

2.4 Related Work on Payment Default

Elaborating the concept of churn previously introduced, we verified that, according to Lu (2002) [31], the churn action presents a broad definition referring to the cancellation of the subscribed services acquired by the client. The act of churn can be initiated, either by the service provider or by the client itself. The action of involuntary churn, also known as service-provider initiated churn, is a concept frequently related to the problematic of the non-paying customer. Whenever a client does not pay until a pre-determined date (referring to it as payment default), the company will react to that by disconnecting the services previously supplied [31].

Several methodologies for payment default prediction abound in the existing literature. A practical example goes back to the work of Beaver (1966) [4] and Altman (1968) [2], concerning the development of univariate and multivariate models, respectively, to predict business failures and default based on a set of financial ratios. Beaver resorted to a dichotomous classification strategy, that is, classifying firms based on their financial ratios as failed or non-failed for his prediction. Altman developed a multiple discriminant analysis technique (MDA), considered to be far superior when compared with Beaver's approach, due to the fact that it takes into consideration several other variables, therefore providing a best overall prediction of business failure. Given the superiority of such an approach, it easily became the most applied statistical technique in default prediction models and was later used by multiple authors, such as Gombola et al. (1987) [23] and Lussier (1995) [32]. After several applications of the MDA technique it was possible to recognise its limitations when applied to the default prediction [3].

The work carried out by Ohlson (1980) [36] is considered the main responsible for demonstrating the advantages of applying logistic regression in problems of default prediction. From a theoretical point of view, logistic regression seems to have the appropriate characteristics for default prediction problems, recognising the importance of each of the variables used for prediction through the analysis and interpretation of different coefficients [3, 36].

Furthermore, according to Crook et al. (2007) [16], the standard approach to estimate the default probability is the logistic regression model. However, some alternatives to this type of model do exist and can also be applied, such as machine learning approaches, as in the work developed by Kruppa et al. (Kruppa et al.) Kruppa et al., responsible for applying such approaches to estimate the required default probability in a large data set of short-term instalment credits [29].

In addition, it is still of particular interest to highlight the work of 2014 (2014) 2014 in the context of solving imbalanced domain problems, in order to improve the performance of prediction models of churn occurrence. In their project, the authors used a churn database from an Indonesian telecommunications company, which was pre-processed through the combination of sampling techniques (namely simple under-sampling and SMOTE) and Weight Random Forest (WRF). After comparing the performance of the prediction models developed in three different contexts, i.e.: (i.) WRF without sampling techniques, (ii.) WRF with a single sampling technique and (iii.) WRF with combined sampling strategies, the authors concluded that, when isolated, the WRF is associated with a low performance, while after the addition of any sampling technique, the model's performance increases significantly. In general, the best F-Score was verified for the methodology initially proposed, that is, WRF coupled to combined sampling strategies, also allowing to reduce the computational cost associated with the implementation of the model. Thus, it was possible to conclude that sampling techniques are crucial in the aid of WRF algorithms, in order to improve the performance of predictive models, and that models with reasonable performances and classifiers with high accuracy and top-decile, allow determining the best strategies to deal with churn [18].

On the other hand, the work carried out by Coussement and Van den Poel (Coussement and Van den Poel) Coussement and Van den Poel allowed demonstrating the effectiveness of the support vector machines (SVMs) approach in the detection of churn in subscription-based services. On the other hand, since they are based on principles of risk minimisation, it favours their performance when applied to noisy databases. In order to demonstrate the true effectiveness of this approach, the authors compared it with the predictive performance provided by logistic regression and random forest, verifying that SVMs are superior to logistic regression only when an appropriate method of parameter-selection is applied. Thus, this study showed that non-traditional methods, when appropriate, can also be applied in churn prediction tasks, with comparable and equally satisfactory results Coussement and Van den Poel.

Consideration should also be given to the contributions of Mishra and Reddy (Mishra and Reddy) Mishra and Reddy in the field of churn prediction, by using deep learning. In this particular project, Convolutional Neuronal Networks (CNN) were implemented, and their

accuracy for predicting churn events was evaluated. According to the authors, the application of CNN in prediction tasks seems to be based on four major steps: (i.) *convolution* (i.e., extraction of the ideal features, preserving dependency between input classes and variables), (ii.) *non-linearity* (i.e., application of functions to connect input variables to model layers), (iii.) *pooling* (i.e., dimensional reduction of the introduced feature pool to train the model quickly) and (iv.) *classification*. From the experimental results point of view, the authors found that neural networks are a better classifier for the involuntary churn prediction problem. This evidence was supported by the results obtained in all model performance evaluation measures, namely accuracy, error rate, precision, recall and F-score. Therefore, it is admitted the existence of a new valid tool for solving problems of this nature.

Considering all the information reviewed and mentioned above, it is of particular importance to mention that our approach to this problem will consist not only on testing the logistic regression model but also comparing the results with tree-based models, such as CART [9], Random Forests [8] and XGBoost [14]. As we are also dealing with an imbalanced domain, we are going to apply not only re-sampling methods in order to change the target variable distribution, but also analyse the combination of such methods with post-processing approaches, such as the threshold method.

The next chapter focuses predominantly on the task of payment default prediction, highlighting the problem definition, the methodology adopted and the defined dataset, being this last parameter subdivided into classification dataset and clustering dataset.

Chapter 3

Payment Default Prediction

This chapter presents a more detailed definition of the problem in question and its implications, as well as the methodology adopted to meet the proposed objectives. Additionally, it also explains the dataset used for both classification and clustering.

3.1 Problem Definition

The objective of our task is to answer the question: "Will the customer pay N days after the contact?". By not limiting our intervention only to the probability of involuntary churn, we were able to intercept different clients, namely: i) clients who will not pay, because they simply do not intend to do it leading to involuntary churn, ii) clients unsatisfied with the requested service and who will not pay until the resolution of their problems and also iii) clients dealing with financial instability and willing to accept payment agreements.

In order to respond to the previous question, we intend to create a model for payment default prediction, predicting not only if the payment will occur until the due date, but also if the client will pay at all before the involuntary churn. Based on this assumption, clients were classified into payers or non-payers, through a classification task in the form of $Y = f(X)$, where Y corresponds to our binary target variable and X to the set of features used for the prediction.

In addition to the first stated objective, a cluster analysis was also performed in order to label and group the different types of clients according to their payment behaviour. After this split, the classification task was performed to these new groups in order to understand if the overall predictive ability increases.

There are two types of payment default, when a client does not pay the invoice until the due date or when the customer does not pay until the company terminates the services (involuntary churn). In this thesis, we will focus on the latter. As a form of organisation, we set the invoice payment deadline as the "day zero" of the dunning days. In order to recover the clients, telecommunication companies have implemented several strategies. One of the most relevant

actions is the use of dunning rules, which correspond to a set of actions previously defined by the company and intended for all customers who have not yet paid the invoice for the month in question. Within the actions of this set, we find: sending a warning message on day one, performing a call on day thirty-five and leading to a disconnect action (i.e., involuntary churn) after several days.

During the first few days after the deadline, we may come across clients who have forgotten to make the payment. For such type of customers, the approach will be relatively simple since they are expected to regularise their situation. Thus, sending a notification message and waiting for payment is typically the best strategy, allowing to save human and monetary resources for cases of greater complexity. The major subset of clients to reach, the appropriate time for the formal telephone contact and the establishment of the prediction will only be determined after the aforementioned clients leave the so-called collections stage (the period after the due limit date). Intuitively, we intend to establish contact with all clients that have reached the stage of involuntary churn, taking into consideration the enormous diversity of existent customers. Despite their variety, we can coarsely divide them into two major groups: (i) those with simpler situations, easily solved with a telephone contact or by means of a payment agreement and (ii) those with high complexity and of difficult resolution, which require increased resource needs (i.e., clients with a high probability of involuntary churn). Therefore, by defining the following question as our objective - "Will the customer pay within N days after the established contact?" - it becomes possible for us to identify these two groups of clients. Figure 3.1 represents an illustrative example of a dunning rule applied in the context of our work.

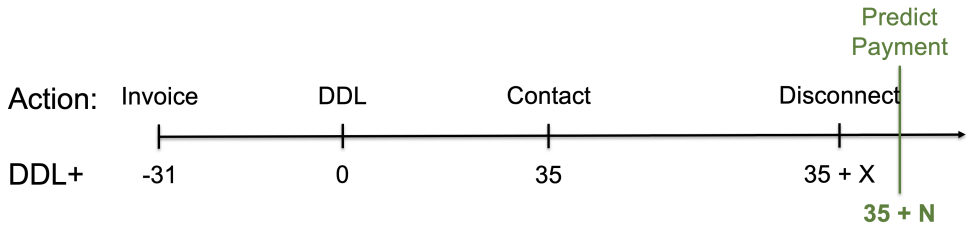


Figure 3.1: Dunning Rules without Payment Plan (PP), where the disconnect action happens X days after the contact (with X between a range of values depending on the behaviour of the customer) and the payment prediction happens N days after the Contact (with N being a fixed number for all customers)

Besides the groups of nonpaying customers, we have four types of clients previously defined by the company itself: i) clients with a payment agreement, ii) clients without payment agreement, iii) new clients and *iv)* old clients. Irrespective of the existence of a payment agreement, the clients have similar dunning rules. In general, the main difference between customers with and without payment agreement lies at the beginning of the set of rules (i.e., those with agreements initiate the interactions at an earlier stage). Thus, in order to include both subsets of clients, it is convenient to obtain similar rule behaviour. In these cases, a shift is performed in the dunning days, so that the two subgroups become aligned in the rule (cf. Figure 3.2). During this process, it is fundamental to take into account that our new customers do not have any historic data.

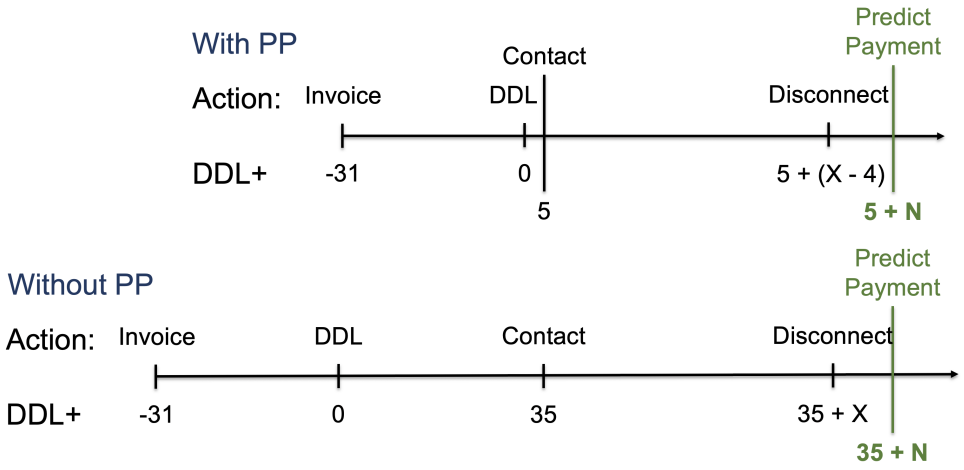


Figure 3.2: Dunning Rules with and without Payment Plan aligned by the day of the contact, where the disconnect action happens X days after the contact (with X between a range of values depending on the behaviour of the customer) and the payment prediction happens N days after the Contact (with N being a fixed number for all customers)

3.2 Proposed Approach

In order to accomplish the goals that were previously set up, the workflow was divided into the following steps:

1. Collection of information from the database provided by the company concerning the following data: dunning actions steps, invoices, payments, and client profile information. In order to perform a correct dunning analysis, it is fundamental to have information regarding a time frame of, at least, four months, because the action that defines the involuntary churn usually occurs on the day seventy.

2. Initial exploratory data analysis (EDA), that allowed the definition of different populations to addressed and the comparison, among each of them, of the number of clients and their distribution in the different dunning actions, the involuntary churn rate and the different factors that may affect this rate. After this, a new EDA was carried out where in order to analyse the target population. Where it was intended to verify how the population reacted to the different dunning steps, their distribution and how does the probability of involuntary churn changed over time.

3. Creation of the dataset for the classification model. In this step, we selected the variables observed in the EDA's and proceeded to aggregate the several variables from the historical months. As we were dealing with real data concerning payments from previous invoices, we already knew information such as the time of payment. Thus, it was essential to ensure the non-inclusion of this type of information, especially when it was posterior to the prediction month. The final dataset consists of customers in payment default on the day before the contact.

4. Definition of the first model to predict involuntary churn (in close collaboration with the team usually responsible for dealing with dunning situations), in order to get a significant amount of domain knowledge. This collaboration was performed in a way that allowed the creation of the model to be iterative where we would test new ideas, analyse the data and discuss the results improving the model in every iteration. It should be noted that the definition of our target was also performed in collaboration with this team.

5. Creation and validation of the model through the implementation of a pipeline for testing the different models, parameters and strategies that allowed us to deal with the imbalanced nature of the dataset.

6. Identification of the possible clusters that defined customer's payment behaviour. From an organisational point of view, it was defined that this clustering task would take into consideration only the clients linked to the company for at least 6 months, the established customers. Here, it was also necessary to return to the database in order to get the customers in this status and to validate that the data obtained was correct. New variables needed for this process were also created.

7. Test the several clusters with the silhouette metric. At this phase, it was also important to validate that the previously defined clusters defined were being created.

8. Confirmation of the obtained results with the dunning team, and eventual discussion about the need to create new approaches (e.g., new dunning rules individualised and adapted for each cluster).

9. Creation of the new models using the pipeline for the classification model on the populations outputted on each of the obtained clusters.

3.3 Dataset

To define the dataset, two different pipelines were developed in order to obtain the necessary data, both for the creation of the payment default prediction model and for the customers clustering task. As such, this section is divided into two parts: the first referring to the classification dataset and the second to the clustering dataset.

3.3.1 Classification Dataset

In order to respond to the classification problem, we have obtained anonymised data from a sample of clients that have entered the Collections Stage in a period of three consecutive months. Note that for this task, all types of customers have been considered, namely new and previously established customers, as well as customers with and without payment agreements this was due

to a necessity by the dunning team [1].

In order to obtain all the necessary data, the work process was divided into several steps, being the four major ones described as follows.

1. **Collection of information** from the database provided by the company, focusing on the following data: dunning actions steps, invoices, payments, and client profile information. In order to perform a proper dunning analysis, it is fundamental for us to obtain information regarding a forward interval of, at least, four months. This has to do with the fact that the action that defines involuntary churn occurs around that period of time, following the default. The dataset was composed of an additional seven months of historical data.

2. **Pre-processing the collected data**, as one of the crucial steps due to the fact that the information contained in the database was in constant updating. The daily rhythm of data updates means that customers who have previously entered dunning, the following day may no longer be there due to payment regularisation (thus, the information in the database regarding that particular customer will change from "unpaid" to "paid out"). In order to create a prediction model, it was necessary to collect information about customers that have entered dunning at least four months ago.

3. **Initial exploratory data analysis** that, as mentioned above, allowed the analysis of different groups of clients: recent clients, old clients, clients without payment agreement and clients with a payment agreement. At this particular stage, we have compared each of these groups regarding the number of clients and their respective distribution by the different dunning actions, the involuntary churn rate and the various factors likely to influence it. It was noted that, although the percentage of payment default appeared to be similar in old customers and those with payment agreements, this phase allowed us to verify the existence of a considerable difference in the probability of payment default among the following groups: new *versus* old customers and customers without payment agreement *versus* customers with a payment agreement. This overall analysis allowed us to attribute the highest risk of default to the most recent customers.

4. **New exploratory data analysis** after the selection of the target population. At this stage, an analysis on how the population in question reacted to the different dunning steps, as well as their distribution and the variation of involuntary churn probability over time was performed.

After obtaining a sample of customers, we prepared a dataset that gathered the following information:: i) historical data, from the previous 7 months, regarding payments, dunning actions, invoice details and type of subscription package and ii) data describing the month in debt, namely duration of the subscription in months, type of subscription, amount to pay on the last invoice

[1] The data used in this paper is proprietary. Understandably, its presentation and description are kept anonymous for commercial and business reasons.

and records about dunning actions. With this historical data, we created several variables that summarize several customer features that might change over the months, such as the behaviour of the payments and differences on the invoice.

The variables used can be divided into four distinct groups, such as: i) customer, ii) behaviour, iii) invoice and iv) payment details. Thus, we gathered a total of 83 variables, which were grouped as shown in Table 3.1.

Table 3.1: The set of features.

Group	**Features Examples**
Customer	- Age group - Duration of subscription - Type of subscription package - Direct Debit Activated Flag - Time elapsed, in months, since the first subscription
Behaviour	- Mean, median, maximum and standard deviation of days elapsed until the regularization of the monthly payment during the last months - Mean, median, maximum and standard deviation of the number of times the customer entered the dunning process during the last months - Maximum and current dunning step
Invoice	- Billed amount and value charged for additional services - Mean, median, maximum and standard deviation of the billed amount and the value charged for additional services during the last months - The difference between the current and previous invoices
Payments	- Mean, median, maximum and standard deviation of the amount paid by the client in the last months - The amount the customer has in debt - Mean, median, maximum and standard deviation of the customer's debt during the last months

The next step consisted in the study of variable correlation by means of the *Pearson Correlation Index*. It should be taken into consideration that the analysis of variables with a high degree of linear correlation is fundamental, since they may compromise the accuracy of our model. Since this is an imbalanced domain problem, whenever we decide to remove a variable with no apparent value for the model, we may be removing vital information. Therefore, when we come across two highly correlated variables (i.e., with a confidence level of 95%) that do not express useful information, one of them must be eliminated. On the other hand, as we are dealing with an imbalanced problem, there might be some variables that have additional useful information for the construction of the model in order to differentiate the under-represented and over-represented classes, the variables in question could be transformed in a single variable by, for example, calculating the difference between them.

After all the above mentioned pre-processing steps, we obtained a dataset composed by over one hundred thousand observations described by 63 attributes, regarding the information mentioned before aggregating the 7 months of information. From the set of all observations, the positive class on our target represents 20% of the customers.

3.3.2 Clustering Dataset

As we wanted to categorise the customers according to their payment behaviour, and because we knew that well behaved customers, when in dunning, are usually the most problematic, we intended to create the dataset with not only the customers that reach the DDL but also, the customers that pay before (i.e., the totality of customers at the time the invoice is emitted).

For this purpose, we gathered an anonymised sample of customers who already had, at least, six invoices emitted. Subsequently, we collected information about the selected population, to define the variables of most interest for the task at hand. In general, the data of interest consisted only in the aggregation of variables related to payment and invoices. The exclusive selection of this type of data, excluding other sources such as customer age range or steps in the dunning rule, allowed us to have a more reliable segregation of this customers in the scope of payment behaviours.

It should be taken into consideration that the time frame established for the data collection consisted of a period of six months. We have also limited the payment date so that it does not exceed 120 days (i.e., the date where we have around 97% of the customers with a paid invoice as shown on Figure 3.3), in order to ensure that no data leakage is being generated.

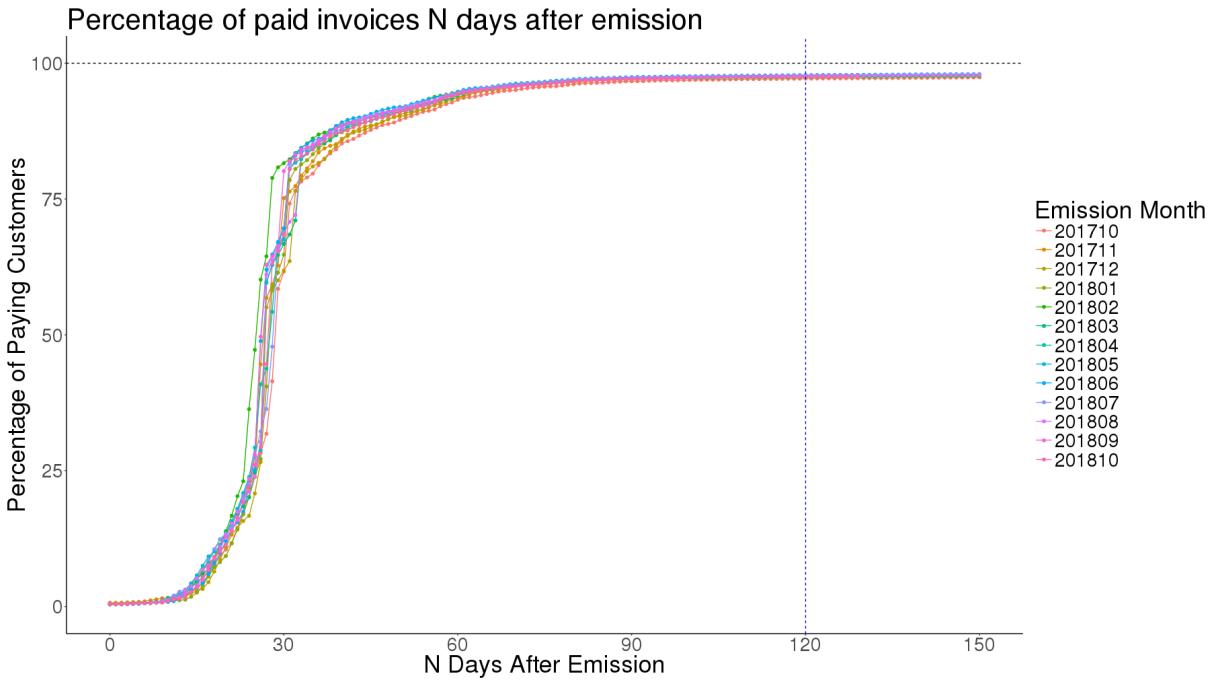


Figure 3.3: The percentage of paid invoices N days after emission for 13 consecutive months

Once collected the data considered necessary, we proceed to the aggregation of variables regarding the referred period, constructing a dataset with the features presented in table 3.2.

As in the classification dataset, a correlation analysis was also performed, excluding those variables with a high degree of correlation.

Table 3.2: Features for the cluster generation.

Group	**Features**
Customer	- Ratio of the number of months the customer had direct debit activated
Behaviour	- Mean and standard deviation of days elapsed until the regularisation of the monthly payment during the last six months - Mean and standard deviation of days elapsed until the regularisation of the monthly payment during the last six months when the customer pays before the Due Date - Mean and standard deviation of days elapsed from the Due Date until the regularisation of the monthly payment during the last six months when the customer pays after the Due Date - Mean and standard deviation of days between payments - Ratio of months the customer pays the invoice - Ratio of months the customer partially pays the invoice - Ratio of months the customer pays before the due Date - Ratio of months the customer pays in the 12 days after the due date

Considering that the population aggregation was done through a clustering algorithm, a simple min-max normalisation technique was applied to the mean and standard deviation of the variables.

Once the two datasets necessary to solve the proposed problem are established, we can now focus on the machine learning component inherent in this task, as described in the next chapter. In addition to the presentation of the learning algorithms utilised and their respective evaluation, we present and discuss the main results obtained. The datasets created, the historical months of data, the time to check if a payment occurred and the time of the model creation are described on Figure 3.4.

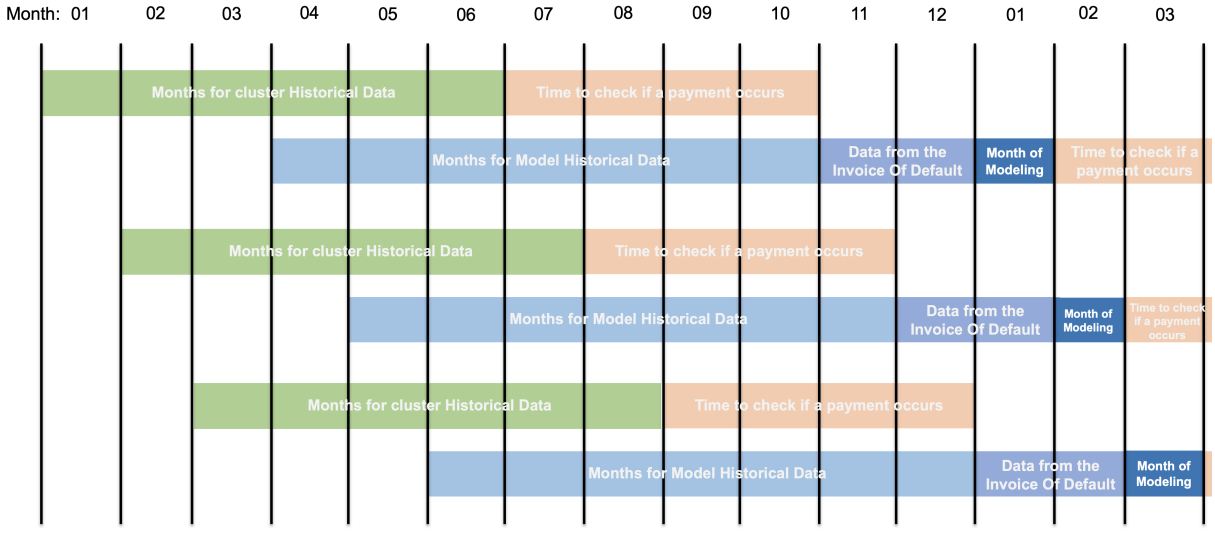


Figure 3.4: Representation of the timeline necessary for the data collection

Once the required datasets were defined, the next approach focused on the creation of the pipelines for the experimental setup and the learning algorithms and models. The next chapter highlights the experimental setup adopted, with particular focus on the learning algorithms used, the experimental methodology and the main results obtained (with and without clustering).

Chapter 4

Experimental Setup

This section describes the experimental evaluation carried out on the obtained datasets. Several different models were tested and compared for a considerable grid of parameters (specific to each model in question). Re-sampling strategies and the threshold method were applied, in order to determine the best approach to our problem. Also, we divided the population by their payment behaviour and applied the models and strategies to these subdivisions of the population. All our experiments were implemented in R language [37].

4.1 Learning Algorithms

The tested algorithms included logistic regression [1], CART [9], Random Forest [8] and XGBoost [14]. Table 4.1 presents the R packages and parameters tested for each learning algorithm.

Table 4.1: Models, respective packages and tested parameters.

Learning algorithm	R Package	Parameters
Logistic Regression	stats [37]	family = binomial(link = logit)
CART	rpart [40]	minsplit = {10, 20, 30}, cp = {0.01, 0.05}
Random Forest	ranger [44]	mtry = {5, 10}, ntree = {100, 250}
XGBoost	XGBoost [14]	eta = {0.05, 0.1}, max_depth = {3, 8}, nround = {100,200}, cst = {0.55, 0.85}

To tackle the imbalanced nature of the problem, we have chosen to use the threshold method and re-balance the two classes using the following methods: (i) random under-sampling the majority class (RUS), (ii) random over-sampling the minority class (ROS) and (ii) SMOTE [13] with under-sampling of the majority class (SMOTE+RUS). The selection of the model with the best parameters was achieved through a grid search using the R package performanceEstimation [41].

In each iteration on the grid search, we applied the threshold method via internal validation, where the best threshold was tested with the validation set of the training data and applied to the final predictions to derive the labels.

In order to subdivide the population into different groups, we used the clustering algorithm CLARA contained in the package "cluster" [33].

4.2 Experimental Methodology

For the decision of the best algorithm of the grid, and given the intrinsic temporal information in the data, we estimated the predictive performance by using growing window. Our intervention began by dividing the dataset by several weeks (a total of 15 weeks), considering the period of the initial 3 weeks for training and the subsequent 2 weeks for testing. Posteriorly, these 2 weeks of testing were added to the first 3 weeks of the train, constituting a new training set, now comprised of 5 weeks. Again, the 2 weeks following these 5 were considered for testing. This procedure was successively repeated until the end of the established 15-week period, having the last validation dataset the first 13 weeks for training and the last 2 for testing (cf. Figure 4.1).

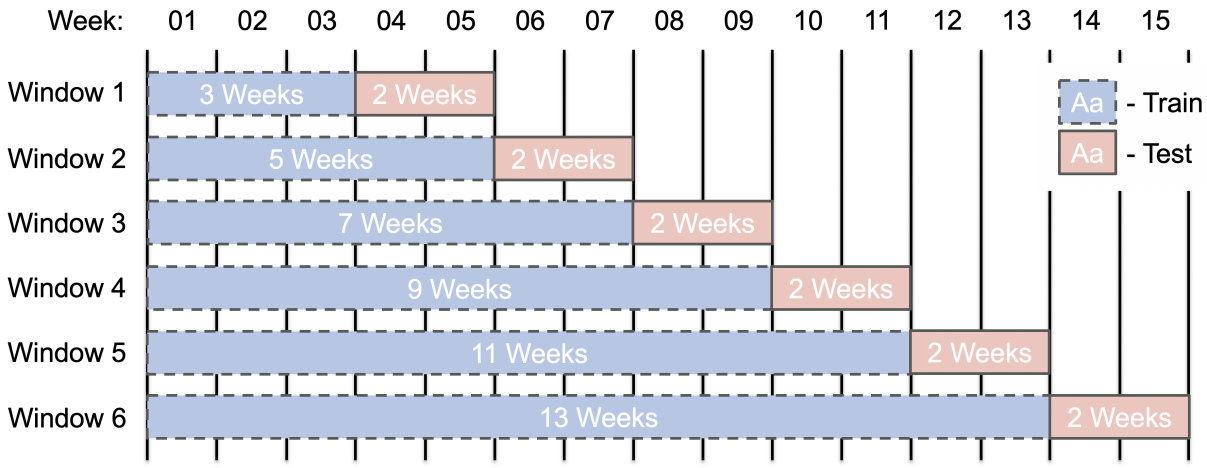


Figure 4.1: The weeks each split of test and train data had on the performed sliding window

All the tested methods were evaluated using the same training and test sets, thus ensuring a fair pairwise comparison of the estimates obtained, whilst maintaining the temporal coherence of the data. For each algorithm, we had a total of 6 datasets.

On Figure 4.2 we can see the pipeline utilised on the experiments. In a simplified way, for each of the grid parameters (i.e., each method plus algorithm combination), we created the six growing window datasets and acted as follows:

1. we started by testing the best threshold for that combination. For this task, we divided the train set using holdout; following this division, 70% of the train data was considered for training, while 30% were considered for testing;

2. afterwards, the new training data was re-sampled, and a model was created for this set;

3. then, a prediction for the new test sample was also performed. At this stage, we intended to obtain the results as probabilities so that we were capable of testing different thresholds

and check the F-score results. Intuitively, the best threshold was the one that achieved the best score;

4. once the threshold was defined, the original training set was re-sampled and the model established. Then, the predictions (in the form of probabilities) for the obtained test set were established, in order to calculate not only the F-score with and without the best threshold (our main focus) but also other metrics such as AUC and first decile LIFT, for a better understanding of the results.

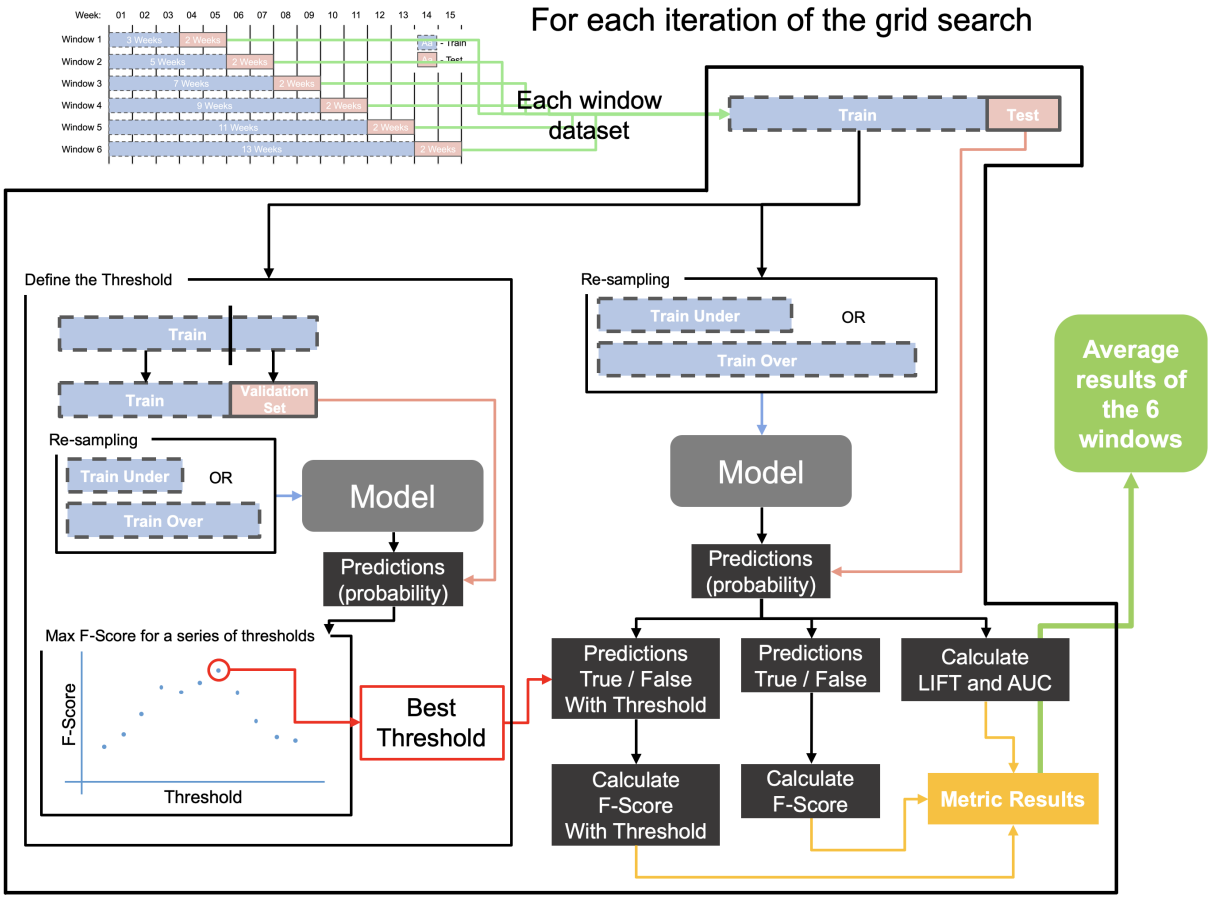


Figure 4.2: The pipeline of the experiments

After performing the two aforementioned steps, the final results were obtained by calculating the mean and standard deviation of the individual findings of each of the six datasets.

The division of the dataset used in the classification task was performed with the CLARA algorithm. In turn, to determine the optimal number of clusters, we resorted to both the silhouette and the elbow methods. After obtaining the new datasets, the different models and methods were tested in this population in order to understand if, by making the models learn only one set of the population at a time, it was possible to achieve better predictive performances. With the purpose to compare this approach with the previously defined, we had to run each set of the population and get the main results. Then, all the predictions were gathered and the several metrics calculated 4.2.

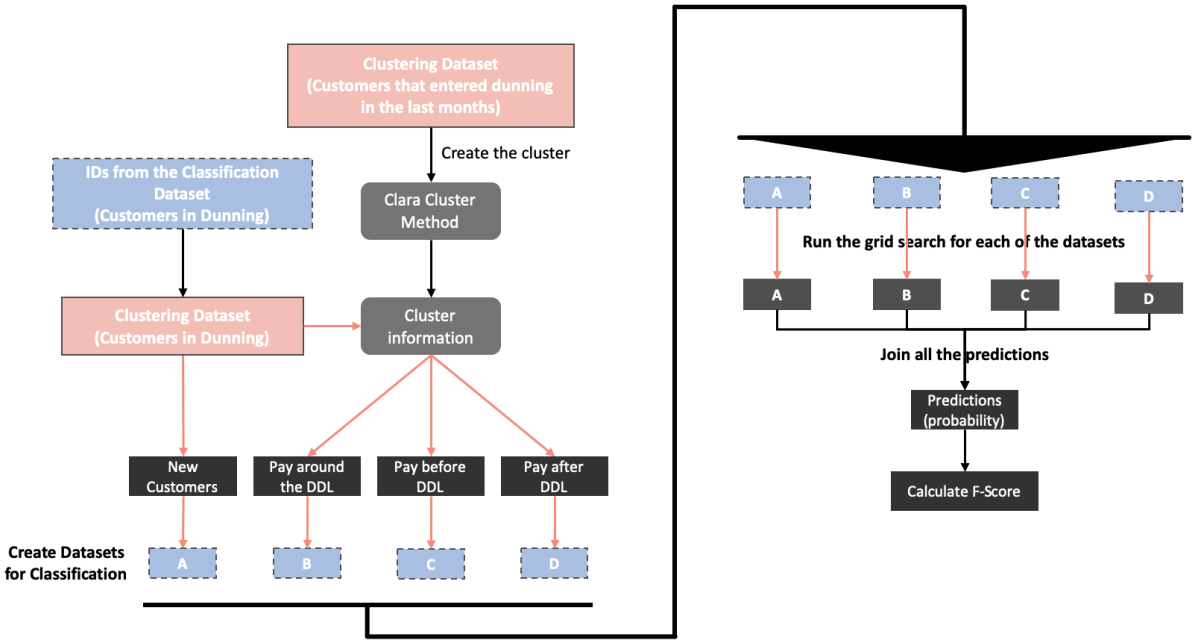


Figure 4.3: The pipeline of the clustering task

4.3 Results without Clustering

Before presenting the results related to the methods and algorithms applied, it is of great importance to highlight the impact of each of the variables on the prediction model. In this way, we verified, using the feature importance from the XGBoost algorithm, that the behavioural group was the most influential, presenting 6 variables in the top 10 of the feature's importance while representing 0.5 of the gain. In the second place, we found 3 features regarding the payment's group, representing 0.12 of the gain.

The following information refers specifically to the results obtained in the scope of the methods and algorithms implemented.

The best F-score results obtained for each of the surpassing pre-processing methods and learning algorithm combination are shown in Figure 4.4.

This graph highlights the positive impact of applied strategies, not only from a global perspective but also in the context of each particular learning algorithm. Such effect is reflected by the improvements evidenced when comparing the baseline (i.e., no re-sampling strategies applied) results with the results obtained after the application of re-sampling strategies. There is also a considerable difference when using the threshold method on the baseline model. In this particular situation, the F-score values resulting from the application of this post-processing technique are at a similar level to those obtained when using pre-processing methods. Nevertheless, the best results derive from the use of the threshold method.

Overall, the various strategies (both of pre and post-processing) appear to be similar in

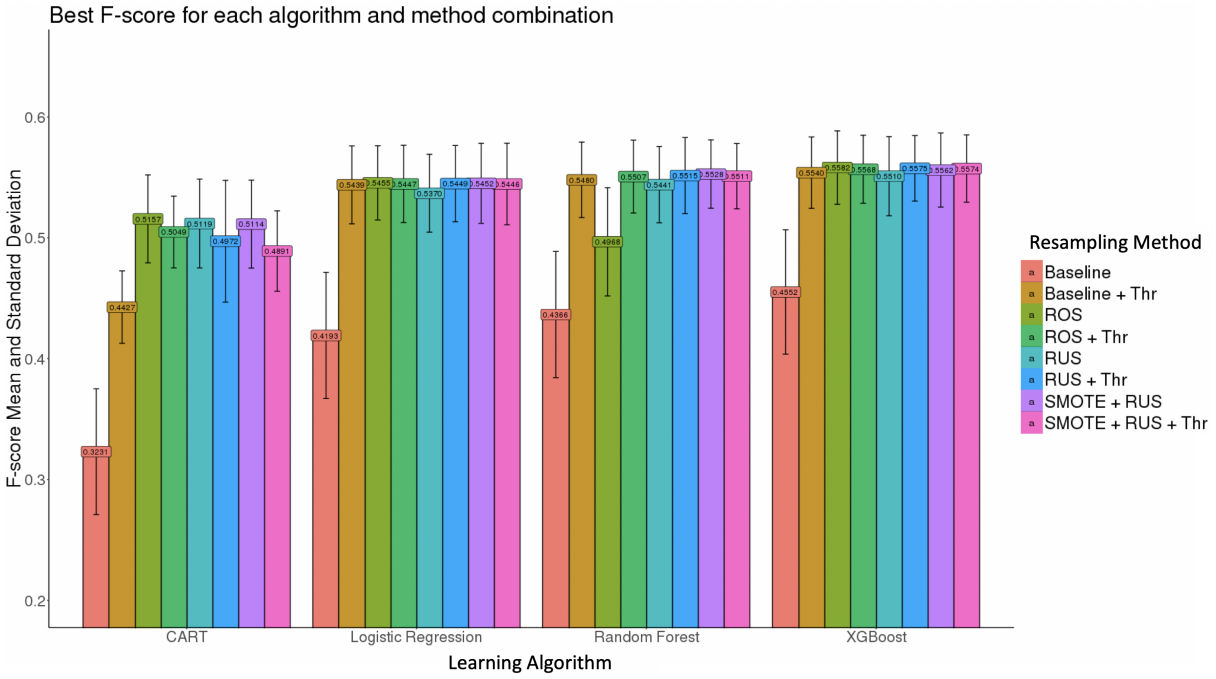


Figure 4.4: Best average F-score and standard deviation obtained by the growing window process for each algorithm and method combination

performance, except for ROS, which had a lower impact on random forests. Regarding the various algorithms, in particular, we found that CART was the worst performer, presenting the highest F-score of 0.5157 when using ROS. This algorithm is followed by: (i) logistic regression, with a result of 0.5455 also when using ROS, (ii) random forest with a result of 0.5528 when using a combination of SMOTE and RUS and (iii) XGBoost, with a score of 0.5582 when using ROS. Thus, although the differences are not large, XGBoost coupled with the ROS method outperformed every other model and technique combinations.

Table 4.2 presents the parameterisation that yields to the results presented in Figure 4.4. The average thresholds applied to the models are summarised in Table 4.3

4.4 Results with Clustering

For the evaluation of the ideal number of clusters, we decided to use both the average silhouette and elbow methods. The result provided by each of these methods was equivalent, being 3 the best k for the clustering step. These results can be observed in Figures 4.5 and 4.6. On Figures A.1 and A.2 it is highlighted the impact of normalisation on the silhouette, increasing its result from an average of 0.4 to 0.62.

As we can see on Table 4.4 the centroids divide the customers into clients that pay before, after and on the due date.

The base hit rate obtained on these populations can be seen on Table 4.5. As expected, the

Table 4.2: Parameters that yield the best F-scores

No Cluster	CART	Logistic Regression	Random Forest	XGBoost
Baseline	minsplit = 10 cp = 0.01	epsilon = 1e-3	mtry = 10 ntree = 250	eta = 0.1 max_depth = 8 nround = 250 cst = 0.85
RUS	minsplit = 10 cp = 0.01 under = 0.25	epsilon = 1e-8 under = 0.25	mtry = 10 ntree = 250 under = 0.5	eta = 0.05 max_depth = 8 nround = 100 cst = 0.55 under = 0.5
ROS	minsplit = 10 cp = 0.01 over = 3	epsilon = 1e-8 over = 3	mtry = 5 ntree = 250 over = 3	eta = 0.05 max_depth = 8 nround = 100 cst = 0.55 over = 3
SMOTE + RUS	minsplit = 10 cp = 0.01 under = 0.5 over = 1.5 k = 5	epsilon = 1e-8 under = 0.75 over = 2 k = 5	mtry = 5 ntree = 250 under = 0.5 over = 3 k = 5	eta = 0.05 max_depth = 8 nround = 250 cst = 0.55 under = 0.5 over = 3 k = 5

Table 4.3: The average thresholds and standard deviation obtained by the growing window process that yield the best F-scores shown in Figure 4.4

	CART	Logistic Regression	Random Forest	XGBoost
Baseline	0.377 ± 0.032	0.260 ± 0.018	0.305 ± 0.015	0.230 ± 0.028
Under	0.605 ± 0.016	0.577 ± 0.015	0.443 ± 0.020	0.412 ± 0.029
Over	0.567 ± 0.029	0.507 ± 0.014	0.348 ± 0.022	0.467 ± 0.020
SMOTE	0.585 ± 0.010	0.492 ± 0.012	0.515 ± 0.016	0.450 ± 0.018

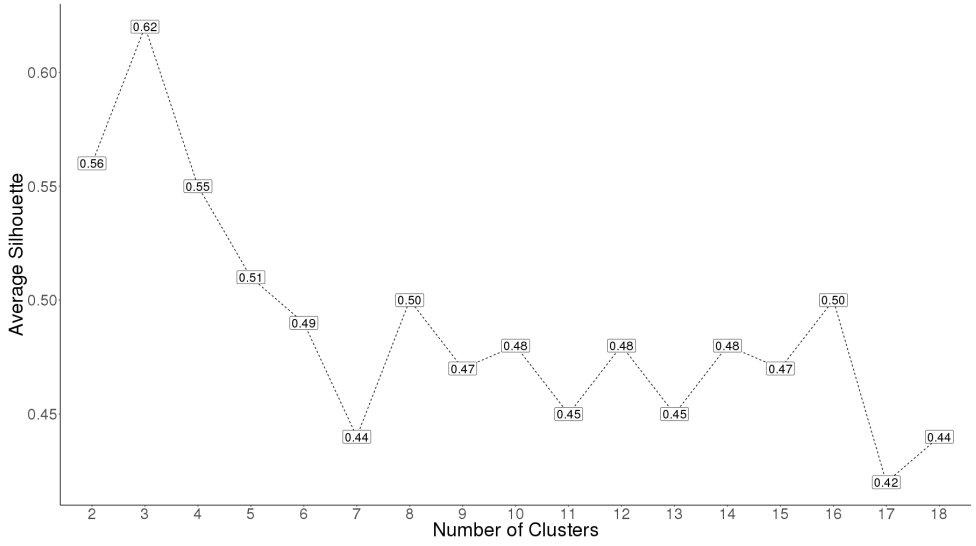


Figure 4.5: Best number of clusters using the Average Silhouette

cluster corresponding to customers with systematic late payment proved to be the one with the largest population. Nevertheless, this cluster presented the lowest hit rate (i.e., hit rate of

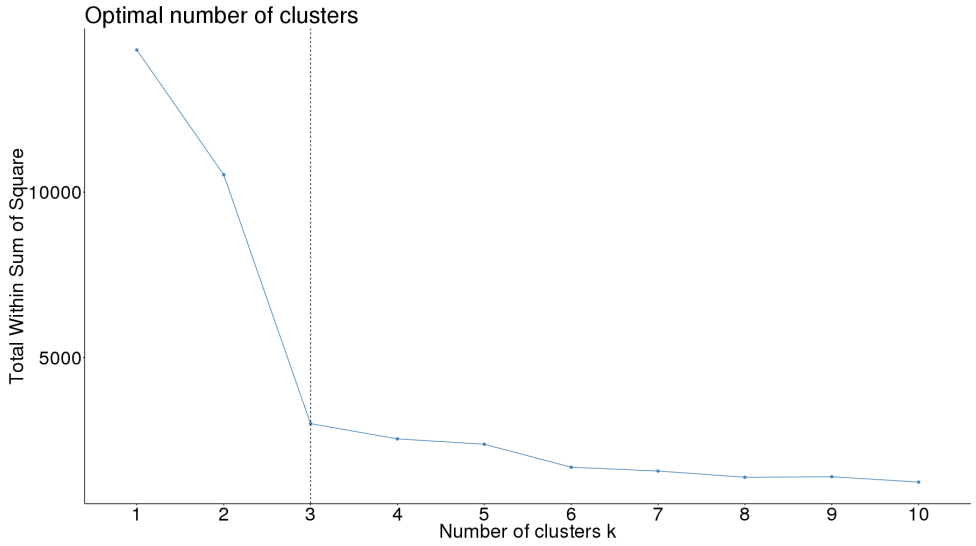


Figure 4.6: Best number of clusters using the Elbow Method

18%), suggesting that such clients, when in dunning, are less likely to progress to payment failure when compared with the rest of the groups. On the other hand, the remaining groups showed a hit rate higher than 30% (i.e., probability of payment failure exceeding 30%). It should be taken into consideration that the percentage of the population does not add up to 100% due to the fact that, when creating the cluster, we did not have all the necessary data for the new customers. Thus, these new customers present the highest hit rate of 40% and represent 27% of the population. Note that these customers were used to create a new group.

In Figure 4.7 we can observe the results concerning the methods applied to each cluster. As we can see, these results are similar to those previously obtained on the unsplit dataset. Additionally, we achieved the same grade of improvement versus the baseline status, and the best algorithm-method combination was the same. Thus, CART had its best F-score of 0.5237 when using ROS, logistic regression a score of 0.5435 when using ROS, random forest a score of 0.5498 when using a combination of SMOTE plus RUS, and finally XGBoost with the best F-score of 0.5536 when using ROS. We can then verify that all models, except for CART, worsened their performance after the distribution of the data across the several clusters.

In addition to the F-score values, we have also obtained the results of AUC and LIFT for each individual cluster. Such results are expressed in the annexed figures and tables (Figures on appendix A). Note that on the LIFT and AUC metrics there was no need for the threshold separation because we have used the probability of belonging to the class to create the measures.

In these models, we were able to find that the clusters with a smaller population are also those that exhibited a worse performance (cf. Figures A.4 and A.5). It is interesting to notice that the logistic regression outperformed every other model particularly on the set of customers that usually pay before the due date. There was also an overall increase in the performance of all metrics when applied in the context of the larger clusters (i.e. clusters of customers with systematic payment delay) The best performance was attributed to the XGBoost algorithm (cf.

Table 4.4: Characterisation of the payment behaviour clusters. In order to easily understand what is the pattern in each of the clusters, each cell has the average of that cluster.

Payment behaviour cluster	Pay around the DDL	Pay before DDL	Pay after DDL
Mean of days elapsed until the regularisation of the monthly payment during the last six months	28,364	23,097	37,613
Standard deviation of days elapsed until the regularisation of the monthly payment during the last six months	1,872	3,435	8,459
Mean of days elapsed until the regularisation of the monthly payment during the last six months when the customer pays before the Due Date	27,419	22,519	26,716
Standard deviation of days elapsed until the regularisation of the monthly payment during the last six months when the customer pays before the Due Date	1,109	2,694	1,183
Mean of days elapsed from the Due Date until the regularisation of the monthly payment during the last six months when the customer pays after the Due Date	2,079	1,755	12,815
Standard deviation of days elapsed from the Due Date until the regularisation of the monthly payment during the last six months when the customer pays after the Due Date	0,968	0,126	6,865
Mean of days between payments	30,376	30,024	30,488
The standard deviation of days between payments	2,609	4,914	11,428
The ratio of months the customer pays the invoice	0,998	0,998	0,963
The ratio of months the customer partially pays the invoice	0	0,001	0,009
The ratio of months the customer pays before the due Date	0,642	0,949	0,264
The ratio of months the customer pays in the 12 days after the due date	0,346	0,039	0,42
The ratio of the number of months the customer had direct debit activated	0,989	0,003	0,018

Table 4.5: Hit Rate and Distribution of the customers by the payment behaviour clusters

Payment behaviour	**Hit Rate**	**Percentage of the population**
Pay arround the DDL	35%	5%
Pay before DDL	33%	4%
Pay after DDL	18%	65%

Figures A.6, A.11 and A.16).

4.5 Discussion

In general, the analysis carried out allowed us to state the following findings:

1. for all learning algorithms, XGBoost was the best performer;

2. the use of strategies to deal with unbalanced datasets was highly beneficial, with an increase

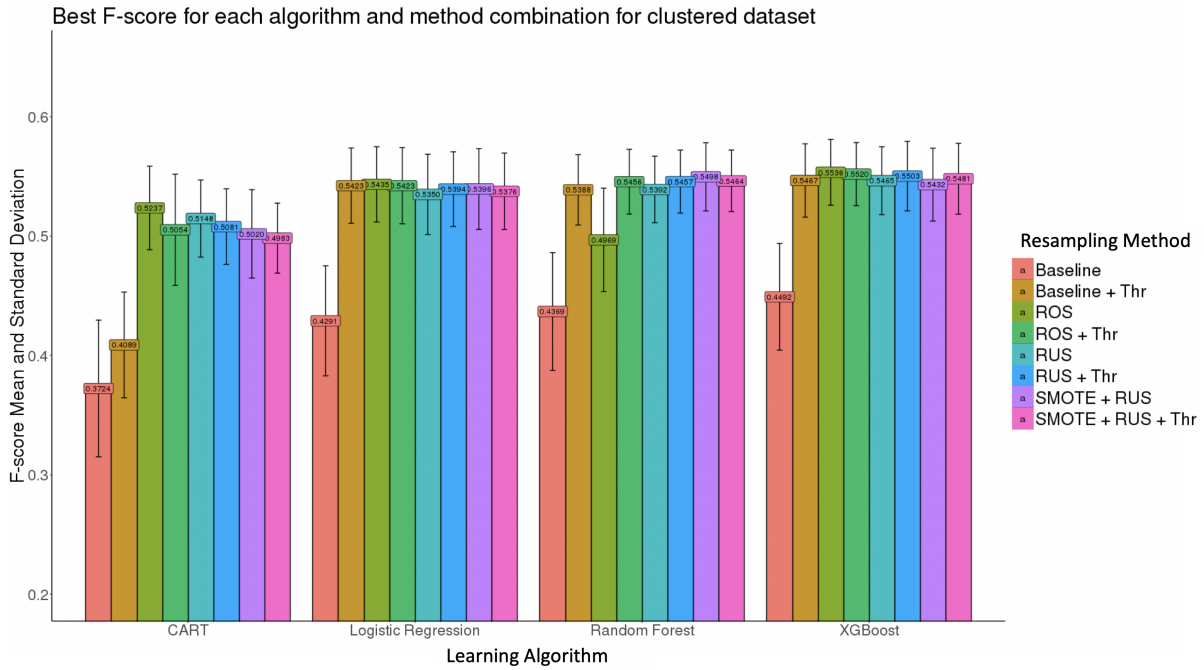


Figure 4.7: Best average F-score and standard deviation obtained by the growing window process for each algorithm and method combination using the clustered datasets

in F-score values;

3. in regarding pre-processing techniques, ROS was the one with the best overall score;

4. simple changes on the threshold were very relevant in terms of performance (i.e., approximate results to those obtained with the above strategies).

Since the threshold is a method with a simple implementation, and usable after the model creation, we consider this one to be a great starting option, and it may even be sufficient in a wide range of situations. Considering these facts, when in the business context we should take into consideration that, even small changes in F-score can be very relevant. Therefore, some cases may benefit from the addition of pre-processing methods. We also verified that the usage of both the threshold and re-sampling strategies can be coupled, introducing a more robust prediction although, for our results, having a slightly smaller F-score.

Still, in the scope of the classification task without clustering we found a worst than expected result on the Random Forest with ROS. A possible explanation for this can be that, as we are randomly replicating the under-represented class (i.e., our positive class), we could be successively replicating the same and worst possible cases, leading to a model with a lower performance.

As far as the clustering task is concerned, we can conclude that its overall performance was no better than the classification task without clusters.

Although we have demonstrated that the XGBoost algorithm is the model that obtained the best performance in the proposed task (i.e., the dichotomous classification of customers

as "payers" or “non-payers"), we admit that it would be of greater interest to carry out the learning to rank approach of the clients according to their payment probability. Such an approach would allow management optimisation of human, temporal and monetary resources available for customer contact (i.e., we can only contact a stipulated amount of customers each month). The telephone contact with the client that presents a greater probability of nonpayment seems to be, intuitively, the best attitude to adopt. However, this task must be performed carefully, because whenever we favour the selection of highly complex clients and, therefore, less likely to return despite significant consumption of resources, we incur the risk of losing customers with easily solved problems.

Therefore, it is imperative to understand the meaning of the obtained results in order to define, not only the number of clients to be contacted, but also their ordering in the respective list (i.e., if we intend to contact, in the first place, customers with the highest probability of payment default or not). To carry out this ranking task, in addition to the importance attributed to the AUC/ROC curve, we have found that the first decile LIFT is also an excellent metric for corporations in general and for this problem in specific.

Considering that customer contact is an expensive task, requiring increased monetary expenses, it is essential to define the appropriate number of clients to address. This can be achieved by noting the change in LIFT versus the associated costs for a specific number of customers. In this regard, XGBoost is, on average and on the large clusters, the best algorithm to adopt (c.f. Figures on appendix sections A.2 and A.3). Here one can also notice that there is no correspondence to the threshold method, due to the fact that, for this metrics, we intend to perform a ranking task by determining the probability of a customer to fail the payment. Hence, we are not going to define it as *YES* or *NO*.

Note that the grid search performed was not exhaustive enough. Although some efforts were carried in order to search for faster implementations of the algorithms with the purpose of increasing the speed of the grid search, this process took several weeks, due to the size of the dataset. Thus, this complexity allows to justify the size of the grid search.

The following and last chapter presents the main conclusions drawn from the elaborated work, as well as the major implications and contributions of this thesis. In addition, suggestions of possible future work are also revealed.

Chapter 5

Conclusion and Future Work

This last chapter presents the main conclusions drawn from the elaborated work, with a critical analysis about the implications of the developed project, its contributions (not only to the global subject of prediction models but also to the particular area of payment default) and limitations, suggesting some possibilities of future work.

5.1 Contributions

Working in dunning in a major company was not an easy task, as the implemented data storage infrastructure that was set available for this thesis was filled with divergences from the ideas conceived by the dunning team itself. Defining the ideal filters for extracting the necessary information from databases was not a simple task. Therefore, it was hard to reach the same values for the population, like the ones that were reported by this team. Similarly, the same reported values for involuntary churn rate were not always achieved – this one was particularly difficult due to the fact that this parameter is usually defined by the company, and often presents several special cases.

It should be noted that the vast majority of the time invested in this thesis was spent on data engineering. Although this has not been defined as the main task to be performed, the possibility of covering the whole process has proved to be very positive and beneficial. Although it was a vital activity, given the confidentiality of the data obtained, it could not be further explored in this thesis.

This thesis studies the application of data pre- and post- processing techniques when formalising the problem of default prediction as an imbalanced domain learning task. These are compared to Logistic Regression, one of the best default prediction models in this scope, and three well-known machine learning algorithms. For the best performance of the imbalanced binary classification task, we have found that the best approach consists of an XGBoost model combined with the RUS data pre-processing method. On the other hand, the ranking activity is superior when using the XGBoost models with the random over-sampling method.

From our perspective, and after proper validation of the obtained results, we consider that the present work adds a interesting contribution to all the companies providing subscription-based services. Such conviction derives from the ability to improve on the results of the previous approach by the dunning team. We have also checked what was the best category of variables, reported their performance and evaluated the performance of subsetting the customers into their payment behaviour group.

In general, the contributions of this thesis are not only related to the idealised experimental setup. In addition to the experiments performed, we have also tested different approaches to the involuntary churn problem, as presented below.

1. Instead of only classifying the customers as involuntary churners or not, we have also facilitated the contact task of the dunning team through the definition of a list concerning the most important customers to reach.
2. To accomplish that, we have set a target that was superior to the simple knowledge of which customers were evolving to involuntary churn, allowing the dunning team to early reach customers with simpler problems, avoiding increased complexity.
3. Creating different payment profile clusters was designed not only as a tool for the dunning team but also to evaluate the effectiveness of combining clustering and classification methods. Although our experiments did not show signs of improvement on the classification when combined with clustering, this application should not be completely disregarded as the exploration of clustering algorithms was not exhaustive.

5.2 Future Work

In the future, some possible improvements may be considered, such as the following.

1. Creating a model to predict if a customer will pay before the due date is one missing approach. We consider that this would help the dunning team taking the preemptive action of sending a message to remind only the most prone to default customers.
2. Testing an approach targeted to each billing cycle, insofar as this seems superior to us in the business model perspective. To perform this, instead of using the growing window algorithm, growing window and sliding window techniques could be applied.
3. As the dunning team has a limited number of daily calls to perform, ranking can be a very helpful method to help scheduling the priority costumers to contact. This task should take into consideration not only the probability of involuntary churn but also the probability of responding to the attempt of contact at every given time period.
4. The identification of not only the clusters that define customer's payment behaviour but also the customer behaviour when in dunning. This particular step will help the collections

team in the process of returning the customers to a regular state, helping them decide the best-individualised approach for each client. Additionally, we could output the probability of belonging to each of the clusters.

5. Testing several other parameters in the grid search, as well as new clustering methods, is one of the missing aspects of this thesis, which is why we consider that further work is needed on this evaluation.

6. Since we have different types of customers, we could evaluate the stacking of different models, where different models learn different parts of the problem.

7. Testing new algorithms of deep learning should be considered, such as reinforcement learning and event sequence mining (i.e., an algorithm where we could learn recommendations for the best set of actions, as well as the proper time for each action to be applied in different customers).

Appendix A

Results

A.1 Clustering results

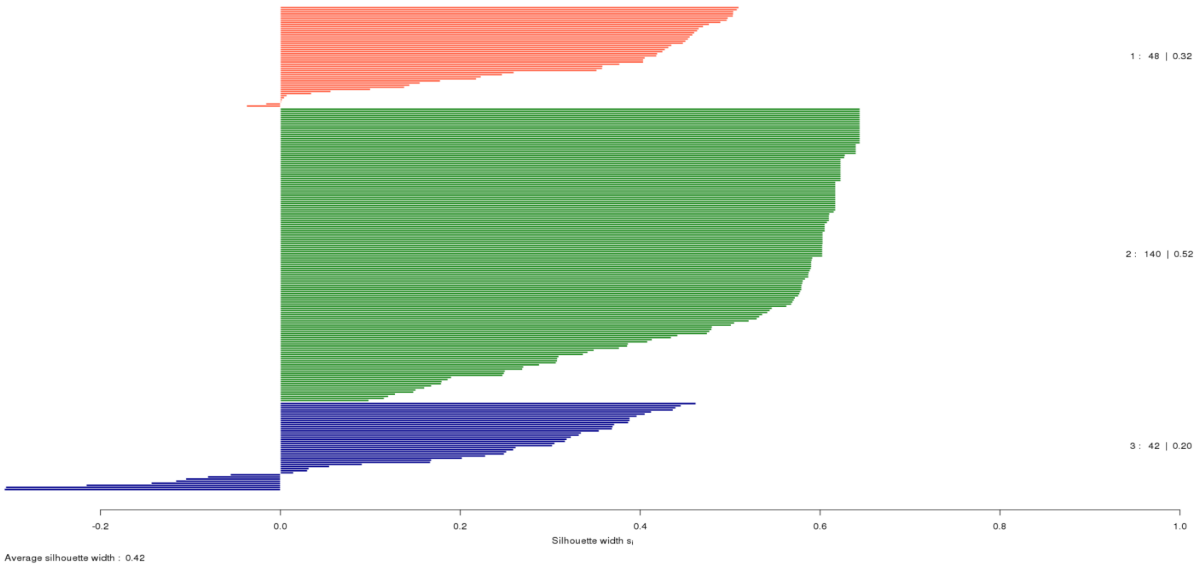


Figure A.1: Silhouette without a normalised dataset

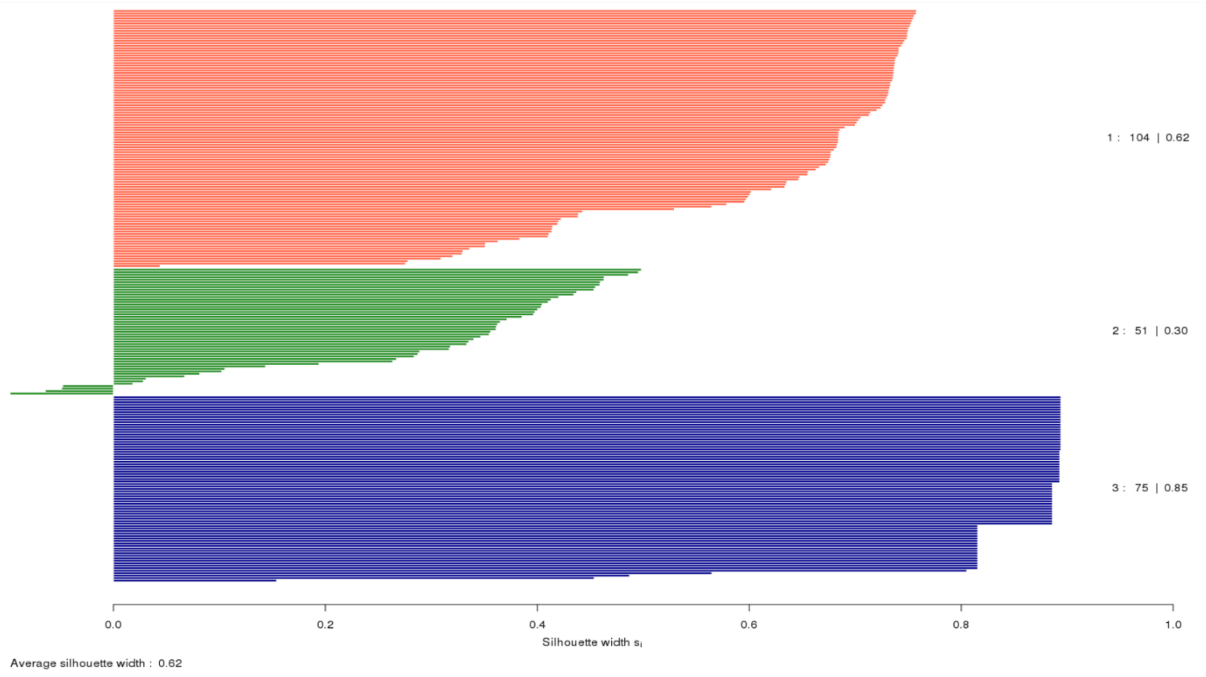


Figure A.2: Silhouette with the normalised dataset

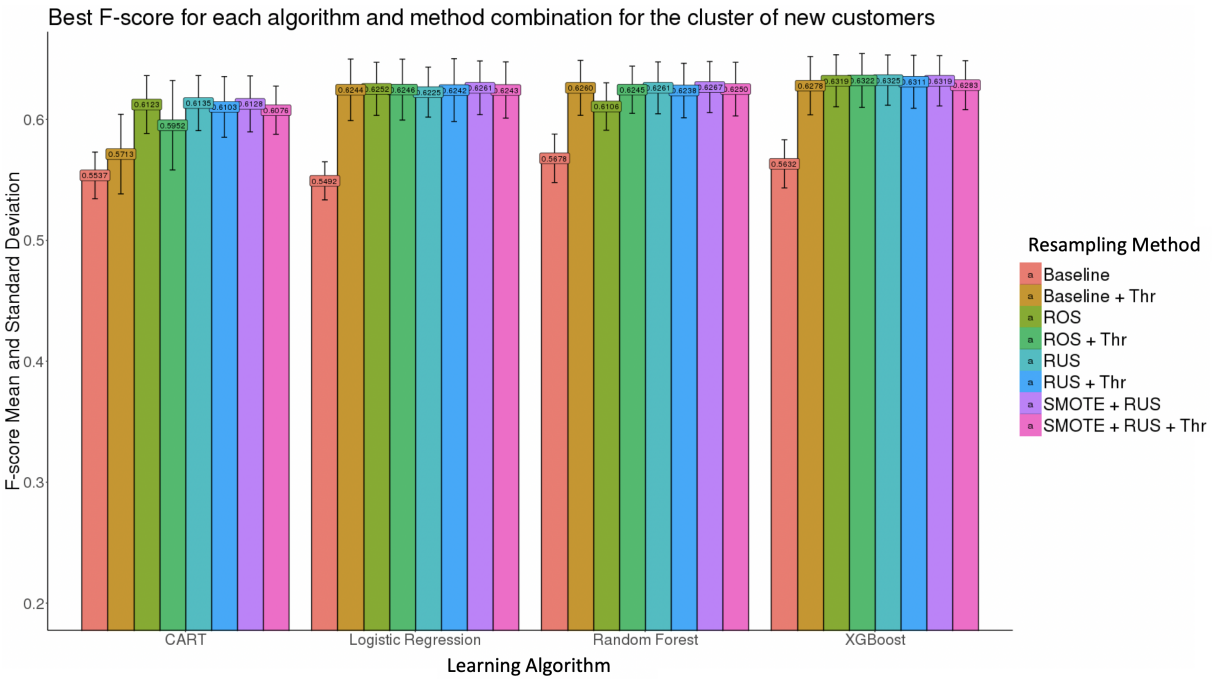


Figure A.3: Best average F-score and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of new customers

Table A.1: Parameters that yield the best F-scores for the cluster of new customers

Cluster of the new customers	CART	Logistic Regression	Random Forest	XGBoost
Baseline	minsplit = 10 cp = 0.05	epsilon = 1e-8	mtry = 10 ntree = 250	eta = 0.05 max_depth = 8 nround = 100 cst = 0.55
RUS	minsplit = 10 cp = 0.01 under = 0.5	epsilon = 1e-8 under = 0.5	mtry = 5 ntree = 100 under = 0.5	eta = 0.05 max_depth = 3 nround = 100 cst = 0.85 under = 0.5
ROS	minsplit = 10 cp = 0.01 over = 2	epsilon = 1e-8 over = 2	mtry = 5 ntree = 250 over = 3	eta = 0.05 max_depth = 3 nround = 250 cst = 0.85 over = 2
SMOTE + RUS	minsplit = 10 cp = 0.01 under = 0.75 over = 1.5 k = 3	epsilon = 1e-8 under = 0.9 over = 2 k = 5	mtry = 10 ntree = 250 under = 0.5 over = 1.5 k = 3	eta = 0.1 max_depth = 3 nround = 250 cst = 0.55 under = 0.5 over = 1.5 k = 3

Table A.2: The average thresholds and standard deviation obtained by the growing window process that yield the best F-scores for the cluster of new customers shown in Figure A.3

	CART	Logistic Regression	Random Forest	XGBoost
Baseline	0.287 ± 0.005	0.308 ± 0.021	0.335 ± 0.018	0.307 ± 0.045
Under	0.503 ± 0.082	0.455 ± 0.023	0.478 ± 0.017	0.473 ± 0.021
Over	0.548 ± 0.067	0.463 ± 0.020	0.392 ± 0.034	0.462 ± 0.024
SMOTE	0.560 ± 0.029	0.498 ± 0.026	0.492 ± 0.016	0.538 ± 0.020

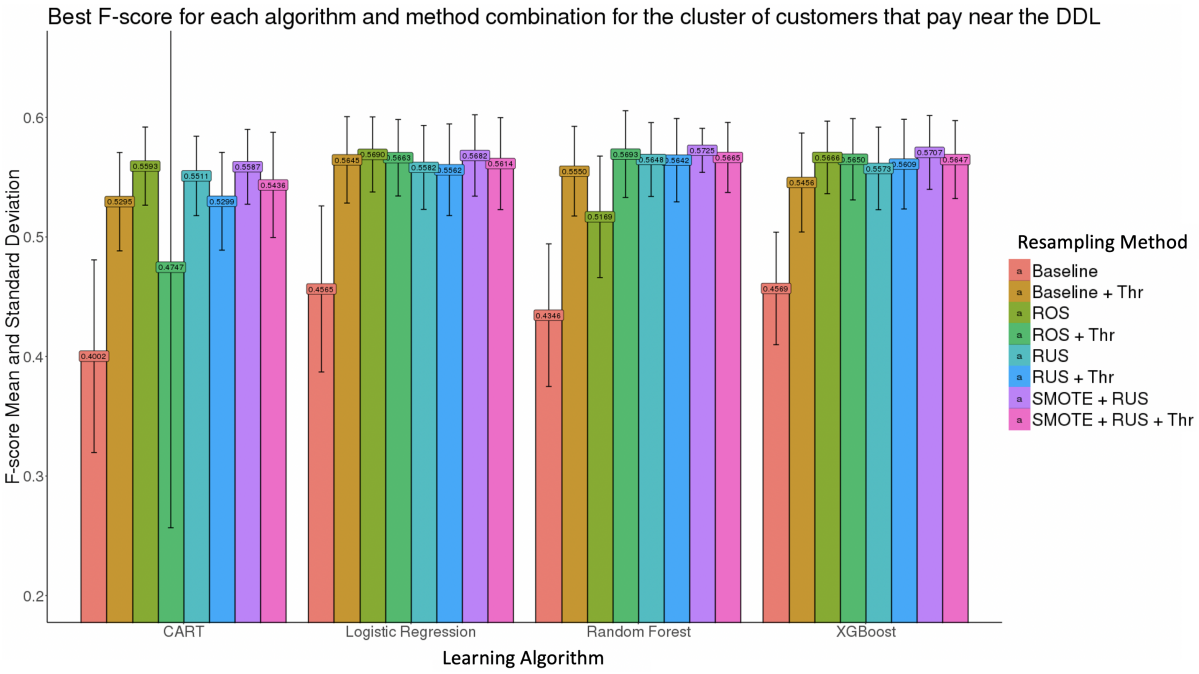


Figure A.4: Best average F-score and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

Table A.3: Parameters that yield the best F-scores for the cluster of customers that pay near the DDL

Cluster that pay by the DD	**CART**	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	minsplit = 10 cp = 0.01	epsilon = 1e-3	mtry = 10 ntree = 250	eta = 0.1 max_depth = 8 nround = 250 cst = 0.55
RUS	minsplit = 20 cp = 0.01 under = 0.25	epsilon = 1e-3 under = 0.5	mtry = 5 ntree = 100 under = 0.5	eta = 0.05 max_depth = 3 nround = 100 cst = 0.85 under = 0.5
ROS	minsplit = 10 cp = 0.01 over = 3	epsilon = 1e-3 over = 3	mtry = 5 ntree = 250 over = 3	eta = 0.05 max_depth = 3 nround = 100 cst = 0.55 over = 3
SMOTE + RUS	minsplit = 10 cp = 0.01 under = 0.5 over = 1.5 k = 3	epsilon = 1e-3 under = 0.75 over = 2 k = 5	mtry = 5 ntree = 100 under = 0.75 over = 3 k = 3	eta = 0.05 max_depth = 3 nround = 100 cst = 0.55 under = 0.5 over = 1.5 k = 5

Table A.4: The average thresholds and standard deviation obtained by the growing window process that yield the best F-scores for the cluster of customers that pay near the DDL shown in Figure A.4

	CART	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	0.308 ± 0.068	0.253 ± 0.036	0.320 ± 0.028	0.083 ± 0.048
Under	0.650 ± 0.248	0.397 ± 0.049	0.425 ± 0.025	0.393 ± 0.064
Over	0.658 ± 0.065	0.503 ± 0.041	0.353 ± 0.043	0.480 ± 0.072
SMOTE	0.465 ± 0.232	0.468 ± 0.059	0.437 ± 0.015	0.472 ± 0.074

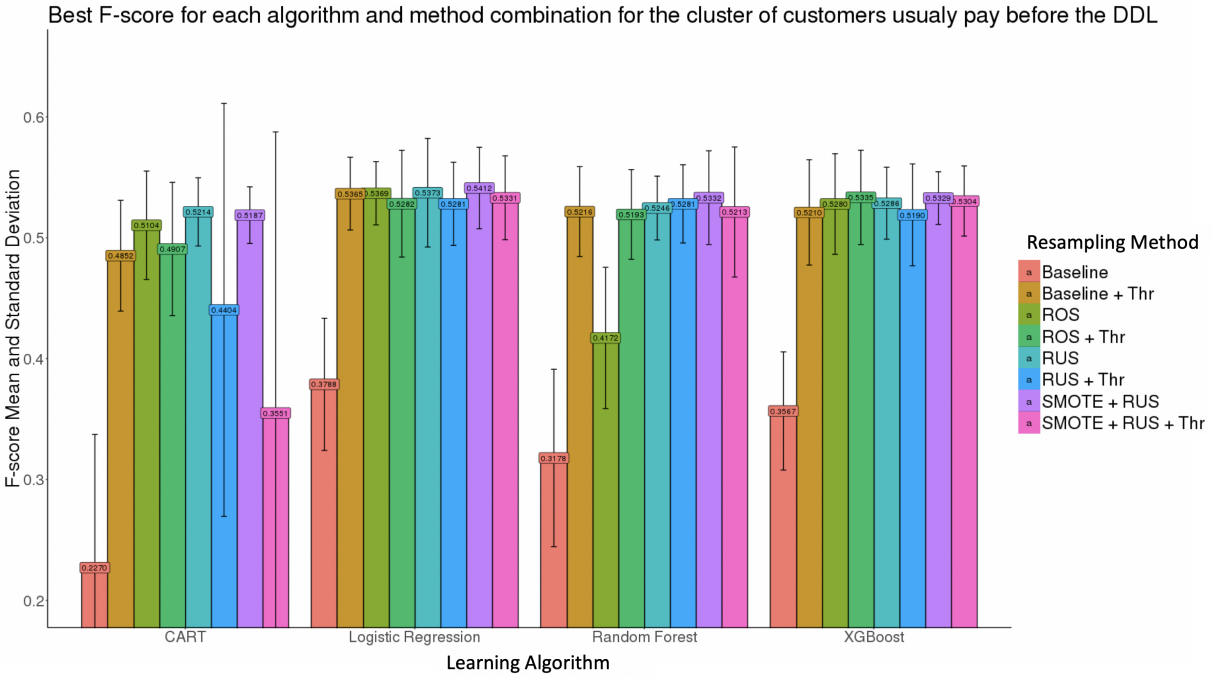


Figure A.5: Best average F-score and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers that pay near the DDL

Table A.5: Parameters that yield the best F-scores for the cluster of customers that usually pay before the DDL

Cluster of customers that usually pay before the DD	**CART**	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	minsplit = 20 cp = 0.01	epsilon = 1e-1	mtry = 10 ntree = 100	eta = 0.1 max_depth = 3 nround = 250 cst = 0.85
RUS	minsplit = 10 cp = 0.01 under = 0.25	epsilon = 1e-1 under = 0.5	mtry = 10 ntree = 250 under = 0.25	eta = 0.1 max_depth = 3 nround = 100 cst = 0.85 under = 0.25
ROS	minsplit = 10 cp = 0.01 over = 3	epsilon = 1e-8 over = 3	mtry = 5 ntree = 250 over = 3	eta = 0.05 max_depth = 3 nround = 100 cst = 0.55 over = 3
SMOTE + RUS	minsplit = 20 cp = 0.01 under = 0.25 over = 2 k = 5	epsilon = 1e-1 under = 0.75 over = 2 k = 5	mtry = 5 ntree = 100 under = 0.5 over = 3 k = 5	eta = 0.05 max_depth = 3 nround = 100 cst = 0.55 under = 0.5 over = 3 k = 5

Table A.6: The average thresholds and standard deviation obtained by the growing window process that yield the best F-scores for the cluster of customers that usually pay before the DDL shown in Figure A.5

	CART	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	0.187 ± 0.070	0.260 ± 0.031	0.308 ± 0.017	0.192 ± 0.047
Under	0.715 ± 0.081	0.340 ± 0.098	0.530 ± 0.028	0.593 ± 0.056
Over	0.520 ± 0.173	0.538 ± 0.080	0.362 ± 0.022	0.477 ± 0.045
SMOTE	0.733 ± 0.250	0.478 ± 0.064	0.502 ± 0.026	0.558 ± 0.045

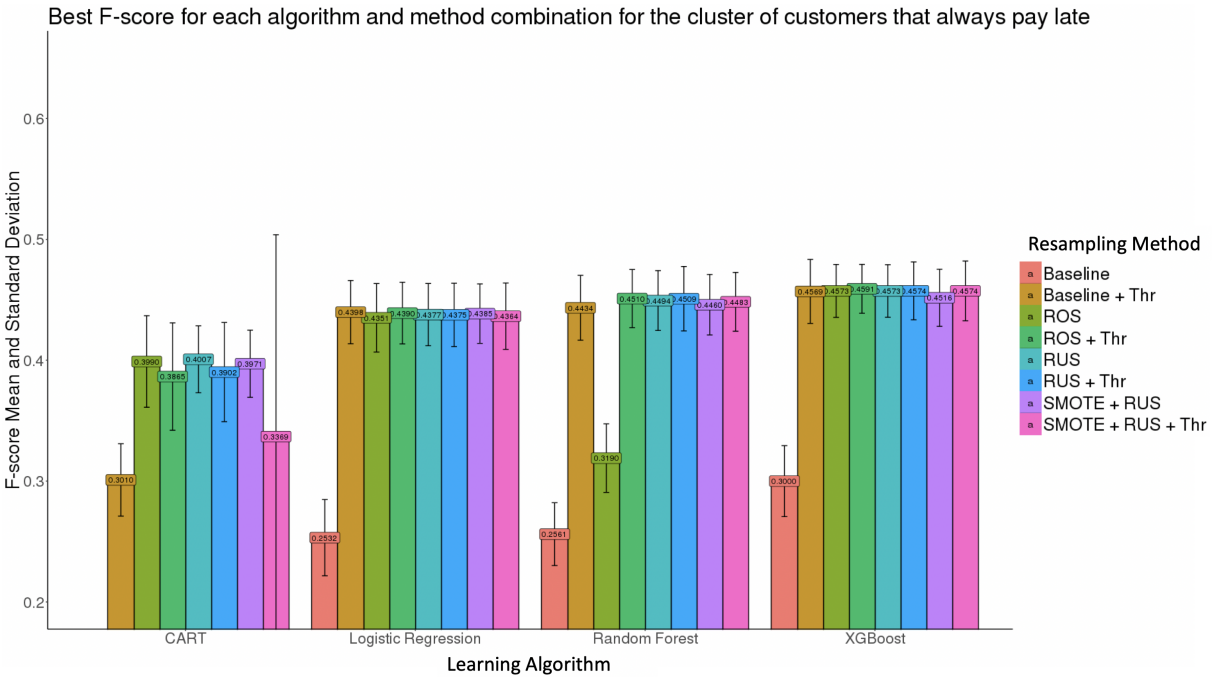


Figure A.6: Best average F-score and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers that always pay late

Table A.7: Parameters that yield the best F-scores for the cluster of customers that always pay late

Cluster of the always late customers	**CART**	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	minsplit = 10 cp = 0.01	epsilon = 1e-8	mtry = 10 ntree = 100	eta = 0.1 max_depth = 8 nround = 250 cst = 0.55
RUS	minsplit = 10 cp = 0.01 under = 0.25	epsilon = 1e-8 under = 0.25	mtry = 5 ntree = 250 under = 0.25	eta = 0.05 max_depth = 3 nround = 250 cst = 0.55 under = 0.25
ROS	minsplit = 10 cp = 0.01 over = 3	epsilon = 1e-8 over = 3	mtry = 5 ntree = 100 over = 3	eta = 0.1 max_depth = 3 nround = 250 cst = 0.85 over = 3
SMOTE + RUS	minsplit = 10 cp = 0.01 under = 0.25 over = 1.5 k = 3	epsilon = 1e-3 under = 0.75 over = 3 k = 5	mtry = 5 ntree = 250 under = 0.5 over = 3 k = 5	eta = 0.05 max_depth = 8 nround = 250 cst = 0.55 under = 0.25 over = 2 k = 5

Table A.8: The average thresholds and standard deviation obtained by the growing window process that yield the best F-scores for the cluster of customers that always pay late shown in Figure A.6

	CART	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	0.168 ± 0.004	0.215 ± 0.015	0.260 ± 0.011	0.198 ± 0.012
Under	0.605 ± 0.050	0.513 ± 0.026	0.542 ± 0.015	0.530 ± 0.015
Over	0.543 ± 0.053	0.453 ± 0.023	0.318 ± 0.018	0.442 ± 0.017
SMOTE	0.705 ± 0.014	0.500 ± 0.040	0.472 ± 0.025	0.577 ± 0.012

A.2 AUC

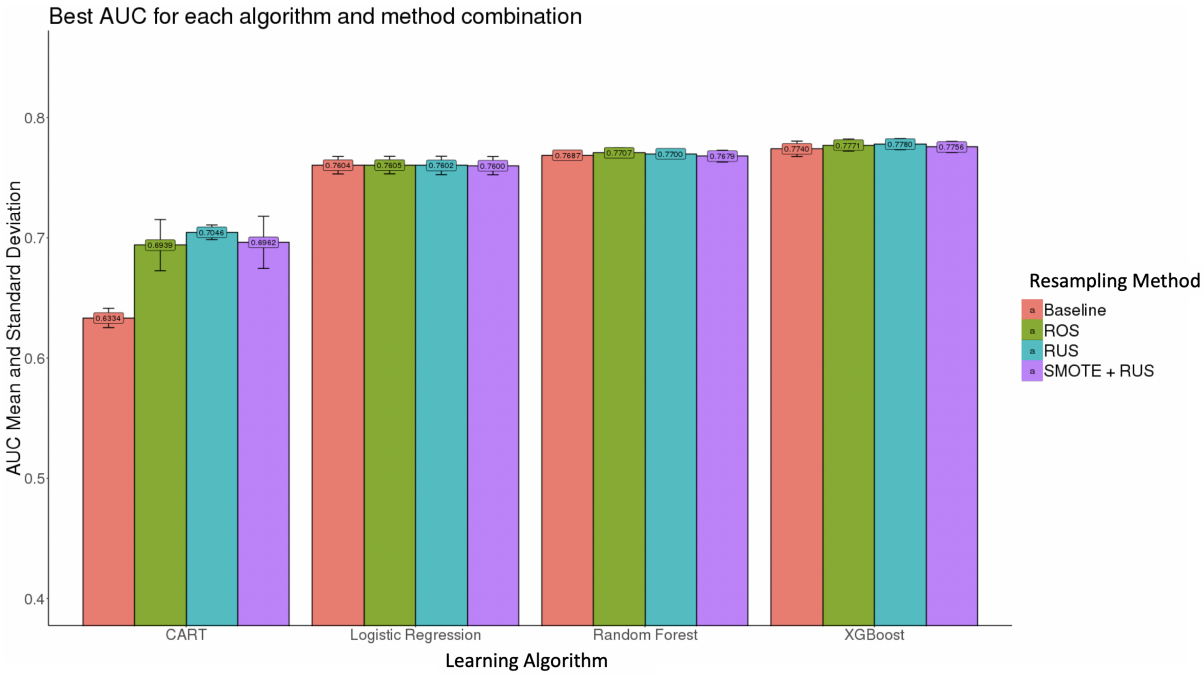


Figure A.7: Best average AUC and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

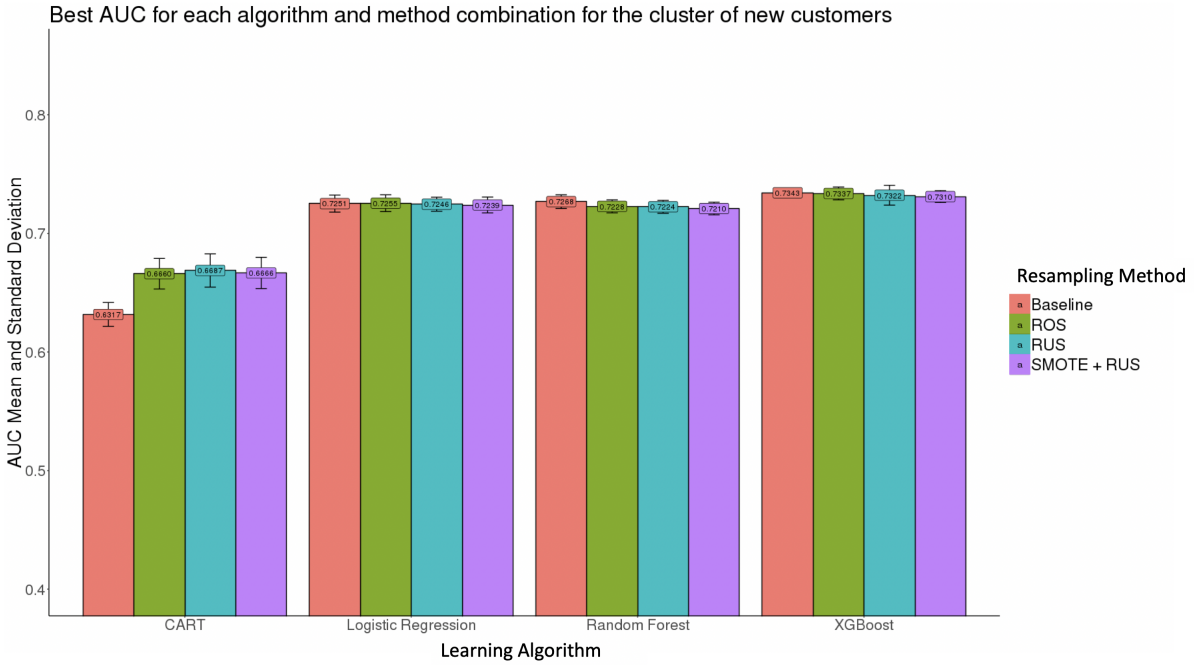


Figure A.8: Best average AUC and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

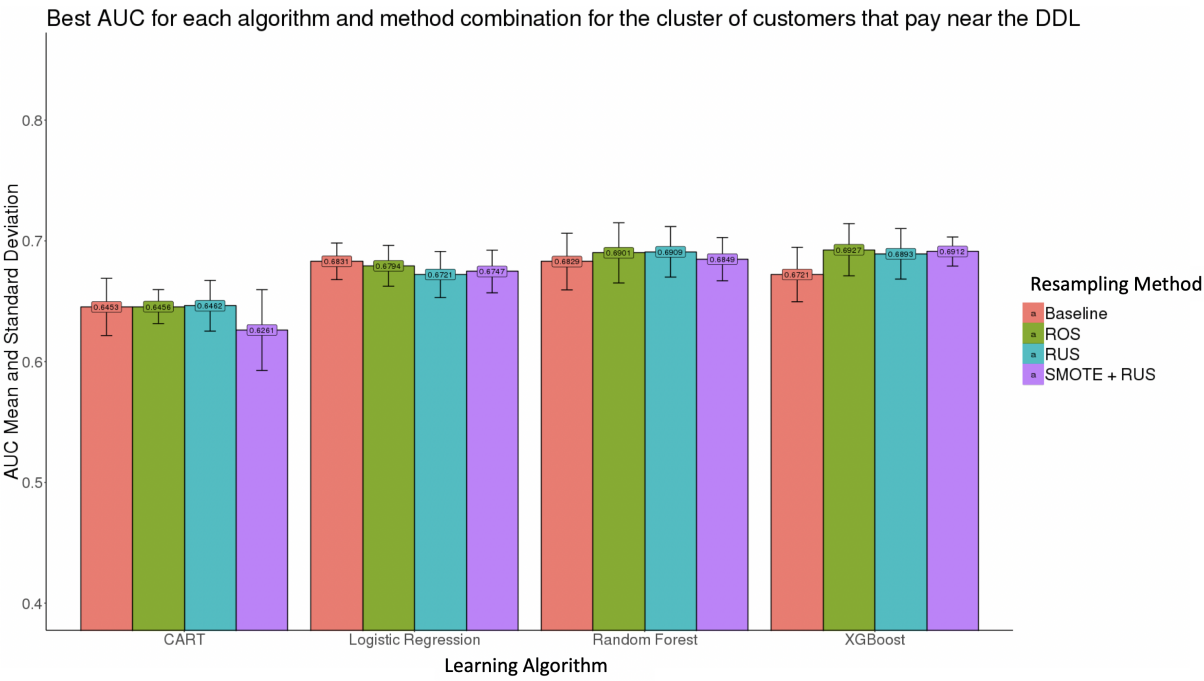


Figure A.9: Best average AUC and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

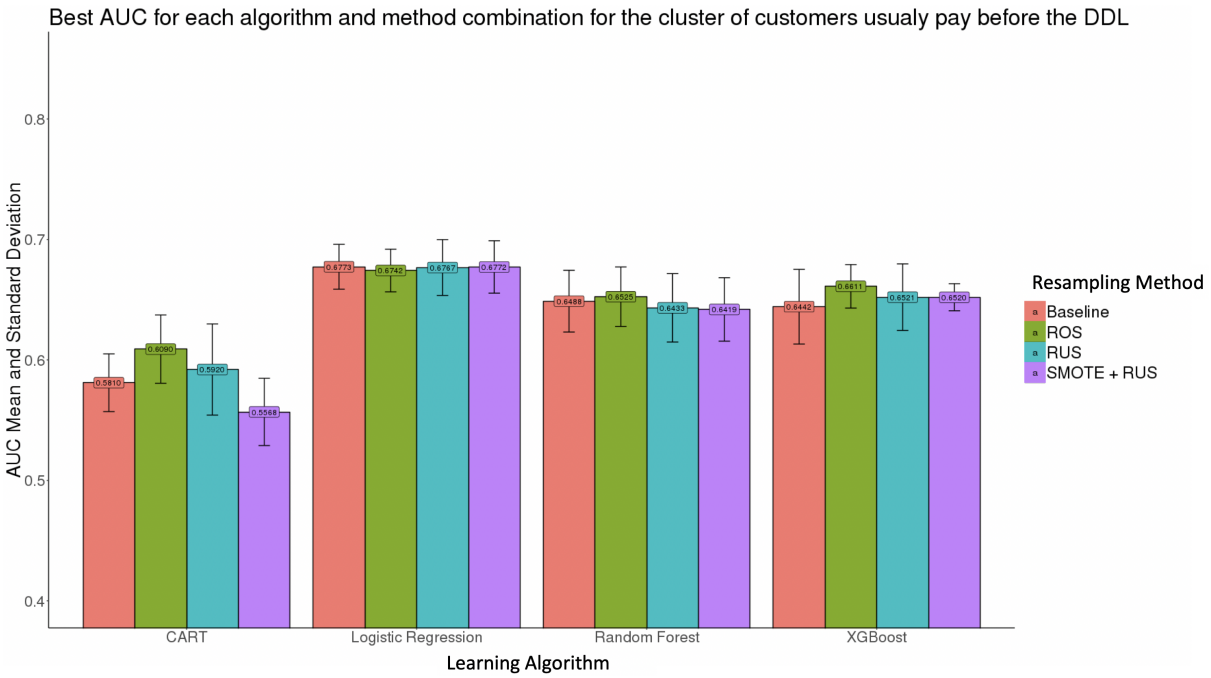


Figure A.10: Best average AUC and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

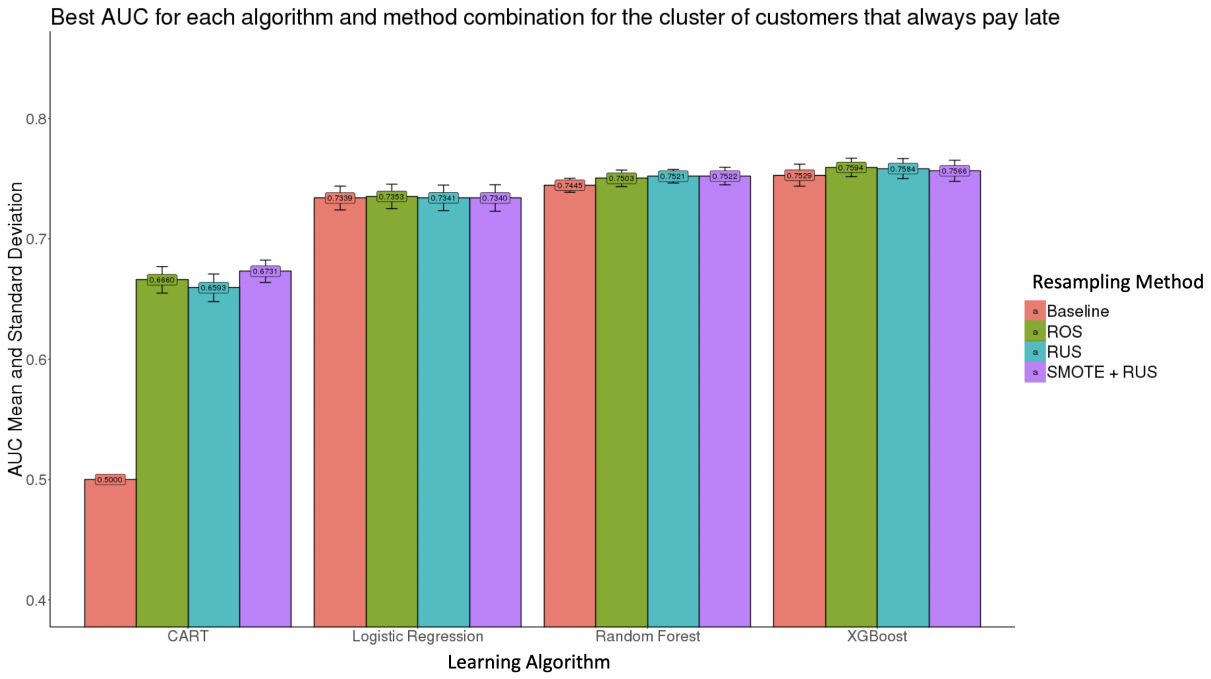


Figure A.11: Best average AUC and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

A.3 LIFT

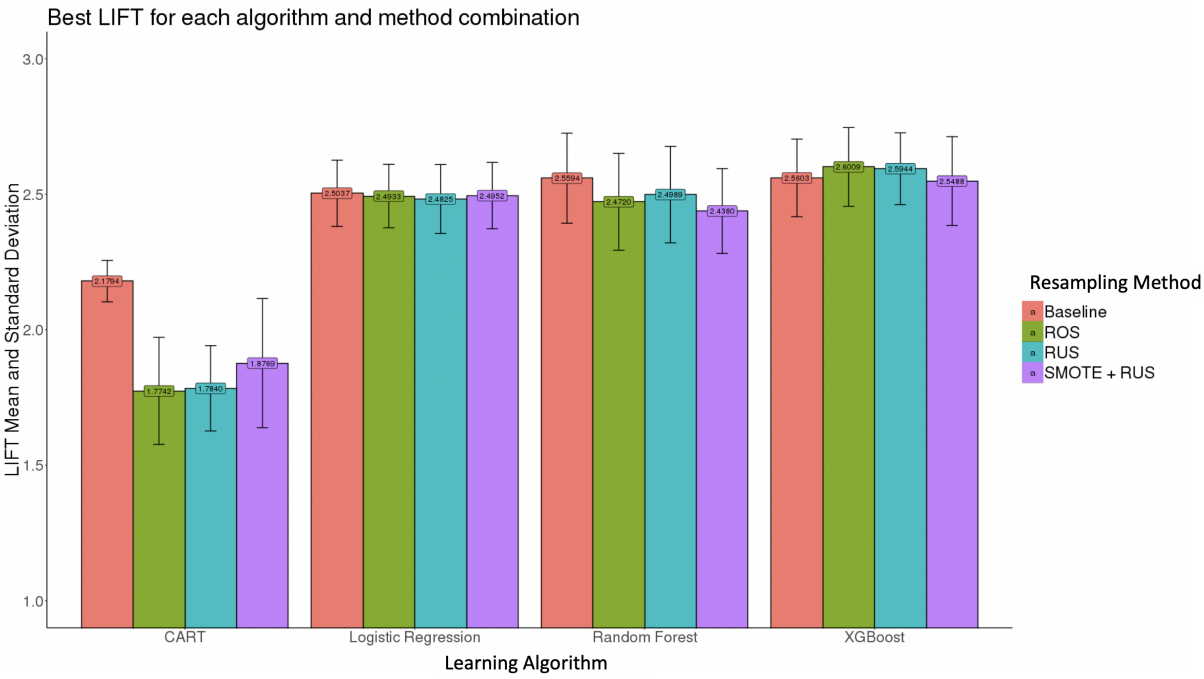


Figure A.12: Best average LIFT and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

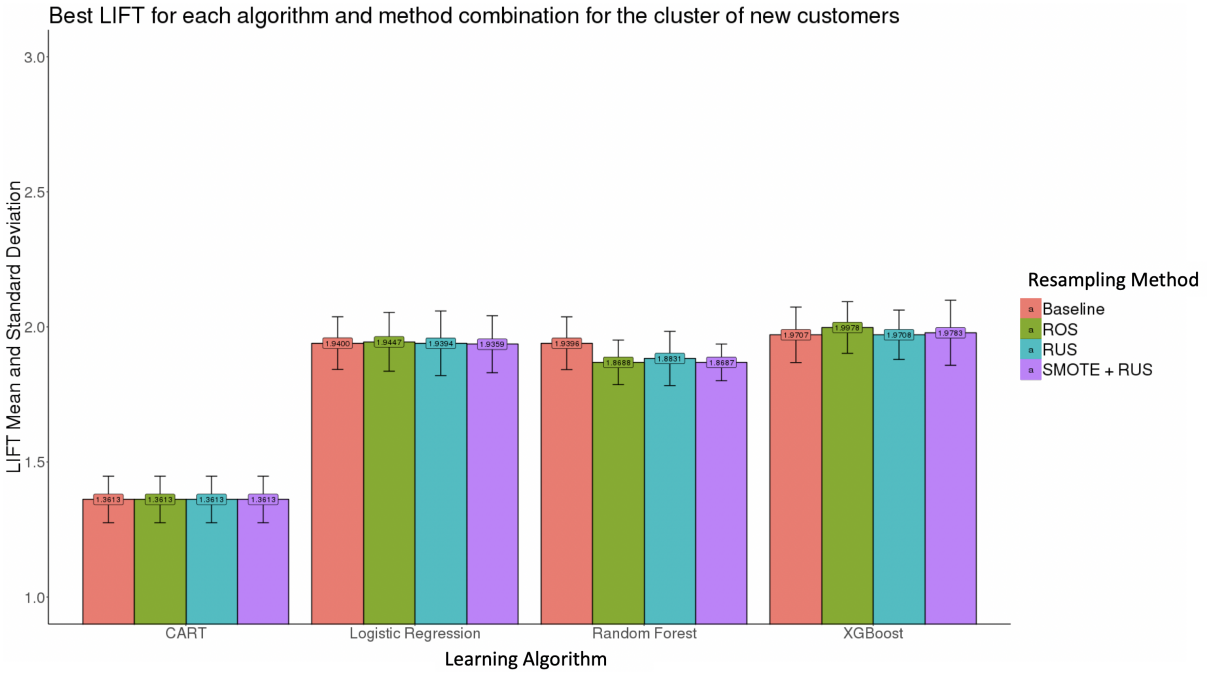


Figure A.13: Best average LIFT and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

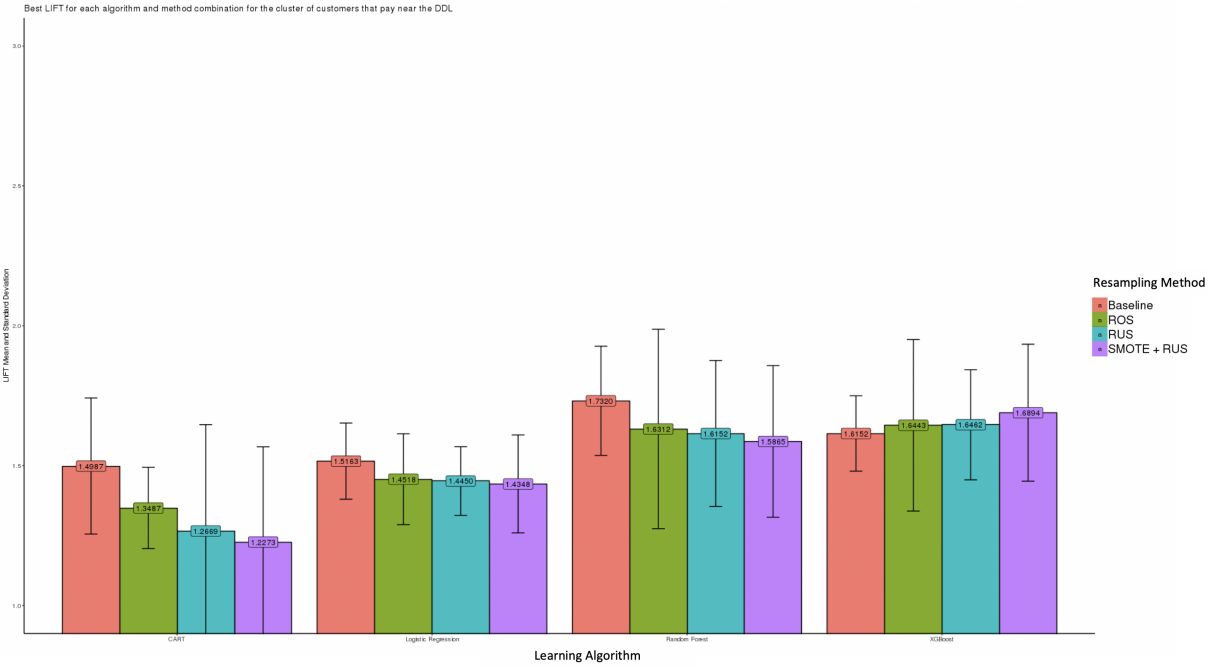


Figure A.14: Best average LIFT and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

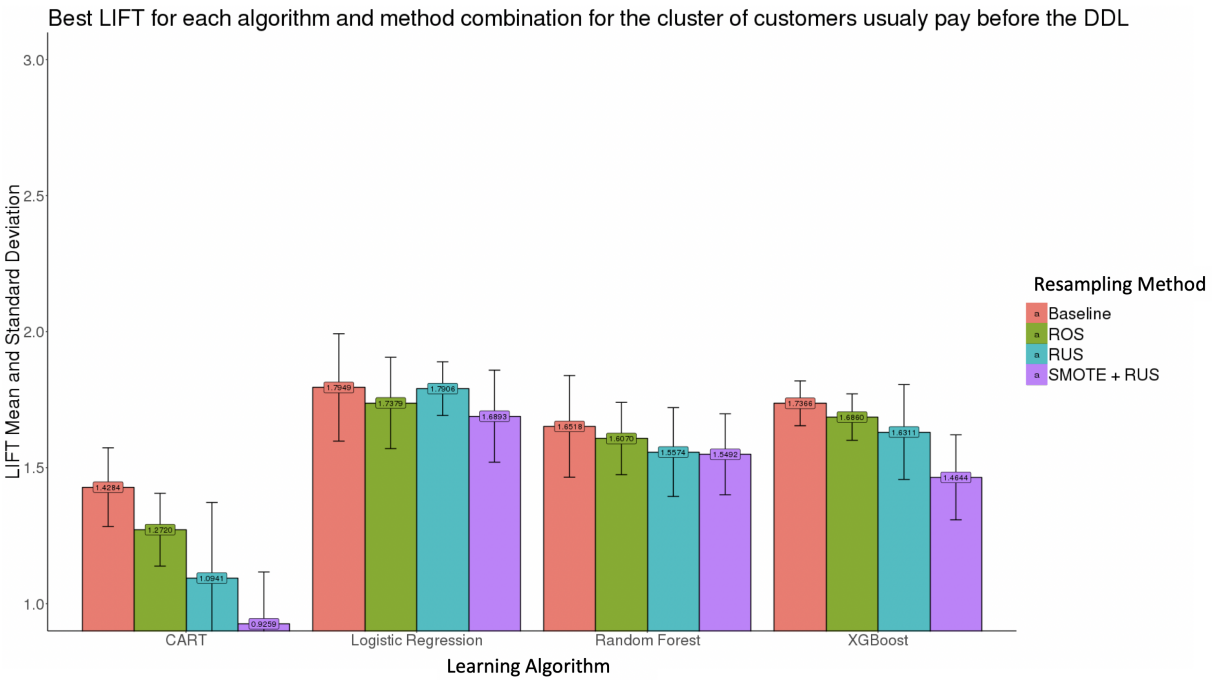


Figure A.15: Best average LIFT and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

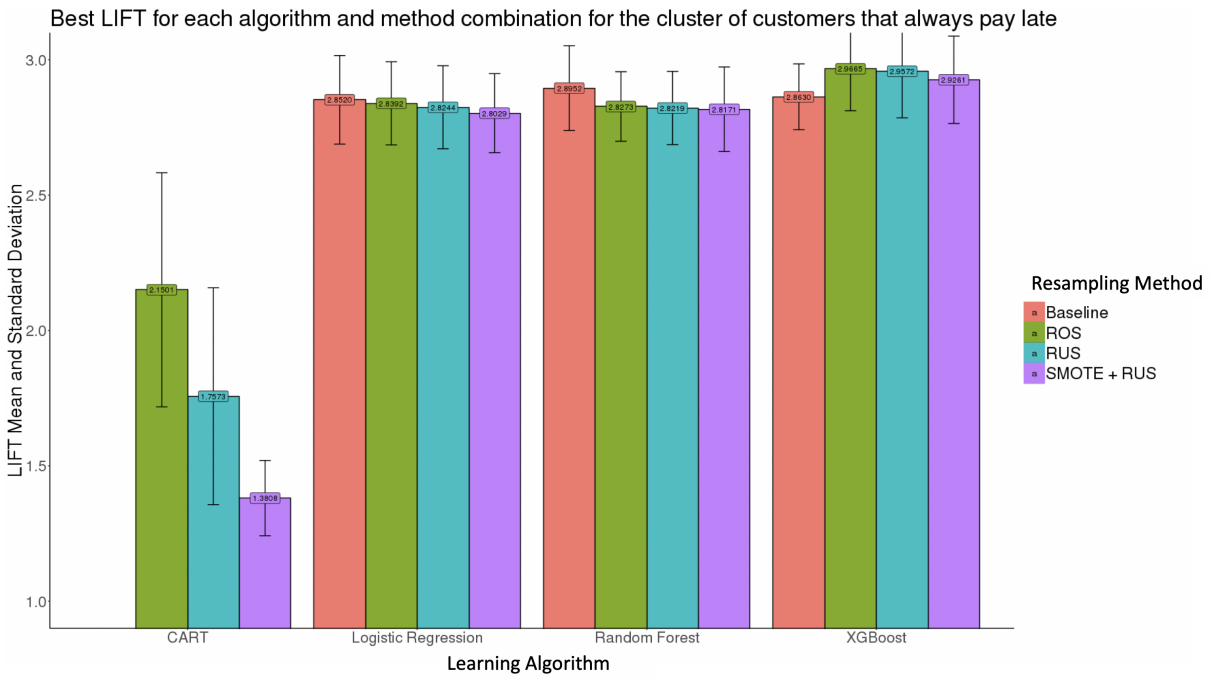


Figure A.16: Best average LIFT and standard deviation obtained by the growing window process for each algorithm and method combination for the cluster of customers usually pay before the DDL

Bibliography

[1] Alan Agresti. *An introduction to categorical data analysis*. Wiley, New York, 1996. ISBN: 0471113387 9780471113386.

[2] Edward I. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, 23(4):589–609, 1968.

[3] Edward I Altman and Gabriele Sabato. Modeling credit risk for smes: evidence from the us market. In *Managing and Measuring Risk: Emerging Global Standards and Regulations After the Financial Crisis*, pages 251–279. World Scientific, 2013.

[4] William H Beaver. Financial ratios as predictors of failure. *Journal of accounting research*, pages 71–111, 1966.

[5] Michael JA Berry and Gordon S Linoff. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[6] Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49, 2016.

[7] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[8] Leo Breiman. Random Forests. *Machine Learning*, 45(1), 2001.

[9] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[10] Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. Classification and regression trees chapman & hall. *New York*, 1984.

[11] Peter Bühlmann, Bin Yu, et al. Analyzing bagging. *The Annals of Statistics*, 30(4):927–961, 2002.

[12] Peter Bühlmann. Bagging, boosting and ensemble methods. *Handbook of Computational Statistics*, 01 2012. doi:10.1007/978-3-642-21551-3_33.

[13] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, 2002.

[14] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[15] Kristof Coussement and Dirk Van den Poel. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert systems with applications*, 34(1):313–327, 2008.

[16] Jonathan N. Crook, David B. Edelman, and Lyn C. Thomas. Recent developments in consumer credit risk assessment. *Europ. J. of Operational Research*, 183(3):1447–1465, 2007.

[17] Annette J Dobson. *An introduction to generalized linear models*. Chapman and Hall/CRC, 1990.

[18] Veronikha Effendy, ZK Abdurahman Baizal, et al. Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest. In *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, pages 325–330. IEEE, 2014.

[19] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88, 1996.

[20] Michele Fratello and Roberto Tagliaferri. Decision trees and random forests. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, page 374, 2018.

[21] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[22] João Gama, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. *Extração de conhecimento de dados: data mining*. Edições Sílabo, 2015.

[23] Michael J Gombola, Mark E Haskins, J Edward Ketz, and David D Williams. Cash flow in bankruptcy prediction. *Financial Management*, pages 55–65, 1987.

[24] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[25] Haibo He and E.A. Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21:1263 – 1284, 2009.

[26] Venkata Jagannath. Random forest template for tibco spotfire® - wiki page, Mar 2017.

[27] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[28] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[29] Jochen Kruppa, Alexandra Schwarz, Gerhard Arminger, and Andreas Ziegler. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 40:5125–5131, 2013.

[30] GR Kumar, VS Kongara, and DG Ramachandra. An efficient ensemble based classification techniques for medical diagnosis. *Int J Latest Technol Eng Manag Appl Sci*, 2:5–9, 2013.

[31] Junxiang Lu. Predicting customer churn in the telecommunications industry - an application of survival analysis modeling using sas. *SAS User Group International (SUGI27) Online Proceedings*, pages 114–27, 2002.

[32] Robert N Lussier. A nonfinancial business success versus failure prediction mo. *Journal of Small Business Management*, 33(1):8, 1995.

[33] Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2018. R package version 2.0.7-1 — For new features, see the 'Changelog' file (in the package source).

[34] Abinash Mishra and U Srinivasulu Reddy. A novel approach for churn prediction using deep learning. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–4. IEEE, 2017.

[35] John A Nelder and Robert William Maclagan Wedderburn. Generalized Linear Models. *Journal of the Royal Statistical Society*, 135:370–384, 1972.

[36] Ja Ohlson. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, 18(1):109–131, 1980.

[37] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2018.

[38] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN: 0408709294.

[39] scikit learn. Logistic regression, Jul 2019.

[40] Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2019. R package version 4.1-15.

[41] L. Torgo. An infra-structure for performance estimation and experimental comparison of predictive models in r. *CoRR*, abs/1412.0436, 2014.

[42] M Antónia Amaral Turkman and Giovani Loiola Silva. *Modelos Lineares Generalizados - da teoria à prática*. Universidade de Lisboa e Universidade Técnica de Lisboa, Lisboa, 2000.

[43] Gary M. Weiss. Mining with rarity: A unifying framework. *SIGKDD Explor. Newsl.*, 6(1): 7–19, 2004.

[44] Marvin N. Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017.

[45] Huiting Zheng, Jiabin Yuan, and Long Chen. Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation. *Energies*, 10 (8):1168, 2017.