Un-factorize Non-food NPS on a food-based retailer

Inês Carvalho

UMinho | 2022

Universidade do Minho
Escola de Ciências

Inês Gonçalves Carvalho

Un-factorize Non-food NPS on a food-based retailer

outubro de 2022

Universidade do Minho
Escola de Ciências

Inês Gonçalves Carvalho

Un-factorize Non-food NPS on a food-based retailer

Dissertação de Mestrado
Mestrado em
Estatística para
Ciência de Dados

Trabalho efetuado sob a orientação da
Prof. Doutora Susana Margarida Ferreira de Sá Faria e de
Dr. Ana da Costa Freitas

outubro de 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Licença concedida aos utilizadores deste trabalho

Acknowledgements

Firstly, I would like to thank MC Sonae for the opportunity to develop my master thesis, it was truly one of the most amazing experiences.

A special thank you to Liliana Bernardino and Ana Freitas, the best mentors I could ever asked for, for welcoming me with open arms and hearts, and teaching me so much. It was incredibly empowering working with you. I also owe my gratitude to my brilliant buddy Francisco Barbosa who guided me from day one, dealt with all my existential crises and taught me the true meaning of the word buddy. I will never forget. To Ana Carvalho for always having the best word of advice and a smile on her face. To Filipe Miranda for all the amazing inputs, motivation and laughing at my jokes. To Nuno Chicória from whom I learned so many great facts and not always about data science. I would also like to thank Vitor Sousa and Alexandre Sousa for always being open to help me. Thank you all for your friendship. I will cherish you always.

Another special thank you goes to my dissertation supervisor, Prof. Susana Faria. I am grateful for her guidance, total availability, support, insights and critical look throughout the process of this dissertation. Thank you for showing me that in everything we do we must apply a dose of passion, but also a lot effort and hard work.

Last, but definitely not the least, I would like to acknowledge the strong support of my family and friends. To my parents and my siblings for always believing in me. To Rui for his unconditional support, patience and love above anything else. To my good-fairies, Jacinta and Eliana, for cheering me up and guiding me all the way through. To Joana, for being my rock. And to all of my friends for hearing me complain and supporting me through everything. You know who you are.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledge the Code of Ethical Conduct of the University of Minho.

Universidade do Minho, 31 de outubro de 2022,

Inês Gonçalves Carvalho

(Inês Gonçalves Carvalho)

Resumo

Un-factorize Non-food NPS on a food-based retailer

O *Net Promoter Score* (NPS) é uma métrica muito utilizada para medir o nível de lealdade dos consumidores. Neste sentido, esta dissertação pretende desenvolver um modelo de classificação que permita identificar a classe do NPS dos consumidores, ou seja, classificar o consumidor como Detrator, Passivo ou Promotor, assim como perceber os fatores que têm maior impacto nessa classificação. A informação recolhida permitirá à organização ter uma melhor percepção das áreas a melhorar de forma a elevar a satisfação do consumidor.

Para tal, propõe-se uma abordagem de *Data Mining* para o problema de classificação multiclasse. A abordagem utiliza dados de um inquérito e dados transacionais do cartão de fidelização de um retalhista, que formam o conjunto de dados a partir dos quais se consegue obter informações sobre as pontuações do Net Promoter Score (NPS), o comportamento dos consumidores e informações das lojas. Inicialmente é feita uma análise exploratória dos dados extraídos. Uma vez que as classes são desbalanceadas, várias técnicas de reamostragem são aplicadas para equilibrar as mesmas. São aplicados dois algoritmos de classificação: Árvores de Decisão e *Random Forests*. Os resultados obtidos revelam um mau desempenho dos modelos. Uma análise de erro é feita ao último modelo, onde se conclui que este tem dificuldade em distinguir os Detratores e os Passivos, mas tem um bom desempenho a prever os Promotores.

Numa ótica de negócio, esta metodologia pode ser utilizada para fazer uma distinção entre os Promotores e o resto dos consumidores, uma vez que os Promotores são a segmentação de clientes mais prováveis de beneficiar o mesmo a longo prazo, ajudando a promover a organização e atraíndo novos consumidores.

Palavras-chaves: Net Promoter Score, *Data Mining*, Classificação, Árvores de Decisão, *Random Forest*

Abstract

Un-factorize Non-food NPS on a food-based retailer

More and more companies realise that understanding their customers can be a way to improve customer satisfaction and, consequently, customer loyalty, which in turn can result in an increase in sales. The NPS has been widely adopted by managers as a measure of customer loyalty and predictor of sales growth.

In this regard, this dissertation aims to create a classification model focused not only in identifying the customer's NPS class, namely, classify the customer as Detractor, Passive or Promoter, but also in understanding which factors have the most impact on the customer's classification. The goal in doing so is to collect relevant business insights as a way to identify areas that can help to improve customer satisfaction.

We propose a Data Mining approach to the NPS multi-class classification problem. Our approach leverages survey data, as well as transactional data collected through a retailer's loyalty card, building a data set from which we can extract information, such as NPS ratings, customer behaviour and store details. Initially, an exploratory analysis is done on the data. Several resampling techniques are applied to the data set to handle class imbalance. Two different machine learning algorithms are applied: Decision Trees and Random Forests. The results did not show a good model's performance. An error analysis was then performed in the later model, where it was concluded that the classifier has difficulty distinguishing the classes Detractors and Passives, but has a good performance when predicting the class Promoters.

In a business sense, this methodology can be leveraged to distinguish the Promoters from the rest of the consumers, since the Promoters are more likely to provide good value in long term and can benefit the company by spreading the word for attracting new customers.

Key-words: Net Promoter Score, Data Mining, Classification, Decision Trees, Random Forest

”All models are wrong, but some are useful.”

George Box

Contents

List of Tables

List of Figures

List of Acronyms

AUC	Area Under the ROC Curve
CART	Classification and Regression Trees
CX	Customer Experience
CM	Confusion Matrix
CP	Complexity Parameter
CRISP-DM	Cross Industry Standard Process for Data Mining
CV	Cross-Validation
DM	Data Mining
DOP	Direção Operacional
DT	Decison Trees
FN	False Negative
FP	False Positive
HEOM	Heterogeneous Euclidean-Overlap Metric
HVDM	Heterogeneous Value Difference Metric
ID3	Iterative Dichotomiser 3
IR	Imbalance Ratio
L12M	Last 12 Months

L6M	Last 6 Months
L3M	Last 3 Months
LM	Last Month
ML	Machine Learning
MVP	Minimum Viable Product
NPS	Net Promoter Score
OOB	Out-of-Bag
PDP	Partial Dependence Plot
P	Precision
RF	Random Forest
ROS	Random Oversampling
RUS	Random Undersampling
ROC	Receiver Operating Characteristics
S	Sensitivity
SMOTE	Synthetic Minority Oversampling Technique
TN	True Negative
TP	True Positive
WOM	Word of Mouth

Chapter 1

Introduction

The following chapter will contextualize the dissertation topic within the retail industry and its objectives. Ultimately, an overview of the dissertation structure will be presented.

1.1 Big Data Analytics in Retail

The business environment is being challenged by the connected community of customers, mostly due to internet revolution and social networks, where the opinion about a certain product, service or the global image of a company is being spread all over [38]. Word of Mouth (WOM) communication and user reviews about previous experiences have become an important source of information to consumers [17]. As such, depending on the information received, companies can either acquire a new customer or lose one. In that sense, customer loyalty is a key differentiation element across industries as companies recognize its role as a part of their commercial success. A way to monitor it is through market research based on metrics like the Net Promoter Score (NPS) [47].

The Net Promoter Score (NPS) was introduced by [56] as an indicator that measures the level of customer loyalty to a company. This metric is measured primarily through surveys in which the consumers are asked to provide feedback such as the willingness for recommending the company's product and/or services to others. The description uses a ten-point response scale segmenting customers as Promoters, those who answer nine or ten; Passives, those who answer seven or eight; and Detractors, those who answer between zero and six.

Overall, employing analytical models to predict the NPS allows the company to deepen their knowledge across various levels (e.g. stores, type of consumer, etc) without having to assess the entire population. Moreover, it also allows the company to identify which bottlenecks are potentially associated with a lower level of loyalty, which can provide guidelines on how to increase customer satisfaction and, as a consequence, customer loyalty.

1.2 Project Setting

The current project was developed at the Advanced Analytics and Insights team at Cartão Continente department on MC Sonae.

MC Sonae

Sonae is a multinational company managing a diversified portfolio of businesses in different sectors, such as retail, wellness and well being (MC Sonae), financial services (Universo and Bright-Pixel), technology (Worten), real estate (Sierra), telecommunications (NOS), fashion (Zeitreel), sports (ISRG) and plant based ingredients (Sparkfood Sonae).

MC Sonae is the leading food retailer in Portugal, with a number of distinct business segments such as Continente (hypermarkets), Continente Modelo (supermarkets) and Continente Bom Dia (convenience supermarkets), Meu Super (franchising proximity store), Bagga (cafeteria), Go Natural (health-oriented supermarkets and restaurants), Make Notes and Note! (bookshop and stationery store), ZU (products and services for pets), Well's (health, well-being and optics), Dr.Wells (dentistry and cosmetic medicine) and Zippy, Mo (children and adult textile) and Cozinha Continente (restaurant).

Loyalty Program

Most companies are unable to manage their relationships with their customers as they do not have detailed information on which they can act to make the relationship more valuable [45].

Cartão Continente was launched in 2007 with the aim to have a better understanding on their customer's behaviour and thus increase the customer loyalty to the brand on acting on a more personal level. It currently has around 4 million active loyalty accounts and more than 80% of the total transactions are done using loyalty cards.

The loyalty card is the main source of knowledge about customer behavior and has allowed the company to improve their management and as a consequence their relationship with the customer.

Nowadays, the card includes 19 permanent partners (Continente Stores, Meu Super, Note, Wells's, Zu, Bagga, Mo and Zippy) as well as some external partners like Galp and the Ibersol group (with brands such as Burger King, SOL, KFC, Pizza Hut, etc).

1.3 Project Motivation and Objectives

More than ever, forward-looking companies are moving toward the goal of understanding their customers as individuals and using that understanding to relate with the company rather than with competitors, translating into increasing sales. For MC Sonae, building a business around the customer relationship is a primary business issue.

In this sense, this project arises with the objective of predicting all Note's customers NPS class, namely, predict the customer as a Promoter, Passive or Detractor. Moreover, it aims to determine what are the most important variables and thus understand which effects can impact the customer's classification. The identification of customers and their loyalty behavior will allow to identify store experiences, services and products that lead to pleasant or unpleasant experiences to consumers, and that way identify areas to improve customer satisfaction.

The project requires the use of data from surveys and Cartão Continente loyalty card. This last data includes information on transactions from different businesses.

Different programming languages were used to extract, merge and analyse data from the surveys and Continente database, such as SQL, R and Python, as well as Excel for data analysis.

1.4 Proposed Approach

In order to achieve the objectives proposed earlier, this project will follow the steps of the Cross Industry Standard Process for Data Mining (CRISP-DM) [63]. Proposed in 2000, it is still one of the most common methodologies for data mining, analytics, and data science projects. This framework involves six phases, namely: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment (Figure 1.1)

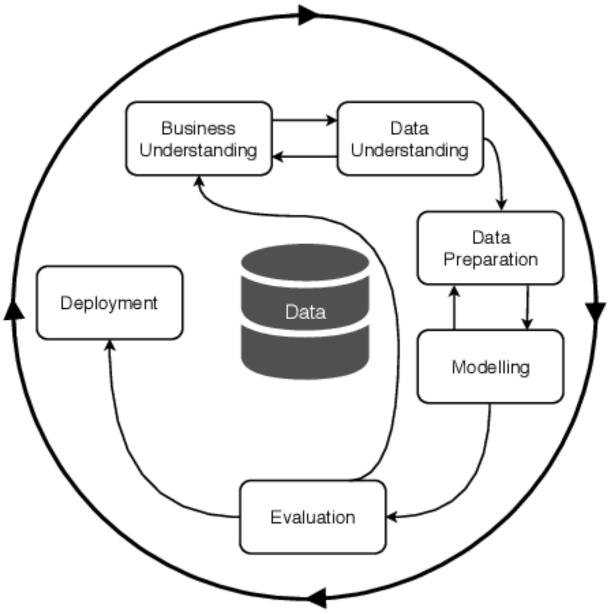


Figure 1.1: CRISP-DM reference model phases. Source: [48]

- Business Understanding aims to understand the business needs and sets clear goals for the project.

- Data Understanding consists on collecting, describing and exploring the data. Basic statistical descriptions provide the analytical foundation for modelling the data, since it provides the context needed to develop the model and interpret its results correctly.

- Data Preparation consists of data preprocessing. Usually the raw data collected tends to be incomplete, noisy, and inconsistent, and so it cannot be applied directly into building the model, hence it is paramount to process the data. This process may be repeated several times, depending on the quality of the data and the accuracy of the model obtained. The preprocessing of the data might include data cleaning, data integration, data selection and reduction, and data transformation [26].

- Modeling involves choosing the modelling techniques and algorithms, as well as the tools to create the model. Once the model is chosen and built, the following step consists of evaluating its performance.

- Deployment integrates a machine learning model into an existing production environment to make practical business decisions based on data.

The CRISP-DM's longevity, especially in a rapidly changing area stems from a number of characteristics. Firstly, it encourages to focus on business goals, so as to ensure that project outputs provide tangible benefits to the organization. Too often, analysts can lose sight of the ultimate business purpose of their analysis and this approach helps to ensure that the business goals remain at the centre of the project. Secondly, it provides an iterative approach, including frequent opportunities to evaluate the progress of the project against its original objectives, which helps minimize the risk of getting to the end of the project and finding that the business objectives have not really been addressed. It also means that the objectives of the project can be adapted in the light of new findings [66].

1.5 Dissertation Structure

The dissertation is organized as follows. Chapter One presents the role of big data in retail, a brief description of the company and the main goal of the project, as well as the project approach. Chapter Two introduces the Net Promoter Score, as well as different customer related concepts of experience. Chapter Three comprises an extensive literature review on the methodological framework of this project. Chapter Four reports the steps taken in finding the right data and the outcomes of the exploratory analysis performed on it. Chapter Five describes the use of the machine learning methods for modelling and the results obtained, as well as an error analysis. And finally, Chapter Six presents conclusions of the work done and gives suggestions for future work in this area.

1.6 Notation

In supervised learning, an algorithm is fed with a data set $E = \{E_1, E_2, ..., E_N\}$ with N observations $E_i = (x_i, y_i)$ where $x_i = (x_{i1}, x_{i2}, \cdots, x_{iM})$. Each observation E_i has an associated label y_i ($i = 1, \cdots, N$) that defines the class the observation belongs to.

Table 1.1 represents a data set with N observations (rows) and M variables (columns). For classification problems, the class variable Y is a qualitative variable that may assume a set of K classes $C = \{C_1, C_2, \cdots, C_K\}$ [6].

The notation used throughout this paper is summarized in Table 1.2.

Table 1.1: Data set in the attribute-value form. Adapted from [6]

	A_1	A_2	$\cdots$	A_M	Y
E_1	x_{11}	x_{12}	$\cdots$	x_{1M}	y_1
. E_2	x_{21}	x_{22}	$\cdots$	x_{2M}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\cdots$	x_{NM}	y_N

Table 1.2: Commonly used symbols and terms.

Symbol	Term
A_m	Variable $m = \{1, \cdots, M\}$ (explanatory variable)
E_i	Observation $i = \{1, \cdots, N\}$
C_k	Class $k = \{1, \cdots, K\}$
K	Total number of classes
Y	Class variable $Y = \{y_1, \cdots, y_N\}$ (response variable)
y_i	Class label of observation i
N_{C_k}	Number of observations in class k
N	Total number of observations $N = N_{C_1} + \cdots + N_{C_k}$
D	Training data set

Chapter 2

Measuring Customer Experience

This chapter introduces the Net Promoter Score (NPS) metric, the research literature related to it and the different concepts of experiences - customer experience and customer satisfaction. The concepts are well established in business literature, and while they are often used as synonyms there are differences when using such terms in customer related discussions.

2.1 Net Promoter Score (NPS)

In recent years, researchers have suggested a number of customer metrics to illustrate the connections between customer behaviour and company growth. Reichheld [55] proposed the concept of Net Promoter Score (NPS) in the Harvard Business Review as one such metric. Nowadays, the NPS is a widely used customer metric across service industries and over other industries across the world.

The author claimed that only one question based on the willingness of a customer to make a recommendation to friends or colleagues would determine customer loyalty and consequently the company's growth. The calculation of the NPS is quite straightforward, the customers are asked the question: "How likely is that you would recommend company X to a friend or colleague?", on a scale of 0 to 10, where 0 means “Not at all Likely” and 10 means “Extremely Likely”. The questions are captured and the customers are then grouped into Promoters (9–10 rating), Passives (7–8 rating) and Detractors (0–6 ratings) (Table 2.1). Since it is the Promoters and the Detractors who will influence the company's reputation, the NPS is calculated by subtracting the percentage of Detractors from the percentage of Promoters (Equation 2.1).

Table 2.1: Net Promoter Score (NPS) customer categorization.

Ratings	NPS Label
9-10	Promoter
7-8	Passive
0-6	Detractor

$$NPS = \%Promoters - \%Detractors. \tag{2.1}$$

Promoters are thought to be extremely satisfied customers and therefore are the most likely to promote the company, service or product. On the other hand, Detractors are the segment of the customers who are likely to detract the company, service or product and are thought to be associated with negative experiences, which might cloud a recommendation. At a neutral point, Passives are neither satisfied nor disappointed with the company, but they can begin to hint at room for improvement [36].

According to Reichheld [56] the willingness to promote is a strong indicator of loyalty and consequently of the growth of the company because the customers put their reputation at stake when they are recommending and they will only do that when they are intensely loyal. The author defines loyalty as the "willingness of someone – a customer, a friend, an employee, a friend – to make an investment or personal sacrifice in order to strengthen a relationship". Loyal customers are more likely to make repeat purchases and use services that provides good value in the long term. Word of mouth (WOM) is one of the most effective promotion tools ([13]), and so loyal customers will also benefit the company by spreading the word for attracting new customers. In that sense the growth of the company would be proportional to the number of Promoters the company is having.

More than a numerical score used to measure the company's performance, the NPS has evolved into a system. According to the authors in [55], to formulate a working system one needs three fundamental elements: systematic data collection process, closed loop learning process and organizational goal to get more Promoters than Detractors. Accordingly, the system can help to transfer companies to become more costumer focused organization. Systematically collecting data and analyzing the NPS enables the information to be communicated through the organization. A closed loop learning regards the learning related to the collected data, and, consequently, the process of building improvements through that knowledge. Finally, leaders of the organization have a responsibility to create a mission for more Promoters and fewer Detractors.

Regularly learning to use the score and related feedback drives improvement and guides the organization to get better results towards greatness. The organization must qualify to understand and take actions in practice to fulfill the three elements of the working system and process.

Whereas the NPS question provides valuable numerical data, it should be complemented with other questions in order to explain and deepen the understanding of the score. The authors in [55] advise companies to have at least this essential follow up question: "What is the primary reason for your score?". Other example of follow-up question is to formulate the question based on the NPS questions score. For instance, asking the Detractors "What is the most important improvement that would make you more likely to recommend us?" or the Promoters "What is the most important aspect for your recommendation?". The answers of the follow-up questions will provide initial diagnosis of the root cause or reason for the particular score. Simultaneously, the diagnosis enables the company to react internally and potentially reach to the customer for further diagnosis, evaluation and responses. Together with the categorization, finding answers to the questions why and how the numbers vary over time, the company is able to develop a closed loop process and formulate a mission to improve its business model, service or products.

2.2 Customer Experience

Customer Experience (CX) was first introduced in [8] and has since evolved as a key differentiation strategy across industries. Traditional sectors like telecom operators, retail stores, banks, hotels, etc, recognize CX as part of their commercial success. The evolution in business world towards consumer-centric thinking, understanding the needs and wants of consumers while providing a good experience has becoming pivotal for companies' success. Understanding the entire customer experience helps the company to adjust and develop its services to satisfy customers better while building stronger loyalty relationships with the company.

Customer experience can be defined as a whole event that a customer comes into contact with when interacting with a certain business. The experience occurs when the interactions take place through the stimulations of goods and services consumed, affecting the emotions of the customer [31]. There are multiple definitions of CX throughout the literature, but they all indicate that the experience is strictly personal and implies the customer's involvement at different levels (rational, emotional, sensorial, physical, and spirititual) [22].

Positive experiences increase the chances of a customer to make continued purchases and develop brand loyalty, which in turn can make customers advocate resulting long term relationships between parties. Loyal and advocate customers engage in positive WOM activities influencing potential customers to develop opinions from another's experiences [57].

2.3 Customer Satisfaction

The concept of customer satisfaction can be seen overlapping with customer experience. However, customer satisfaction can be described as the attitudes or feelings that customers form based on their experiences with a company. As the expectations of the customers are met and surpassed, the customers tend to be satisfied [29].

According to [4] customer satisfaction is linked to customer loyalty, which in turn is related to profitability. It can be said that the more satisfied the customers are, more loyal and more profitable they tend to be for businesses.

As in customer experience, understanding better the customer satisfaction can help companies to evaluate their ability in meeting customer's needs and expectations, whilst identifying areas in need for development.

Chapter 3

Methodological Framework

This chapter comprises an extensive literature review on the methodological framework of this project, which can be divided into three topics. The first topic covers data preparation and transformation, where the steps necessary to extract, prepare and transform the data are detailed. The second topic is related to data mining techniques used for NPS classification. And, lastly, there is a review on evaluation metrics.

3.1 Data Preparation

Low-quality data will lead to low-quality mining results. And as real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size and likely multiple sources, it is paramount to process the data before applying data mining techniques [26].

Data preparation can be referred to as the set of techniques that initialize the data properly to serve as input for a Data Mining (DM) algorithm. The methods for data preparation can be summarized as data cleaning, data integration, data reduction, and data transformation.

Data cleaning routines deal with missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. This process is usually performed as an iterative two-step process of discrepancy detection followed by data transformation. That is once discrepancies are detected, transformations are applied to correct them.

Data integration combines data from multiple sources to form a coherent data store. This process must be carefully performed in order to avoid redundancies and inconsistencies, resulting in a decrease of the accuracy and speed of the subsequent DM processes.

Data reduction comprises the set of techniques that obtain a reduced representation of the original data. A form of data reduction is variable selection. Reducing the dimensionality of the data by eliminating inappropriate variables improves the performance of learning algorithms and produces a more compact representation. Moreover it enables an easier interpretation of the target concept, making the user focus on the most relevant variables [76].

In data transformation, the data is transformed or consolidated into forms appropriate for mining. Some strategies for data transformation include smoothing, variable construction, normalization and discretization. Smoothing works to detect and remove noise from the data through techniques such as regression and clustering. Variable construction constructs and adds new variables to the given dataset to help the mining process. Normalization scales variables so as to fall within a smaller range, such as -1 to 1, or 0 to 1. This technique attempts to give all variables an equal weight. Discretization transforms quantitative data into qualitative data, in a way that numeric variables (*e.g.*, age) are replaced by interval labels (*e.g.*, 0–10, 11–20, etc), or conceptual labels (*e.g.*, youth, adult, senior) [21].

It is relevant to note that there is much overlap between the major data preprocessing tasks. For instance, smoothing is also a form of data cleaning, as it removes noise from the data.

The need for data preparation is determined not only by the type of model but also by the databases in which the model will be used. As a result, there are a variety of data preparation methods and many possible combinations between them. The choice among these alternatives can influence the quality of the result of the mining. For instance, some procedures, such as tree-based models are notably insensitive to the characteristics of the predictors [24].

3.1.1 Data set Balancing

Imbalance data sets are characterized by a rare class, also called minority class, which represents a small portion of the entire population. Class imbalance can be intrinsic to the problem or it can be determined by the limitation of data collection, caused by economic or privacy reasons. In such scenario almost all the observations belong to a specific class (majority class) and the minority class is scarce, as their own patterns, but those information is extremely important for the trained model to discriminate the small samples from the rest.

It has been shown that in a classification problem, in case of class imbalance, training the classifier using conventional classification techniques results in higher performance, however the classifier gets overwhelmed by the majority class and consequently ignores and misclassifies the minority one since there are not enough examples to recognize the patterns and the properties of the minority class [3]. Hence the importance of firstly handle class imbalance when designing models.

The Imbalance Ratio (IR) is a commonly used measure to describe the imbalance extent of a dataset, and can be defined as:

$$IR = \frac{N_{Maj}}{N_{Min}} \tag{3.1}$$

where N_{Maj} is the sample size of the majority class and N_{Min} is the sample size of the minority class. When there are multi-classes, i.e. the number of classes is larger than 2, N_{Maj} is the sample size of the largest class and N_{Min} is the sample size of the smallest class. The larger the IR, the larger the imbalance extent of the dataset [79].

In literature as well as in real applications, class imbalance problem is mainly handled by three methods: data preparation approach, algorithmic approach and Variable selection approach. In this work, only the first approach in data preparation will be exploited.

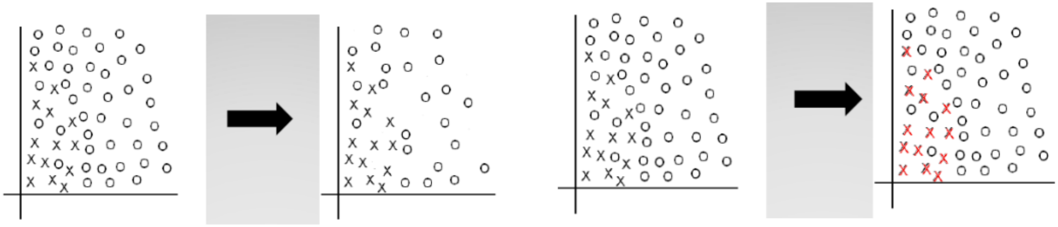
The first approach on data preparation refers to resampling. This method is used before the training of the learning model and it aims at changing the class distribution. There are two types of resampling: natural and artificial.

In natural resampling the main goal is to collect more data of the minority class. Which is not always possible, especially in cases where the imbalance problem is part of its own nature.

As for artificial resampling, there are strategies that preprocess the given imbalanced data set, changing the data distribution. These methods are convenient since they can be applied to any existing learning tool, and the chosen models are biased to the goals of the user. These strategies are: Random Undersampling (RUS), Random Oversampling (ROS) and Synthetic Minority Oversampling Technique (SMOTE).

Random Undersampling (RUS) balances class distribution through randomly removing observations of the majority class (3.1a). A major drawback is the possible loss of information, since this method can remove potentially useful observations, and thus increase variance [3].

The second method, Random Oversampling (ROS) randomly replicates observations of the minority class that are added to the data (3.1b). As this approach makes exact copies of the minority class, it increases the likelihood of overfitting, which occurs when the model fits exactly against its training data. Moreover, it may also decrease the classifier performance and increase computational effort [6].



(a) Randomly removes the majority sample.

(b) Replicate the minority class samples.

Figure 3.1: (a) Random Undersampling and (b) Oversampling techniques. Source: [42]

Lastly, there is the Synthetic Minority Oversampling Technique (SMOTE) [15]. This technique addresses the overfitting issue by generating new variables of the minority class by interpolating between several minority class observations and their nearest neighbours. It also causes the decision boundaries for the minority class to spread further into the majority class space [6]. The SMOTE technique is a standard procedure in the classification of imbalanced data and is experimentally evaluated on a variety of datasets with various levels of imbalance and different size data [3].

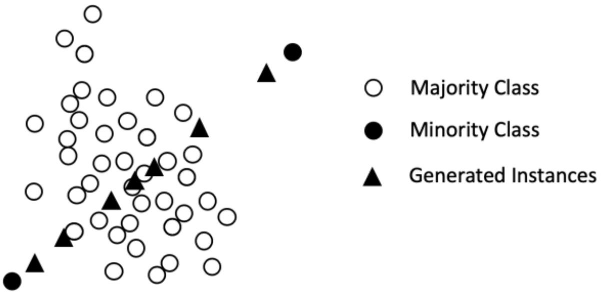


Figure 3.2: Example of minority class oversampled by SMOTE. Adapted from [19].

When applying the SMOTE method to the imbalanced data set, there are several distance metrics which can be applied to calculate the distance between the minority class observations and their nearest neighbors, such as the Euclidean, Manhattan and Canberra distance for numerical variables, Overlap metric for nominal variables, as well as Heterogeneous Euclidean-Overlap Metric (HEOM) and Heterogeneous Value Difference Metric (HVDM) for dealing with both nominal and numeric variables.

The HEOM distance function uses different distance functions on different types of variables. One approach is to use the Overlap metric for nominal variables and normalized Euclidean distance for linear variables. If either of the variables values is unknown, they are handled by returning a distance of 1, which is equivalent to the maximal distance. The distance between two values x and y of a given variable a can be defined as:

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown, else} \\ overlap(x,y), & \text{if } a \text{ is nominal, else} \\ rn\_diff_a(x,y) \end{cases} \tag{3.2}$$

The function overlap and the range normalized difference rn_diff are defined as:

$$overlap(x,y) = \begin{cases} 0, & \text{if } x = y \\ 1, & \text{otherwise} \end{cases} \tag{3.3}$$

$$rn\_diff_a(x,y) = \frac{|x-y|}{range_a} \tag{3.4}$$

The value $range_a$ is used to normalize the variables, and can be defined as:

$$range_a = max_a - min_a \tag{3.5}$$

where max_a and min_a are the maximum and minimum values, respectively, observed in the training set for variable a [75].

The definition for d_a returns a value which, typically, ranges between 0 and 1, whether the variable is nominal or linear. The overall distance between two input vectors x and y is given by the HEOM function:

$$\text{HEOM} = \sqrt{\sum_{a=1}^{M} d_a(x_a, y_a)^2} \tag{3.6}$$

where M is the number of variables.

The HVDM function is very similar to the HEOM function. It also returns the distance between two input vectors x and y, however it uses Value Difference metric instead of an Overlap metric for nominal values and it normalizes differently. The HVDM can be defined as:

$$\text{HVDM} = \sqrt{\sum_{a=1}^{M} d_a^2(x_a, y_a)} \tag{3.7}$$

The function $d_a(x, y)$ returns a distance between the two values x and y for variable a and uses one of two functions, depending on whether the variable is nominal or linear. It can be defined as:

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown, else} \\ normalized\_vdm_a(x,y), & \text{if } a \text{ is nominal, else} \\ normalized\_diff_a(x,y), & \text{if } a \text{ is linear} \end{cases} \tag{3.8}$$

The $normalized\_vdm$ and $normalized\_diff$ functions can be described as followed:

$$normalized\_vdm_a(x,y) = \sqrt{C * \sum_{k=1}^{K} \left| \frac{N_{a,x,k}}{N_{a,x}} - \frac{N_{a,y,k}}{N_{a,y}} \right|^2} \tag{3.9}$$

where

- $N_{a,x}$ is the number of observations in the training set (D) that have value x for variable a;
- $N_{a,x,k}$ is the number of observations in D that have value x for variable a and output class k;
- K is the number of output classes in the problem domain.

and

$$normalized\_diff_a(x,y) = \frac{|x-y|}{4\sigma a} \tag{3.10}$$

where σa is the standard deviation of the numeric values of variable a [75].

The SMOTE approach may be especially problematic in the case of highly skewed class distributions where the minority class observations are very sparse, thus resulting in a greater chance of class mixture (Figure 3.2) [10]. Moreover, the opposite can also happen since interpolating minority class observations can expand the minority class clusters, introducing artificial minority class observations too deeply into the majority class space [6].

Several variations and improvements of this method exist, such as SMOTE+Tomek Links, which creates better defined class clusters.

Two variables E_i and E_j form a Tomek Link [69] if only they belong to different classes and are each other nearest neighbours [9]. Namely, if $d(E_i, E_j)$ is the distance between (E_i, E_j), then a (E_i, E_j) pair is called a Tomek Link if there is not a variable E_l such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$. If two variables form a Tomek link, then either one of these variables of the majority class is eliminated or both variables are eliminated.

The SMOTE+Tomek Links approach applies Tomek links to the oversampled training dataset. Thus, instead of removing only the majority class variables that form the Tomek links, variables from both classes are removed [6]. The application of this method is illustrated in Figure 3.3. This method is applicable when the data sets are highly imbalanced or there are a few variables of the minority class [16].

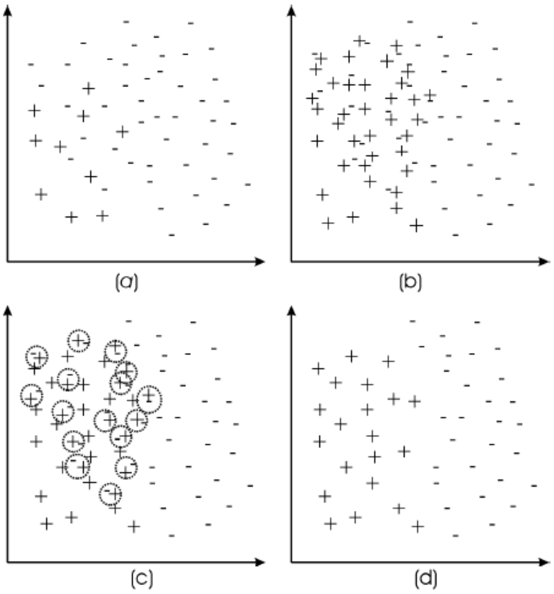


Figure 3.3: SMOTE+ Tomek Links: (a) original data set; (b) oversampled data set with SMOTE; (c) Tomek links identification; (d) Tomek links removed, producing a balanced data set with well defined class clusters. Reprinted from [6].

3.2 Data Mining Techniques for NPS Classification

The data that businesses collect about their customers is one of their greatest assets. Within the vast amount of collected data there is valuable information that can make a difference in the way companies run their businesses, and the way they interact with their current and prospective customer, which can lead them to gain edge on the competitors [2].

Data mining is an important component of analytic in customer relationship management, it can extract hidden predictive information from large databases to predict future trends and behaviours, allowing businesses to make knowledge-driven decisions. In a business perspective, the goal of data mining is to allow a corporation to improve its marketing, sales, and customer support operations through a better understanding of its customers. The result is happy, loyal customers and profitable businesses [41].

Data mining techniques for NPS prediction are divided in two main approaches: supervised and unsupervised methods. Both of these approaches are based on training an algorithm with a record of observations from the past. Supervised methods require that each of those observations used for learning has a label with the class it belongs to. In the context of NPS prediction, this means that for each observation we know if it belongs to the class Promoters, Passives or to the class Detractors. Supervised methods attempt to discover the relationship between independent variables (input variables) and a dependent variable (target variable), and, as a prediction model, they can later be used for predicting the value of the target variable whenever the values of the input variables are known [58].

As an important data mining technique, prediction is a supervised method that predicts values of certain variables in unknown situations from historical data. There are two main prediction models: Classification models and Regression models. The difference between them is that Classification uses categorical or discrete data as the target variable, whereas Regression uses numerical or continuous data [24].

As a highly application-driven domain, data mining has incorporated many techniques from other domains such as statistics and machine learning. Statistics studies the collection, analysis, interpretation, and presentation of data, and it has an inherent connection with data mining. Statistical models are widely used to model data and data classes. For instance, in data mining tasks like classification, statistical models of target classes can be built and, subsequently, be the outcome of a data mining task. Furthermore, statistics is useful for mining various patterns from data as well as for understanding the underlying mechanisms generating and affecting the patterns. Machine Learning (ML) investigates how computers can learn or improve their performance based on data. There are a number of ML techniques available for classification modelling, such as decision trees, random forests, neural networks, among others [26].

Classification is probably the most common data mining activity today, and it can be described as a procedure that given a data set E and an unclassified observation E_i it assigns a class label y_i to E_i.

Data classification consists of a two-step process. Firstly, there's a learning step, where a classification model, also known as classifier, is built by analyzing a training dataset (D) made up of database observations (E_i) and their associated class labels (y_i).

In the second step, the model is used for classification. First, the predictive accuracy of the classifier is estimated using a test dataset. The test dataset is made up of test observations from the original dataset and their associated class labels. Such observations are independent of the training ones, meaning that they were not used to construct the classifier. The test set is used to measure the performance of the classifier, if it is considered acceptable, the classifier is then used to classify future data observations for which the class label is unknown [26].

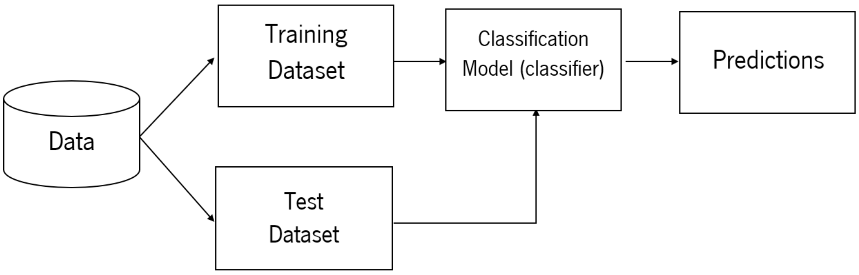


Figure 3.4: Classification process.

Generally, there are three main types of classification tasks: binary, multi-class and multi-label. Binary classification refers to those classification tasks with two class labels. For example, these can be used in churn detection in order to predict whether a given customer is going to churn or not, in email spam detection, cancer detection, etc. Typically, these tasks involve one class that is the normal state, and usually assigned the class label 0, and another class that is the abnormal state, and assigned the class label 1.

Multi-class classification tasks have more than two class labels. Unlike binary classification, multi-class classification does not have the notion of normal and abnormal outcomes. Instead, variables are classified as belonging to one among a range of known classes. Examples include plant species classification, face recognition, etc.

Multi-label classification refers to those classification tasks that have two or more class labels, where one or more class labels may be predicted for each variable, unlike binary and multi-class classification where a single class label is predicted for each variable. Considering an example of photo classification, a given photo may have multiple objects in the scene and a model may predict the presence of multiple known objects in the photo, such as a bicycle, apple, person, etc [12].

In NPS classification, the customers are segmented into Promoters, Passives or Detractors, therefore it sets our problem into a 3-class classification problem.

There is not a method that single-handedly works best for every data set. Hence it is an important task to decide for any given set of data which method produces the best results. Selecting the best approach can be one of the most challenging parts of performing statistical learning in practice. Two important concepts that arise in selecting a statistical learning procedure for a specific data set are the train and test error rate. Let it be represented $\hat{f}(x)$ as the prediction that $\hat{f}$ gives for the ith observation (E_i). The training error rate is the proportion of mistakes that are made if we apply $\hat{f}$ to the training observations:

$$\frac{1}{N}\sum_{i=1}^{N} I(y_i \neq \hat{y}_i) \tag{3.11}$$

Here $\hat{y}_i$ is the predicted class label for E_i using $\hat{f}$. And $I(y_i \neq \hat{y}_i)$ is an indicator variable that equals 1 if $y_i \neq \hat{y}_i$ and zero if $y_i = \hat{y}_i$. If $I(y_i \neq \hat{y}_i)$ = 0 then E_i was classified correctly by our classification method; otherwise it was misclassified.

Equation 3.11 refers to the training error rate because it is computed based on the data that was used to train the classifier. The test error rate associated with a set of test observations of the form (x_0, y_0) is given by:

$$Ave(I(y_0 \neq \hat{y}_0)) \tag{3.12}$$

where $\hat{y}_0$ is the predicted class label that results from applying the classifier to the test observation with predictor x_0. A good classifier is one for which the test error 3.12 is smallest [35].

There are many classification techniques, the most common include decision trees, neural networks, logistic regression, discriminant analysis and emerging tools such as rough sets and support vector machines, among others [62]. In [51], the authors provide an academic database of literature, published between 2000 and 2006, and propose a classification scheme to classify the articles. Their aim was to give a research summary on the application of data mining and determine which techniques are most often used in the customer relationship management domain. With respect to the research findings, they have concluded that classification models, in particular neural networks and decision trees are the most commonly applied algorithms for predicting future customer behaviour.

3.2.1 Decision Tree

Decison Trees (DT) are one of the most popular supervised learning algorithms in data mining, and its main objective is to construct a training model that can be used to predict the class or value of target variables through learning decision rules inferred from the training data. Their popularity is mostly due to their simplicity and transparency, since they are easier to interpret than other techniques [58].

The decision tree algorithm creates a hierarchical partitioning of the data that can be summarized in a tree. The basic structure of a DT consists of nodes that form a rooted tree, meaning it is a directed tree with a node called root that has no incoming edges (or branches), whilst all other nodes have exactly one incoming edge. A node with outgoing edges is called an internal node. All other nodes are called leaves. Moreover, a node that gets divided into sub-nodes is also known as parent node, and these sub-nodes are known as child nodes (Figure 3.5).

Variables are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path, and then assigned to one class representing the most appropriate target value [65]

The overall approach is to try to recursively split the data so as to maximize the discrimination among the different classes over different nodes [1]. Basically all decision tree algorithms require splitting criteria for splitting a node to form a tree, where the main aim is to reduce the impurity of a node. A partition is pure if all the observations in it belong to the same class. There are many criteria by which node impurity is minimized in a classification problem. Some commonly used metrics include: Classification error rate, Entropy, Gini index, Information Gain and Gain ratio [26].

Classification error rate is simply the proportion of observations misclassified over the whole set of observations, which in DT corresponds to the proportion of observations that do not belong to the most common class. However, it turns out that classification error is not sufficiently sensitive for tree-growing, and in practice other measures are preferable.

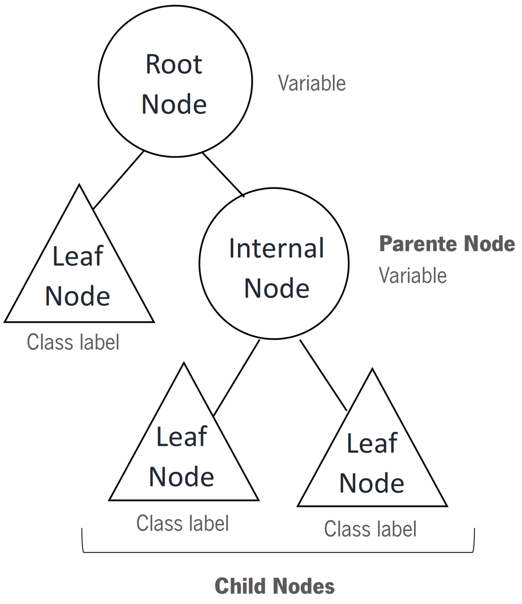


Figure 3.5: Decision Tree structure.

Entropy is the measure of impurity of a node. It can be written as:

$$Entropy = -\sum_{k=1}^{K} \hat{p}_{nk} \log \hat{p}_{nk}. \tag{3.13}$$

Here $\hat{p}_{nk}$ represents the proportion of training observations in the nth node that are from the kth class. Since $0 \leq \hat{p}_{nk} \leq 1$, it follows that $-\hat{p}_{nk}\, log\, \hat{p}_{nk} \geq 0$. The value of this metric lies between 0 and 1, and the closer its value to 0 the better [35].

The Gini index is another impurity-based metric that measures the total variance across the K classes (Equation 3.14)

$$GiniIndex = \sum_{k=1}^{K} \hat{p}_{nk} \left(1 - \hat{p}_{nk}\right) \tag{3.14}$$

This metric takes on a small value if all of the $\hat{p}_{nk}$'s are close to zero or one. For this reason in the Gini index, like entropy, a small value indicates that a node contains predominantly observations from a single class [35].

Information gain uses the entropy measure as the impurity measure, and it is defined as the difference between the entropy of the node before splitting (parent node) and after splitting (child node) [37].

$$InfoGain(\bigtriangledown) = entropy\,(parent\,node) - entropy\,(child\,node) \tag{3.15}$$

The impurity-based criterions described above can be biased towards Variables that have large number of distinct values. Namely, they prefer input Variables with many values over those with less values. For that reason, sometimes it is useful to "normalize" the impurity-based measures [35].

The Gain Ratio normalizes the information gain, and is determined as:

$$Gain\,Ratio = \frac{Information\,Gain(\bigtriangledown)}{Entropy} \tag{3.16}$$

Usually, either entropy or the Gini index are typically used to evaluate the quality of a particular split, since these two approaches are more sensitive then others.

The Gini Index may encounter problems when the domain of the target variable is relatively wide [65]. In such cases it is possible to employ a binary criterion called Twoing criteria. This criteria is designed for multiclass problems and favors separation between classes. Every multiclass split is treated as a binary problem. Splits that keep related classes together are favored. The approach offers the advantage of revealing similarities between classes [49].

There are many DT algorithms available in Data Mining, the most common are Classification and Regression Trees (CART) [11], Iterative Dichotomiser 3 (ID3) [53] and C4.5 [54] which use different splitting criteria for splitting the node at each level to form a homogeneous node. For instance, CART uses the Gini index, whereas ID3 uses information gain (entropy) and C4.5 the Gain Ratio.

When building a classification tree, the splitting phase continues until a stopping criterion is triggered. Some common stopping rules conditions are listed below [65]:

1. All variables in the training set belong to a single value of Y.
2. The maximum tree depth has been reached.
3. The number of observations in the terminal node is less than the minimum number of observations for parent nodes.
4. The best splitting criteria is not greater than a certain threshold.

A tree can be grown to be quite large, almost to the point where it fits the training data perfectly, resulting in overfitting. The model that overfits predicts very well the training data but poorly on independent test sets, resulting in poor predictions. One way to prevent overfitting is to limit tree growth by setting some rules at the beginning, before overfitting occurs [1]. A preferred approach is to grow an overly large tree until some minimum node size is reached, and then prune the tree back to an optimal size. Optimal size can be determined using an independent test set or cross-validation [49].

Although some research papers state that pruning might be helpful with imbalanced data sets in some circumstances [14], other papers conclude that it should be avoided. Most pruning schemes, including the one used by C4.5, attempt to minimize the overall error rate. And so they can be detrimental to the minority class, since reducing the error rate in the majority class, which stands for most of the observations, would result in a greater impact over the overall error rate. On the contrary, it still seems to be unclear if pruning can lead to a performance improvement for decision trees grown over artificially balanced data sets [6].

Decision trees are effective forms to represent and evaluate the performance of algorithms due to its simplicity, comprehensibility, no parameters, and being able to handle mixed-type data [65]. But despite its popularity, this technique also presents some disadvantages: the optimal decision-making mechanism can be deterred and incorrect decisions can follow; it needs a large dataset to construct a good performance classifier; the possibility of overfitting which results in high variance; and they're also relatively expensive, from a computational perspective, as they need to identify splits for multiple variables [37].

The basic characteristics of the three algorithms mentioned earlier, CART, ID3 and C4.5, are explained in Table 3.1. Among all the available learning algorithms of DT, the CART strategy is known to be one of the most successful techniques [23]. Thereby this study will employ the CART algorithm.

Classification and Regression Trees (CART)

CART is a nonparametric procedure that stands for Classification and Regression Trees [11]. It is characterized by the fact that it develops binary trees, namely the parent nodes are always split into two child nodes, and the process is repeated by treating each child node as a parent node. This is called binary recursive partitioning. This process is repeated until further partitioning is impossible or is limited by some criterion. Once the first terminal node has been created, the algorithm repeats the procedure for each set of observations until all observations are categorized

Table 3.1: Basic characteristics of three common decision tree algorithms. Source [37].

Algorithm	Splitting criteria	Variable type	Missing values	Prunning strategy	Outlier detection
CART	Gini Index Twoing criteria	Handles both categorical and numeric values	Handles missing values	Cost-complexity pruning	Can handle outliers
ID3	Information Gain	Handles only categorical values	Does not handle missing values	No pruning	Susceptible on outliers
C4.5	Gain Ratio	Handles both categorical and numeric values	Handles missing values	Error based	Susceptible on outliers

as terminal nodes [73].

Although different splitting criteria can be used in decision trees, and in the CART algorithm, in this study, the splits are selected using the Gini index criteria, which is the default method and the one that usually performs best [73].

When overfitting occurs, the obtained tree is pruned by Cost–complexity pruning. This prunning algorithm is parameterized by a (Complexity Parameter (CP)) α ($\geq$0), which assesses the 'cost' of adding another variable to the model. This parameter is used to define the cost-complexity measure $R_\alpha(T)$ [68]:

$$R_\alpha(T) = R(T) + \alpha|T|. \tag{3.17}$$

Here $|T|$ is the number of terminal nodes in a given tree T and $R(T)$ is the risk of T, the total misclassification rate of the terminal nodes.

The CART algorithm has the advantage of handling both numerical and categorical variables; it can identify the most significant variables and eliminate non-significant ones, and also its performance is not considerably affected by the presence of outliers in the input parameters [65]. On the contrary, small modifications of the learning sample can result in an unstable decision tree, and imbalanced classes may result in under-fitted trees [23].

3.2.2 Random Forest

While an individual tree is overfit to the training data and is likely to have a large error, bagging uses the insight that a suitably large number of uncorrelated errors average out to zero to solve this problem.

It is well known that averaging a set of observations reduces variance, hence a natural way to reduce the variance and increase the test set accuracy of a statistical learning method is to take many training sets from the population, build a separate prediction model using each training set, and average the resulting predictions. However this is not practical since we generally do not have access to multiple training sets.

Bagging chooses multiple random samples of observations from the training data, with replacement, constructing a tree from each one. Since each tree learns from different data, they are fairly uncorrelated from one another. But let us suppose that there is one strong predictor in the data set, along with other moderate predictors. In the collection of bagged trees, most or all of the trees would use the strong predictor in the top split and look quite similar to each other. Random forests overcome this problem by forcing each split to consider only a subset of the predictors [35].

The Random Forest (RF) algorithm, first introduced by [11], is often a collection of hundreds to thousands of bagged trees, where each tree is grown using a bootstrap sample of the original data, introduction a first level of randomization. Using bootstrap samples means that instead of training on all the observations, each tree of RF algorithm is trained on a subset of the observations, more specifically, about one-third of the observations are left out of the sample. These are called OOB samples, whereas the chosen subset is called the bag.

A second layer of randomization is introduced at the node level when growing the tree. Each time a split in a tree is considered, a random sample of predictors ($mtry$) is chosen as split candidates from the full set of M predictors. The split is allowed to use only one of those $mtry$ predictors, meaning that the algorithm is not even allowed to consider a majority of the available predictors. The purpose of this two-step randomization is to decorrelate trees so that the forest ensemble will have low variance.

When used for classification, a random forest obtains a class vote from each tree, and then classifies using majority vote, that is the overall prediction is the most commonly occurring majority class among the predictions. [28].

In RF algorithms, the OOB samples are used to get a running unbiased estimate of the classification error as trees are added to the forest. In order to calculate the OOB error rate, first, the response for the ith observation (E_i) is predicted using each of the trees in which that observation was OOB. At the end of the run, the predicted class is the one that got more votes every time E_i was OOB. An OOB prediction can be obtained this way for each of the M observations. Finally, the

OOB error rate is computed through the proportion of times that the predicted class is not equal to the true class of E_i, averaged over all M observations. Overall, the OOB error rate is the average error for each observation of the training data.

The construction of a RF algorithm can be described in the following steps:

1. Draw $ntree$ bootstrap samples from the original data.

2. Grow a tree for each bootstrap data set. At each node of the tree, randomly select $mtry$ out of all M possible variables for splitting. Find the best split on the selected $mtry$ variables

3. Each decision tree will generate an output. Aggregate information from the $ntree$ trees and vote the trees to get predictions for new data.

4. Compute an OOB error rate by using the data not in the bootstrap sample.

The main limitation of random forest is that a large number of trees can make the algorithm too slow. In most real world applications, the RF algorithm is fast enough but there can certainly be situations where run time performance is important and other approaches would be preferred.

3.3 Data Sets for Supervised Classification

The application of supervised classification requires a training set and a test set to validate the classifier's performance. The idea is that some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on 'new' data. Both sets are randomly sampled partitions of the original data, in which the testing data is cleanly separated from the training data. This is the main idea behind the whole class of model evaluation methods called cross validation.

The Holdout method is the simplest kind of cross validation method to implement as it consists of randomly partitioning the given data into two disjoint sets, a train set and a test set. Commonly, two-thirds of the data are allocated to the train set, and the remaining to the test set (Figure 3.6). The classifier fits a model using the training set only, and then predicts the output values for the data in the testing set, which it has never seen before [26].

This method has the advantage of being easily implemented and having a lower computational time compared to other methods. On the other hand, the evaluation depends on which data points end up in the training set and in the test set, and thus the evaluation may be significantly different depending on how the division is made.

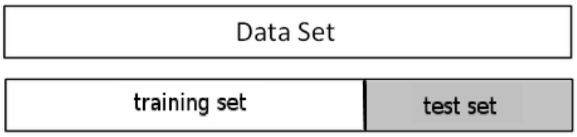


Figure 3.6: Holdout method.

Cross-Validation (CV) also known as k - fold cross-validation is one of the most common methods to estimate the predictive performance of a model and a way to improve over the holdout method. The data set is randomly divided into approximately equal sized k subsets or *folds*, and the holdout method is repeated k times (Figure 3.7) [35]. For each of the test sets the respective training set will be formed by the remaining $k - 1$ partitions. The k − fold cross-validation estimate will be the average of the k individual scores obtained on each test partition [71].

The advantage of this approach is that it matters less how the data gets divided. Every data point is in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as k is increased. However, in this method the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation.

Leave-one-out is a special case of k - fold cross-validation where k is set to the number of initial observations. That is, only one sample is “left out” at a time for the test set [30].

3.4 Performance Evaluation

Evaluating the performance of a model is a fundamental aspect of machine learning, thus a selection of suitable evaluation metrics must be taken into consideration in order to discriminate and obtain the optimal classifier. There are several performance measures for this purpose, such as accuracy, precision, sensitivity and F-measure. There’s no unified generalized metric to assess a classifier’s performance, rather, the best methodology is to use several metrics.

The majority of classification metrics are predefined for binary cases. The outputs of a binary

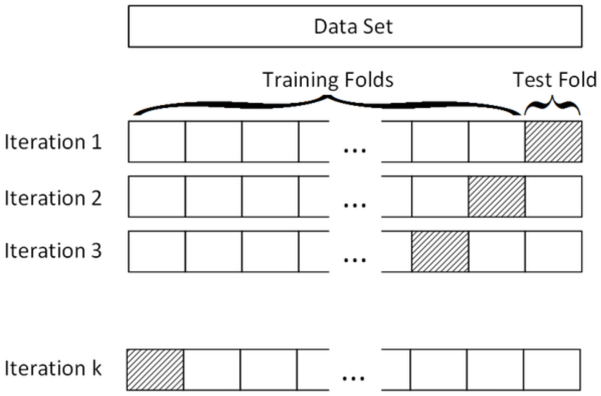


Figure 3.7: K-fold cross-validation. Adapted from [18].

classification problem uses a Confusion Matrix (CM), which is a 2×2 matrix that reports the number of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN).

However, on the problem of predicting the NPS, there are three classes - Promoters, Passives and Detractors. Consequently, the evaluation measures will be applied in a 3×3 multi-class confusion matrix [47] (Table 3.2)

Table 3.2: The NPS classification confusion matrix.

		Predicted Class		
		Promoters	Passives	Detractors
Actual Class	Promoters	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$
	Passives	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$
	Detractors	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$

There are four elements considered during performance measurement. TP are the correctly predicted positive values, whereas the FP are the negative values that are incorrectly labeled as positive. In the NPS classification problem, the applied machine learning algorithms predict the class label, either with success or with misclassification [47], e.g., a customer is either classified as a promoter (positive class) or non-promoter (negative class), which contains the other NPS categories (passives and detractors). The positive class can be one of the three categories of the NPS (Promoter, Passive or Detractor). This way, these values, TP and FP, indicate when a given customer is categorised as the positive class and is classified as true or false, respectively.

Further, TN are the negative values that are correctly labeled by the classifier, whilst the FN are the positive values that are mislabeled as negative. These are the situations where the customers are not the positive class and are classified as false or true, respectively.

On multi-class problems, a way to extend the binary metrics is to use the one-versus-all method. This method brakes down multi-class problems into a series of binary datasets for each class. So, for each dataset one binary classifier will be built, where the data from class C_k is treated as positive, and the data from all the other classes are treated as negative; and their performance is measured using a binary classification metric. Then, to represent these metrics across all classes, averaging techniques are used [50].

There are 3 averaging techniques that can be applied to multi-class classification, such as macro, micro and weighted averaging. The macro approach gives equal weights to all classes, so that there is no distinction between highly and poorly populated classes. The micro averaging approach considers all the units together, without taking into consideration possible differences between classes [25]. It is calculated by dividing the sum of the diagonal cells of the matrix by the sum of all the cells (accuracy). Lastly, the weighted averaging accounts for class imbalance by determining the average of binary metrics weighted by the number of observations of each class in the target. For example, if 3 precision scores for 3 classes are: class 1 (0.85), class 2 (0.80), and class 3 (0.89), the weighted average will be calculated by multiplying each score by the number of observations of each class (N_{Ck}) and dividing by the total number of observations (N) [72].

In general, for imbalanced datasets macro averaging is a good choice when all classes are equally important since it treats all classes equally. The weighted average can also be used for imbalanced datasets when it is prefered to consider the different contributions of each classes.

In the NPS problem we consider the detractors and promoters the important classes. Therefore, since the detractors are the minority class, with the least amount of observations, the weighted method would give the detractors the lowest weight. Ergo, this paper will only exploit the macro method as it gives equal contribution to all classes.

Accuracy

It's good to note that although accuracy is a specific measure, the word "accuracy" is also used as a general term to refer to a classifier's predictive abilities. As explained in [26], given K classes (where $K \geq 2$), an entry, $C_{k,j}$ indicates the number of observations of class k that are labeled by the classifier as class j. For a classifier to have good accuracy, ideally most of the observations would be represented along the diagonal of the CM, from entry $C_{1,1}$ to entry $C_{K,K}$ with the rest of the entries being zero or close to zero.

As for the measure itself, the *accuracy* of a classifier determines the number of correctly predicted class labels in relation to the total number of predictions (Equation 3.18). The accuracy for each class *k* is determined as:

$$\widehat{ACC}(k) = \frac{TN + TP}{FP + TN + FN + TP} \tag{3.18}$$

Conventionally, multi-class accuracy is the average of the different classes' accuracy.

$$\widehat{ACC} = \frac{1}{K}\sum_{k=1}^{K}\widehat{ACC}(k) \tag{3.19}$$

Overall, this is an important measure for evaluating the quality of a model, but it's insufficient, especially in cases where there's an imbalanced distribution of the class. In these situations, accuracy can be misleading since the dataset contains significantly more majority class instances than minorities. [58]. For instance, a classifier would be able to achieve an accuracy of 90% in a dataset where there is a 90/10 distribution of the target variable simply by predicting all values as of the majority class, and such a classifier would not be useful [43].

Sensitivity and Precision

Sensitivity (S), also known as recall, measures the true positive recognition rate, *i.e.*, the proportion of positive values that are correctly identified (Equation 3.20).

$$S = \frac{TP}{TP + FN} \tag{3.20}$$

For multi-class problems, applying the macro method, sensitivity can be determined as:

$$Macro_S = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_{(k)}}{TP_{(k)} + FN_{(k)}} \tag{3.21}$$

where K is the total number of classes.

A perfect sensitivity score of 1 for a given class k means that every item from that class was labeled as such, however it does not give any information on how many observations were incorrectly labeled as belonging to C_k. Overall, sensitivity should be optimized if the aim is to decrease the number of false negatives [26].

Precision (P) is the proportion of observations labeled as positive that are actually such (Equation 3.22).

$$P = \frac{TP}{TP + FP} \tag{3.22}$$

Identical to sensitivity, applying the macro method for multi-class, precision can be determined as:

$$Macro_P = \frac{1}{K} \sum_{k=1}^{K} \frac{TP_{(k)}}{TP_{(k)} + FP_{(k)}} \tag{3.23}$$

A perfect precision score of 1 for a class k means that every observation that the classifier labeled as belonging to class k does indeed belong to that class. However, it does not give any information about the number of class k observations that the classifier mislabeled. In general, the model for precision should be optimized when the goal is to decrease the number of false positives [26].

F1-Measure

Usually, there is a trade off between the precision and sensitivity measures. Trying to improve one measure often results in a deterioration of the other measure. The F_1-measure represents the harmonic mean between sensitivity and precision values, as it gives equal weight to both measures [32].

$$F_1 = \frac{2 \times P \times S}{P + S} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{3.24}$$

This metric ranges between 0 to 1, where the value of 1 indicates perfect precision and sensitivity. When one of the metrics assume values close to 0, the final F score suffers a huge drop, since the harmonic mean tends to give more weight to lower values [25]. The macro averaged F_1-measure is determined by calculating the F_1-measure of every class and then average them.

$$Macro_{F1} = \frac{\sum_{k=1}^{K} F1_k}{K} \tag{3.25}$$

In summary, the accuracy measure works better when the data classes are fairly evenly distributed. The other measures, such as sensitivity, precision, and F_1-measure are better suited upon an imbalanced class problem, where the main class of interest is rare [26].

ROC and AUC

A ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graphs are two-dimensional graphs in which, for a given classifier, the TP rate, also known as sensitivity ($\frac{TP}{TP+FN}$), is plotted on the Y axis and the FP rate ($\frac{FP}{FP+TN}$) is plotted on the X axis. A ROC graph depicts relative tradeoffs between benefits (TP) and costs (FP) [20].

Many binary classifiers, such as decision trees, are designed to produce only a class decision, *i.e.*, a Yes or No on each variable. They are designated as discrete classifiers. When such classifiers are applied to a test data set, they yield a CM, which in return corresponds to one ROC point. Thus, a discrete classifier produces only a single point in ROC space. The point (0, 1) represents perfect classification (Figure 3.8). Informally, one point in ROC space is better than another if it is to the northwest (TP rate is higher, FP rate is lower, or both) of the first. The lower left point (0, 0) represents the strategy of never issuing a positive classification; such a classifier commits no FP errors but also gains no TP. Moreover, the upper right point (1, 1), unconditionally issues positive classifications [20]. In turn, some classifiers, such as a Naive Bayes classifier, naturally yield an instance probability that represents the degree to which an instance is a member of a class. As such, TP and FP can vary as the threshold on the output varies between its extremes (0 and 1). The resulting curve is called the ROC curve [33].

In case of multi-class problems, they can be handled by producing one ROC for each class.

To compare classifiers, a common method is to calculate the Area Under the ROC Curve (AUC), which is the surface between the ROC curve and the horizontal axis. This method is a good measure to reflect quality as ROC curves that are more to the northwest are better and have a bigger surface. AUC values are often used as a single evaluation metric in imbalanced classification problems [1].

The AUC ranges between 0 and 1. When the AUC is approximately 0.5 (called a random classifier), the model has no discrimination capacity to distinguish between positive class and negative class, so no realistic classifier should have an AUC less than 0.5.

One approach to calculating multi-class AUC is described in [64] as:

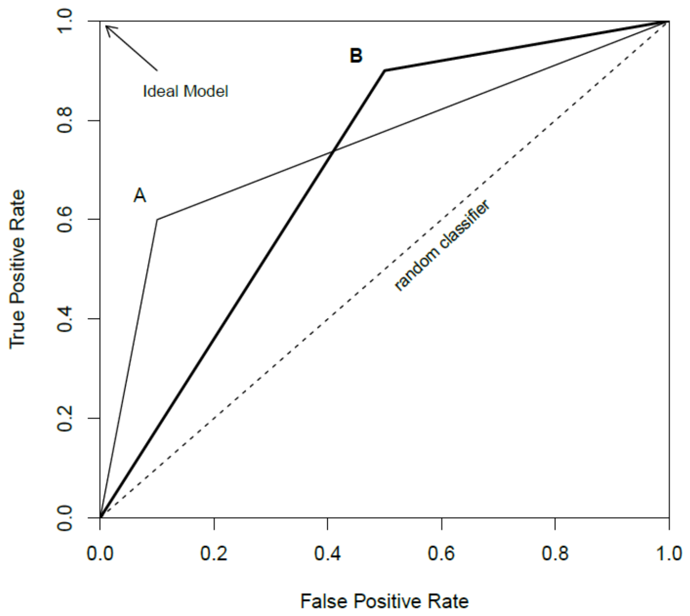


Figure 3.8: ROC curve of three classifiers: A, B and random. Reprinted from [10].

$$AUC_{total} = \sum_{c_k \in K} AUC(c_k) \cdot prev(c_k) \tag{3.26}$$

where K is the total number of classes, $AUC(c_k)$ denotes the AUC for class reference, and $prev(c_k)$ the prevalence of c_k in the data, which can be calculated as $prev(c_k) = c_k/N$ where c_k is the number of positive labels and N the total number of observations.

The AUC has a known statistical meaning: it is equivalent to the Wilconxon test of ranks, and is equivalent to several other statistical measures for evaluating classification and ranking models [6]. In [27], the authors point out that $Gini + 1 = 2 \times AUC$ [20].

Brier Score

The Brier Score is a calibration metric that measures how close the predictions ($\hat{p}$) are to the true classification (p) of an event. This score ranges between 0 and 1. Let $C_j \in C$ be the class j and C the set of all possible classes, for binary classification problems the score is defined as [64]:

$$B\hat{r}ier(C_j) = \frac{1}{N} \sum_{i=1}^{N} (p_j^i - \hat{p}_j^i)^2 \tag{3.27}$$

where p_j^i and $\hat{p_j^i}$ denote the actual C_j outcome (0 or 1) for observation i and respective prediction.

Following the multi-class AUC metric in 3.26, the multi-class Brier score approach is defined as

$$Brier_{total} = \sum_{c_k \in K} Brier(c_k) \cdot prev(c_k) \tag{3.28}$$

The lower the value of the score, the better is the calibration and so the better the model. The perfect model presents a Brier score of 0.

Chapter 4

Data Understanding and Preparation

The NPS is used to measure, understand, and track customer experience. In that sense, the more data we collect and analyse alongside the NPS score, the more we will be able to understand what is driving the customer experience, allowing to prioritise improvements that have the biggest impact on the customers.

The following chapter will detail the procedures of understanding, preparing and analysing the collected data. These procedures were conducted using SQL, R and Python programming languages to extract and merge data from the surveys and Cartão Continente database, followed by data analysis performed through R and Excel.

4.1 Data Set in Study

The data set in study is a combination of both survey data and loyalty card data. Our analysis takes advantage of an individual-level match of Cartão Continente loyalty card transactions using a customer identifier. Observing first-party data from both domains is an unique opportunity to obtain more knowledge about the customer and their purchasing habits.

4.1.1 Data Collection

To build the data set, we decided that the main unit to analyse would be each individual answer of the survey. With that in mind, the objective at this stage was to build a dataframe where each row would correspond to one answer from a customer and each column would represent different

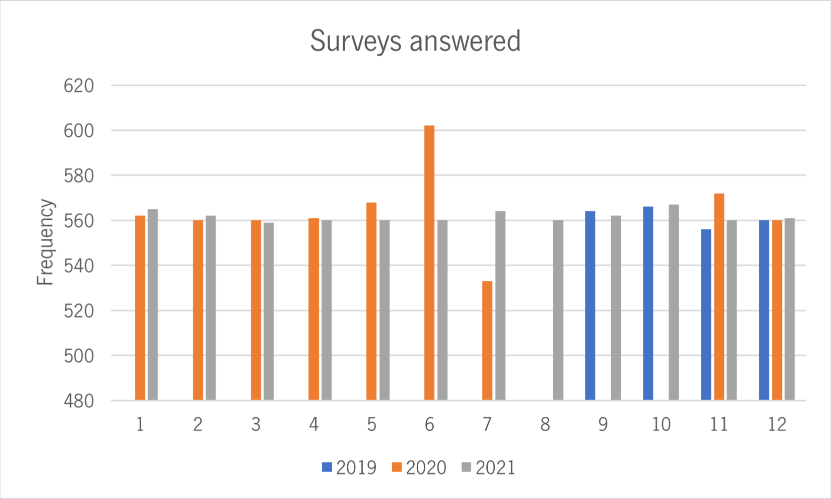


Figure 4.1: Distribution of answered surveys by year.

variables regarding the customer itself.

Survey data

The survey data was collected from September 2019 to December 2021, by an external company. The survey was conducted by calling customers that had recently visited a Note store and asking "To what extent are you willing to recommend a Note store to a friend or family? On a zero - would absolutely not recommend -, to a ten - would absolutely recommend - scale.". Through the survey answers, a data set was built with 44 variables, which can be categorized as: (i) Demographic data, (ii) Store data; (iii) Dates of the inquiry and the customer's last visit to the store; (iv) NPS rating score; and (v) Customer experience.

The resulting data set refers to 25 months with approximately 562 observations each (Figure 4.1). Due to Covid lockdown and consequently stores closing up, there are no observations from August to October of 2020. Each score was converted into a NPS label (see Table 2.1), and a comparison of the NPS trend (Figure 4.2) shows that the stores maintained their NPS levels throughout the years.

It is important to note that the surveys were not structured in an adequate form to reproduce a cohesive analysis since not all questions were included in all years. Thus, the variables that did not have information during the entire period of the survey were removed to proceed with the study. Other variables considered non interesting to the study were also eliminated. After guaranteeing that each customer answered the survey no more than once a month, the survey data set collects

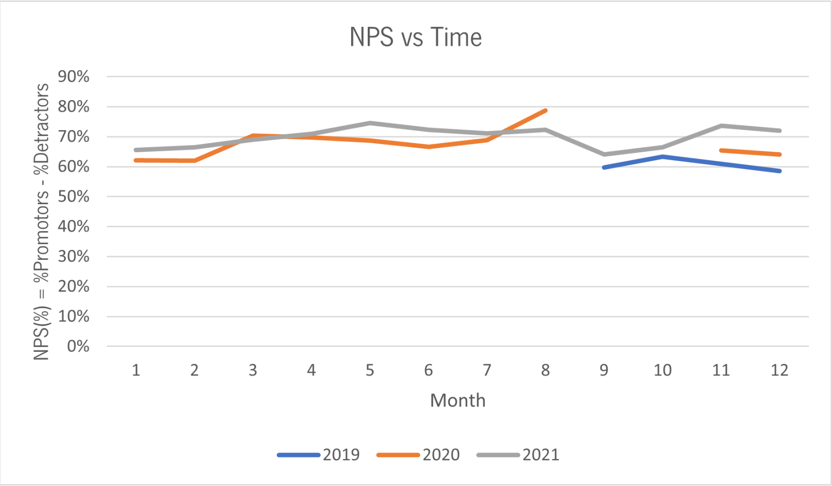


Figure 4.2: NPS trend in relation to time.

13367 eligible observations within 10 variables. A brief description of each variable is presented in Table 4.1.

Table 4.1: Initial data set description.

Total Number of Observations	13367
Initial Variables	Variable Description
id_cliente	Customer's account number
cod_loja	Store's location code as the most recent/previous store visited by the customer
data	The date of the customer's last visit
data_inq	Date of the inquiry
DOP	Operacional Direction - Regional area where the store is located (north, south, Porto, Lisbon, etc)
Concept	Store antiquity concept (old, intermediate, new)
Gender	Gender (M/F)
Age	Age
q2	NPS rating score (0-10)
NPS	NPS classification label (Detractors, Passives and Promoters)

Loyalty card data

The Cartão Continente loyalty card tracks the customer's purchases. The data records timestamp, item code, item hierarchy information, quantity purchased, price listed, price paid, a customer identifier, a transaction identifier, a store identifier, among many other things. We were given access to all the data related to transactions, stores and products in the database of Cartão Continente. However, this data was not organised in one single data set, and so it must be mined for relevant information.

An important aspect of Cartão Continente's loyalty card is that two or more people, usually from the same household, can link their loyalty cards to the same account number. In that sense a card can be considered to be representative of a household and not just of an individual.

The choice of the variables to include in the model depended both on business knowledge, which allowed the selection of the variables that are expected to be more relevant, as well as the availability of information. The chosen variables are categorized as: Customer Profile, Purchase Behaviour and Store Profile. All these variables are related to the Last 12 Months (L12M) before the survey.

Customer Profile (Table 4.2) gathers variables regarding the customer. At this point an issue arose concerning the customer's gender, because, even though customers indicated their gender whilst answering the survey, this information might not be coherent with the information on the customer's account. We know that at the limit, an account can have both genders. In this sense, and based on the customer's transactions, the initial survey-gathered gender variable was replaced by two dichotomous variables, *GenderF* (1=yes, 0=no) and *GenderM* (1=yes, 0= no). So, if an account has a purchase history of female-oriented products then *GenderF* will have the value 1, and, consequently, *GenderM* will have the value 0, and vice-versa. In a case when an account has a purchase history of female and male-oriented products then both variables will have the value 1.

Additionally, information on the customer segments was extracted. The different segmentations profile the customers by providing a holistic view of the customer and their behaviour at Sonae MC.

- The Lifestyle Segmentation (*Slifestyle*), based on the customer's transactions, typifies the customer's motivations and choices - why they choose a store or certain products. There are seven established groups. Groups 1 - Os Saudáveis Exigentes and 2 - Os Urbanos Sofisticados, are more focused on quality and healthy products, groups 3 - Os Pais Práticos, 4 - Os Generalistas Disciplinados, 5 - Os Tradicionais Frequentes, are more focused on family, groups 6 - Os Económicos Focados and 7 - Os Promocionais Atentos, drive their options based on price and promotions.

- Continente Value Segmentation (*Svcontinente*) determines the value of the customer to Continente, taking into account customer loyalty and spending. There are 7 segments: 1 - Loyal Small, 2 - Loyal Medium, 3 - Loyal Large, 4 - Frequent Small, 5 - Frequent Medium, 6 - Occasional Small and 7 - Occasional Medium.

- Note Value Segmentation (*Svnote*), like Continente Value Segmentation, determines the value of the customer to Note. This segmentation has 8 segments: 1 - Loyal Medium, 2 - Frequent Large, 3 - Occasional Large, 4 - Frequent Medium, 5 - Loyal Small, 6 - Occasional Medium, 7 - Occasional Small and 8 - Infrequent. .

- The Lifestage Segmentation (*Slifestage*) combines socio-demographic information to infer about the customer life stage. This segmentation considers the age of the customers and whether they have dependents or not in their aggregate. There are five segments: 1 - Active Adults, 2 - Senior, 3 - Family with kids, 4 - Family with young adults and 5 - Family Supporters.

Studies [47] indicate that Promoters are more loyal and more likely to purchase products than Detractors. Therefore, it is essential to collect data on the respondents' purchases to understand their habits and economic value as customers. The Purchase Behaviour variables (Table 4.2) combine transactional and product data. These variables help us understand how and what the customer buys, which from previous business knowledge, we know these tend to be the core variables when predicting the NPS.

Besides information about the customer and its purchasing habits it is imperative to have information about the company under analysis, Note's stores. Accordingly, Store Profile (Table 4.2) integrates variables related to the customer preferential Note store, i.e., the store the customer visits the most. It is to be noted that the variables in *Store Traffic* and in *Transactions Distribution* are measured through five periods of the day. These periods are divided as follows: (1) morning, (2) lunch, (3) early afternoon, (4) afternoon, and (5) evening. So in the data set, each of these variables is represented five times, for instance, *Store Traffic 1* represents the store traffic on the customer's preferential store in the morning period.

Table 4.2: Profile's description.

	Name	Type	Description
Customer Profile	*Age*	Continuous	Age
	Account Longevity	Continuous	The number of years the customer has had the account
	GenderF	Categorical	Inferred female behaviour
	GenderM	Categorical	Inferred male behaviour
	Segmentations	Categorical	Customer's segmentations
Purchase Behaviour	Discounts	Continuous	Direct and Deferred discount benefited
	Gross Amount	Continuous	The total gross amount purchased and returned
	Basket	Continuous	The gross amount spent/returned
	Quantity	Discrete	Number of items bought
	SKU	Discrete	Number of SKUs (SKU=product identifier)
	Price	Continuous	Price of the product
	Subcategory	Discrete	Number of subcategories of the product
	Category	Discrete	Number of categories of the product
	BU	Discrete	Number of business units of the product
	Transactions	Discrete	Number of transactions
	App Transactions	Discrete	Number and percentage of app transactions
	Recency	Discrete	Number of days since the last transaction
	Mission	Continuous	The percentage of transactions in each mission
Store Profile	*DOP*	Categorical	Operacional Direction
	Concept	Categorical	Store antiquity concept
	Store Type	Categorical	Type of store based on its area and the range of products it sells
	Type of Location	Categorical	Store location
	Store Traffic	Continuous	The percentage of store traffic during the week and weekends for each period of the day
	Transactions Distribution	Discrete	The percentage of transactions for each period of the day during the week and at weekends

4.1.2 Exploratory Analysis on the Data

To understand how the data behaves it is important to submit it to an exploratory analysis, calculating descriptive statistics and interpreting them. The analysis in this chapter is done with the previously collected data (Table 4.2), which is composed by 13175 observations, within 60 variables where eleven of them are qualitative (categorical), and the remaining quantitative (discrete and continuous). These variables can be consulted in Appendix A.

Categorical Variables

The database has eleven categorical variables, where one is for identification purposes (*id_cliente*) and does not go into the model, and the remaining ten are composed by different levels. The distribution of these variables according to their levels and to the three *NPS* classes (*Detractors*, *Passives* and *Promoters*) are represented in Figures 4.3 and 4.4.

Analyzing the barplots in Figures 4.3 and 4.4, we can see how the observations in each variable are distributed, and it is possible to see that most of the observations belong to the *Promoters*, followed by the *Passives*, whereas a minority belong to the *Detractors*.

Furthermore, in Figures 4.5 and 4.6 , we can see how each level of the different variables is distributed according to the *NPS*. Across all variables, the distribution of the *NPS* seems equitable for the different levels. That said, the *NPS* does not seem to be influenced by the different levels of the variables.

Quantitative Variables

Regarding the remaining 48 variables in the database, they are quantitative. Table 4.3 shows statistics for some of the most relevant results of the quantitative variables. The presented values were multiplied by a random factor due to data privacy issues. Looking at the table we can note the high values of standard deviation (at the same level as the mean). This suggests that there are very distinct patterns of behaviour in each class for each variable.

When looking at differences between classes we can see that there are some differences in the minimum values of *Account Longevity*, where the *Detractors* appear to be customers for longer than the customers from the remaining classes. In *Gross Amount* (Max), the *Promoters* are the customers who spend more, and in consequence have a bigger *Basket*. By looking at the maximum

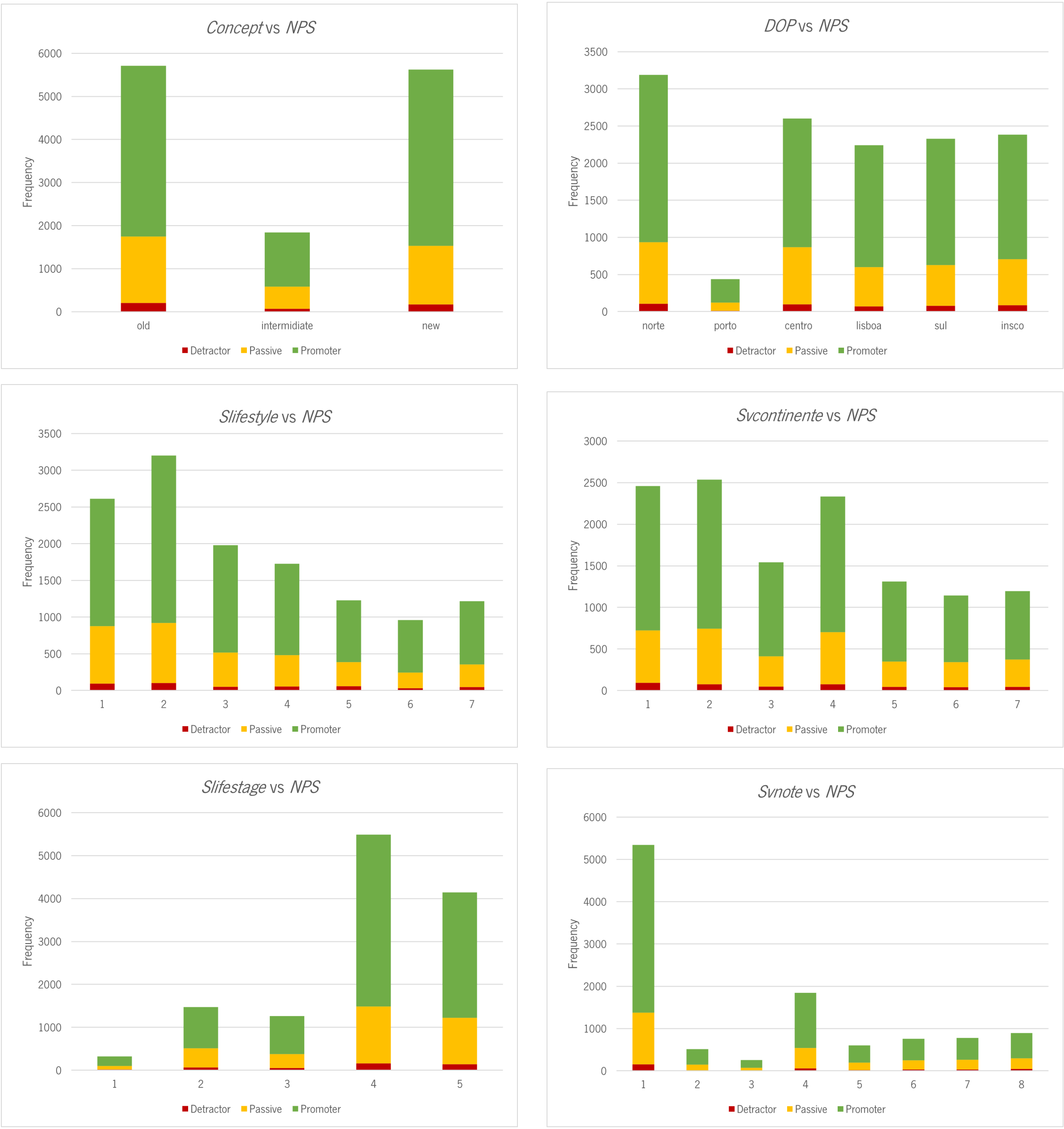


Figure 4.3: The distribution of each categorical variable per *NPS* - Part I.

and standard deviation values of the variables *Quantity*, we can see that there is a big difference (intra and inter class) in the number of items the *Detractors* buy. Even though the *Detractors* shop the highest quantities, the *Passives* have a more diversified shopping basket, with more products *SKU* (Max), from more different categories (*Categories* (Max)). The maximum values in *Transactions* differentiate the *Passives* from the other classes, albeit the *Promoters* have more app transactions (*App Transactions* (Max)). Lastly, the *Detractors* have a lower recency, meaning they visited the store more recently prior to responding to the survey than the rest. These results reveal some key

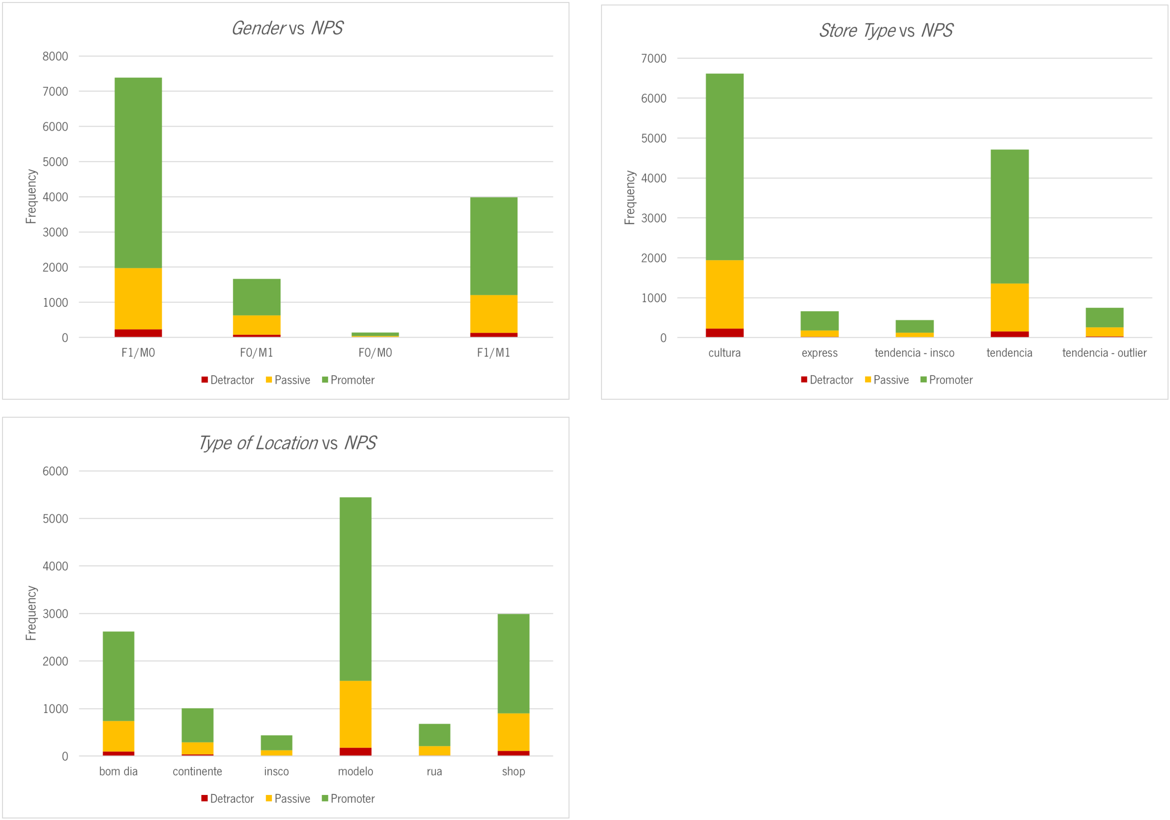


Figure 4.4: The distribution of each categorical variable per *NPS* - Part II.

inputs that work as baselines to the succeeding section 4.1.3 on Variable Construction. By having differentiated values among classes, it is easier for the model to distinguish each class and therefore make a better prediction.

As an explanatory analysis, it is paramount to verify if the data follows a Normal distribution. In Figures 4.7 and 4.8 are presented the histograms with the respective density curve for each core variable. Only the variable *Age* seems to be normally distributed. Due to the sample size of the data set, the normality of the data was rejected for every variable with different tests at different significance levels (see Appendix B.1).

Another topic to be analysed is the correlation between the variables to assess if predictors are independent from each other. Given the context of the quantitative data it is expectable to obtain high correlation between most of the variables, since they are obtained through each other. Figure 4.9 represents the correlation matrix for the core quantitative variables with the respective Spearman correlation coefficients. As it can be observed there are several variables whose correlation is extremely high, namely between *gastos_sales_total* and *qty_sales_total* (r_s = 0.90);

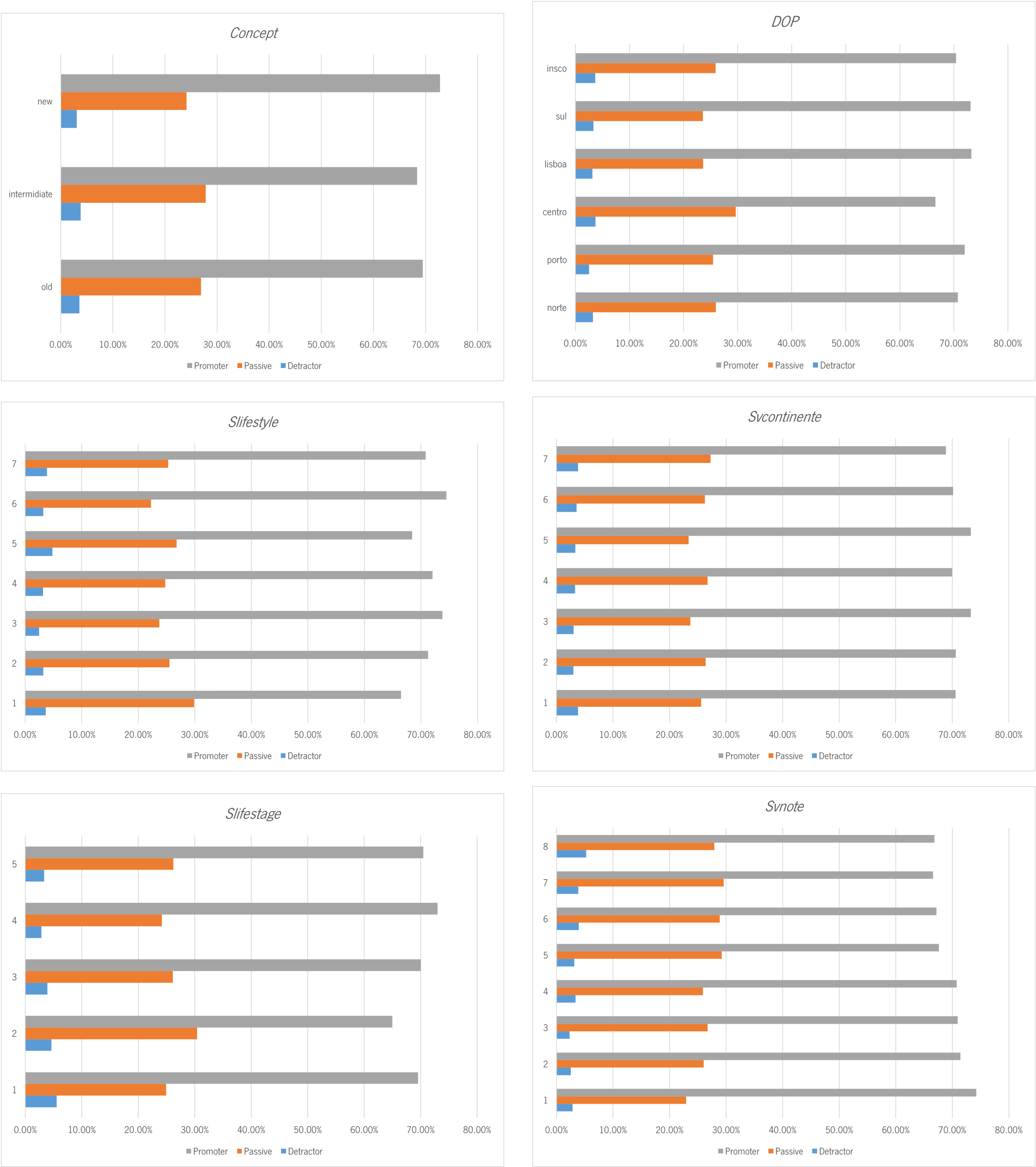


Figure 4.5: The distribution of each level of the different variables according to the *NPS* - Part II.

gastos_sales_total and *n_trx_total* (r_s = 0.88); and *qty_sales_total* and *n_trx_total* (r_s = 0.88). It is easy to understand since a customer who buys more quantity tends to spend more. In the same way, the more transactions the customer makes, the more it tends to spend, or the more transactions, the more quantity it tends to buy.

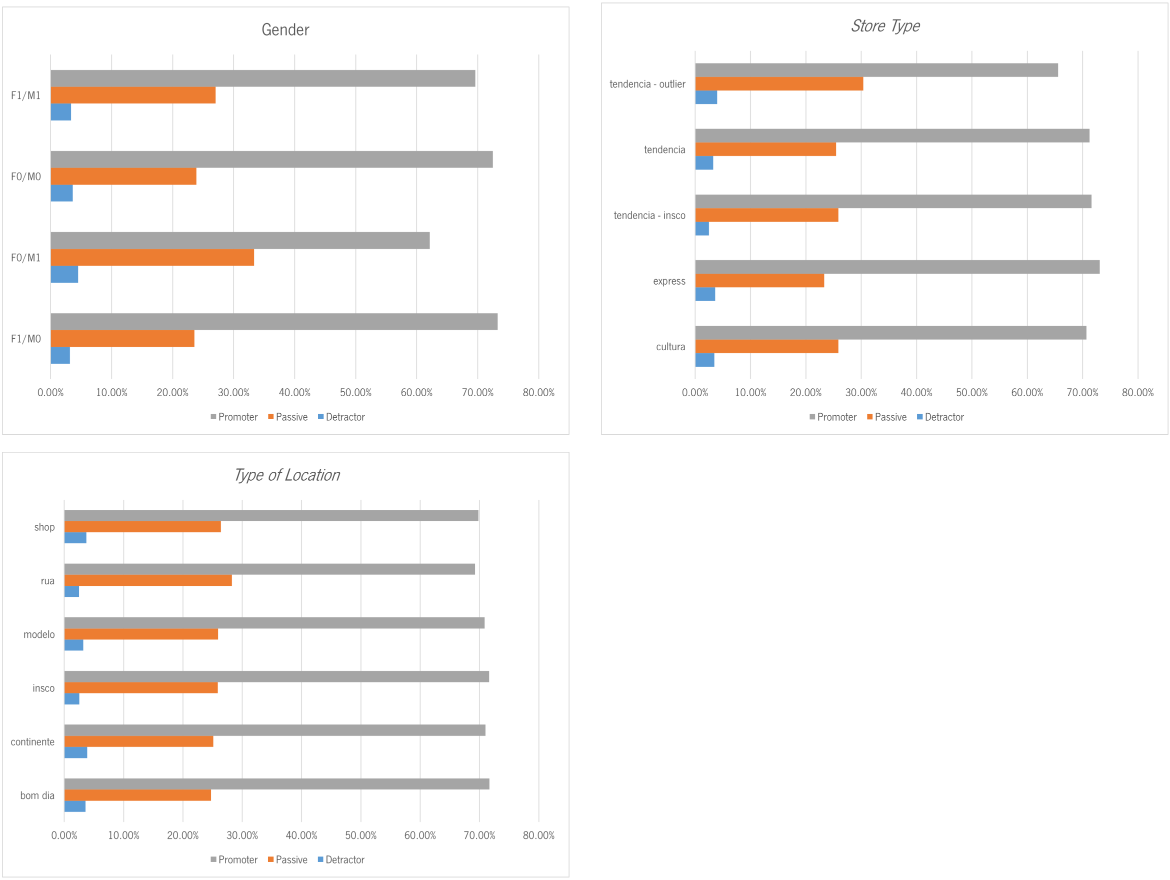


Figure 4.6: The distribution of each categorical variable per *NPS* - Part II.

Table 4.3: Summary statistics for quantitative variables.

		Min	Median	Mean	Max	SD	CV
Age	D	18.00	45.00	47.15	78.00	12.48	0.26
	N	18.00	44.00	45.76	79.00	12.48	0.27
	P	18.00	43.00	44.40	79.00	12.39	0.28
Account Longetivity	D	251.00	4329.00	3772.00	5466.00	1415.82	0.38
	N	3.00	4538.00	3815.00	5467.00	1455.28	0.38
	P	17.00	4406.00	3730.00	5467.00	1520.24	0.41
Gross Amount	D	0.00	54.85	174.88	3469.93	352.19	2.01
	N	0.00	63.06	176.58	6264.20	376.83	2.13
	P	0.00	87.38	227.98	13526.45	499.15	2.19
Basket	D	0.00	16.57	24.85	317.00	32.31	1.30
	N	0.00	18.07	25.68	353.78	29.49	1.15
	P	0.00	20.34	28.63	945.72	29.49	1.03
Quantity	D	0.00	12.00	181.10	37181.00	1806.06	9.97
	N	0.00	15.00	70.47	7146.00	278.46	3.95
	P	0.00	21.00	93.00	25534.00	435.07	4.68
SKU	D	0.00	5.00	16.59	689.00	50.53	3.05
	N	0.00	6.00	14.64	896.00	30.03	2.05
	P	0.00	8.00	18.73	750.00	35.17	1.88
Categories	D	0.00	3.00	9.92	346.00	30.84	3.11
	N	0.00	4.00	8.54	855.00	23.31	2.73
	P	0.00	5.00	10.65	391.00	23.76	2.23
Transactions	D	0.00	2.00	8.52	336.00	29.57	3.47
	N	0.00	3.00	6.87	853.00	21.81	3.17
	P	0.00	4.00	8.43	368.00	21.45	2.54
App Transactions	D	0.00	0.00	0.63	32.00	2.74	4.36
	N	0.00	0.00	0.64	31.00	2.04	3.21
	P	0.00	0.00	1.02	189.00	4.35	4.27
Recency	D	0.00	9.00	8.65	37.00	5.31	0.61
	N	0.00	8.00	9.25	772.00	17.93	1.94
	P	0.00	8.00	8.34	552.00	7.97	0.96

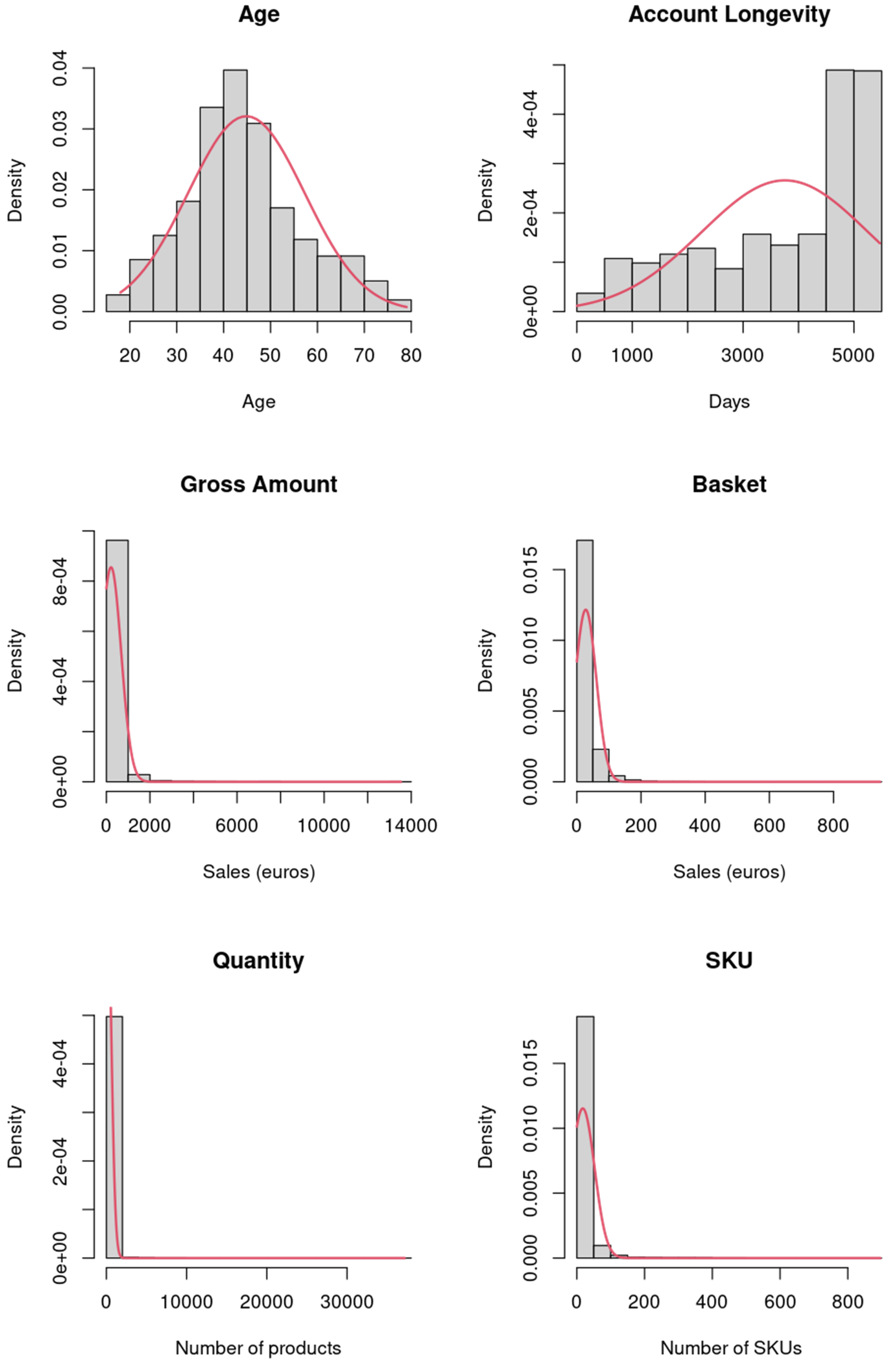


Figure 4.7: Histograms with density curve for the core variables - Part I.

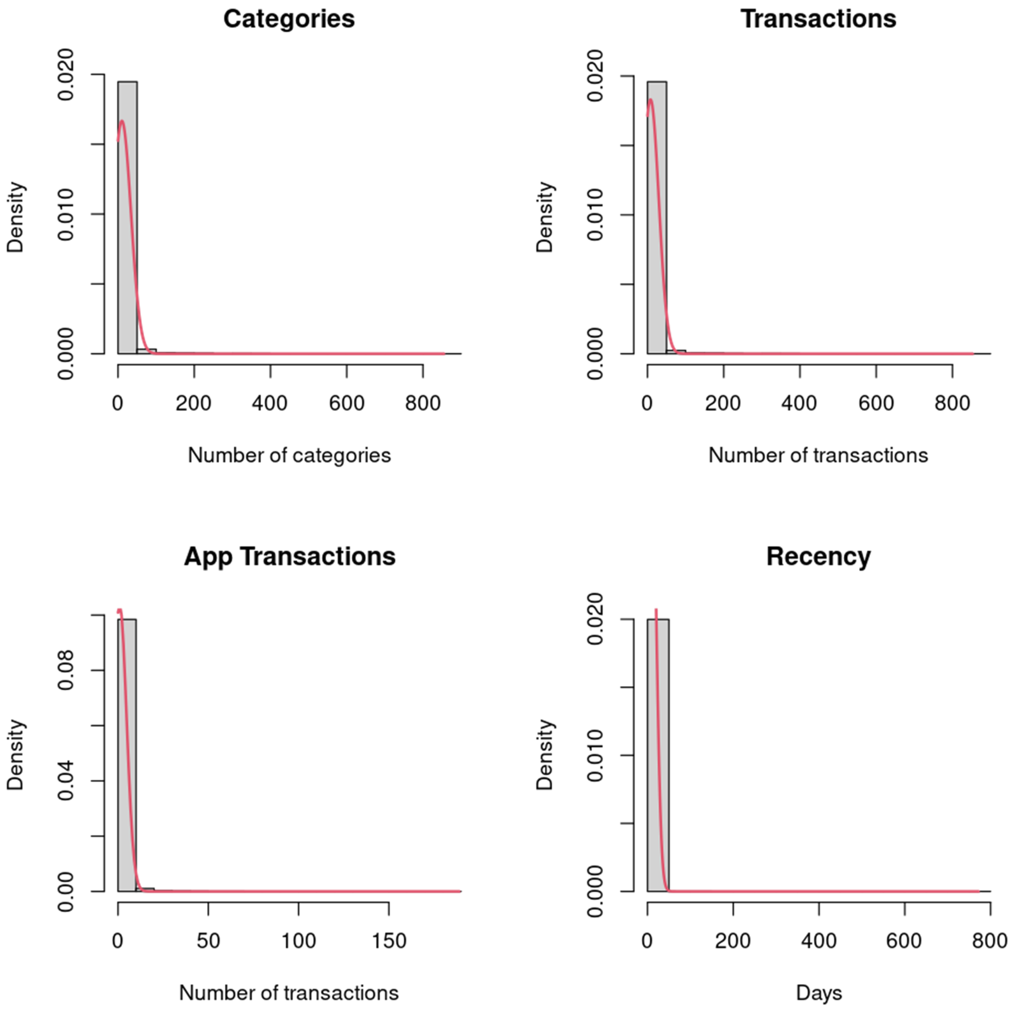
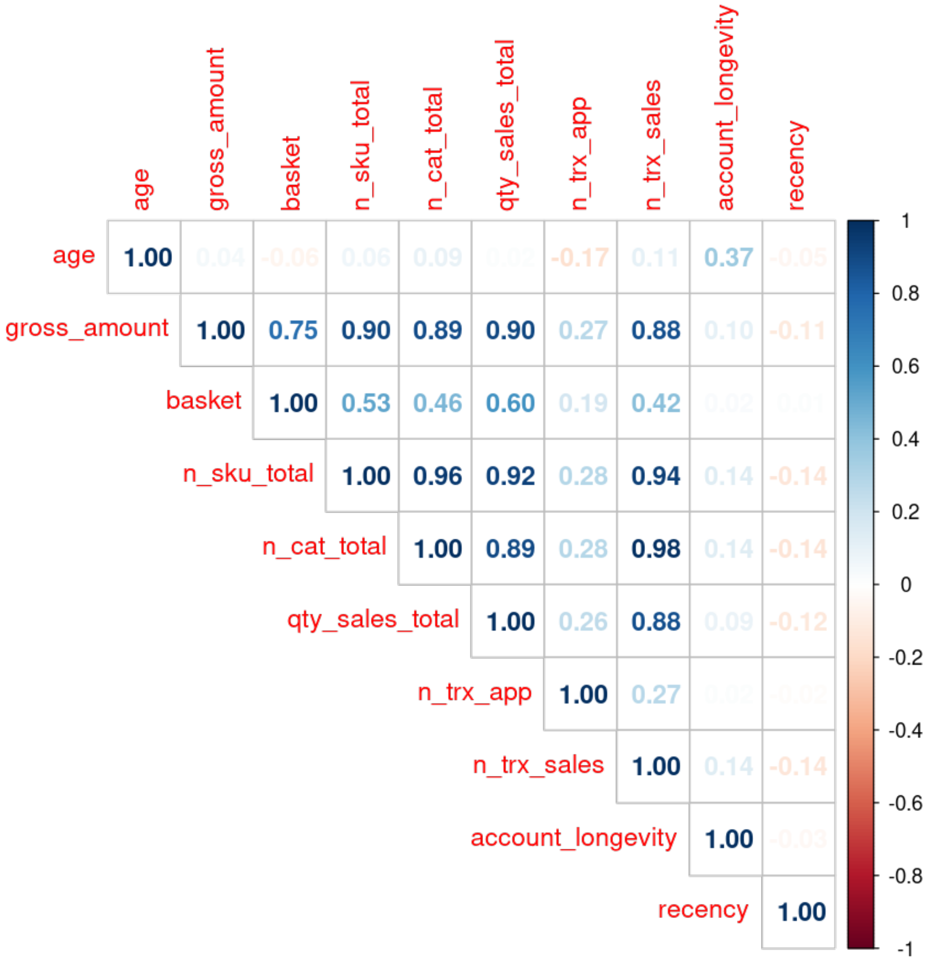


Figure 4.8: Histograms with density curve for the core variables - Part II.



	age	gross_amount	basket	n_sku_total	n_cat_total	qty_sales_total	n_trx_app	n_trx_sales	account_longevity	recency
age	1.00	0.04	-0.06	0.06	0.09	0.02	-0.17	0.11	0.37	-0.05
gross_amount		1.00	0.75	0.90	0.89	0.90	0.27	0.88	0.10	-0.11
basket			1.00	0.53	0.46	0.60	0.19	0.42	0.02	0.01
n_sku_total				1.00	0.96	0.92	0.28	0.94	0.14	-0.14
n_cat_total					1.00	0.89	0.28	0.98	0.14	-0.14
qty_sales_total						1.00	0.26	0.88	0.09	-0.12
n_trx_app							1.00	0.27	0.02	-0.02
n_trx_sales								1.00	0.14	-0.14
account_longevity									1.00	-0.03
recency										1.00

Figure 4.9: Spearman correlation matrix for the core variables.

To have a better understanding of the influence of the different periods of the day on the data, we studied the variables in *Store Traffic* and *Transactions Distribution*, which hold the distribution of the customer's preferential store traffic and the distribution of the customer's transactions for each period of the day, during the week and on weekends, respectively. For reference, the periods are divided as follows: (1) morning, (2) lunch, (3) early afternoon, (4) afternoon, and (5) evening.

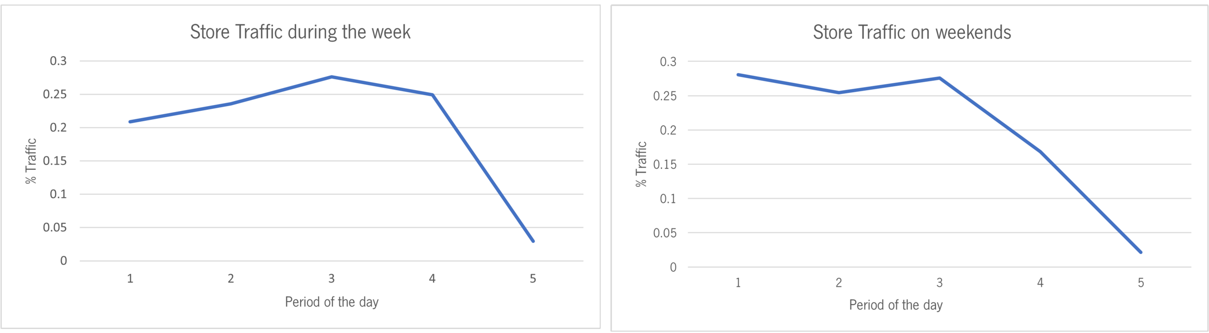


Figure 4.10: Distribution of store traffic for each period of the day, during the week (left) and on weekends (right).

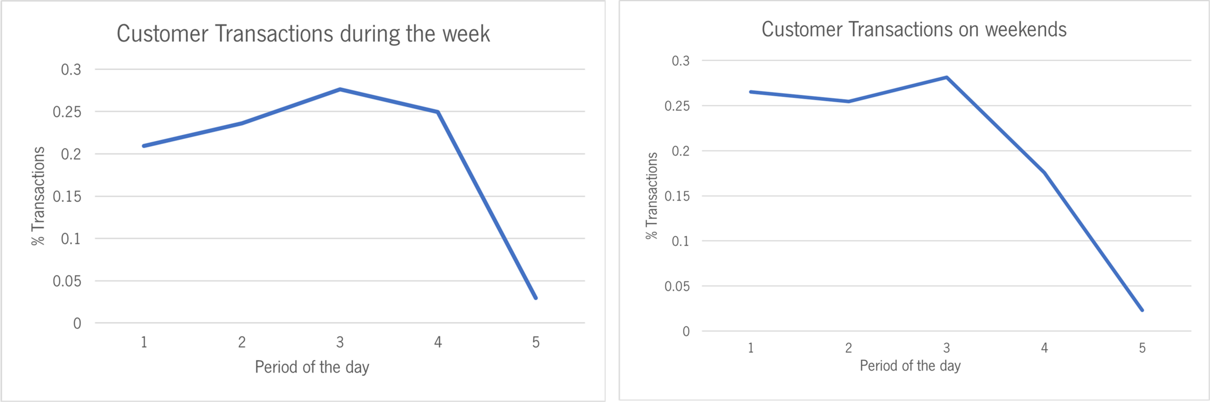


Figure 4.11: Distribution of customers transactions for each period of the day, during the week (left) and on weekends (right).

These two groups of variables, *Store Traffic* and *Transactions Distribution*, are deeply interrelated since the store traffic at a given time is a consequence of the number of customers shopping at that specific time.

In Figure 4.10 (left), we can notice that there are two peaks in the chart, one more prominent on the third period of the day, early afternoon, and another one on the fourth period, in the afternoon, meaning these are the periods with more traffic on the stores during the week. These go in conformity with Figure 4.11 (left), which tells us that the customers tend to buy more in those periods. The same can be observed on weekends. Figure 4.10 (right) tells us the stores have more traffic in the first and third periods of the day, which matches the periods when customers make more transactions (Figure 4.10 (right)). We can conclude that the great majority of transactions during the week are made after lunch and prior to the evening and on weekends during the morning until the afternoon. As a consequence these are also the periods with more store traffic.

4.1.3 Data Preparation and Transformation

The exploratory analysis presented in the previous chapter gives a first insight regarding the relevance and behaviour of each variable when predicting the NPS. In this chapter we show how to prepare and transform the data set to include in the machine learning algorithms.

Handling missing values

The first step in data preparation is to treat missing values. In the process of creating the data set for the study, the removal of inconsistent data and the treatment of missing data was performed in parallel.

In the survey data, the percentage of missing values across study variables were 0.91% for *DOP* and 1.16% for the variable *Concept*. These missing values were handled by imputing information from Cartão Continente databases. This process started with creating a new variable with the customer's preferential store (*fave_store*). This new variable replaces the store location (*cod_loja*) presented in the survey, which only gives information about the last store the client visited and might not match the preferential one. This way, we can get a better history of the customer store preference. Afterwards, the missing values of the variables *DOP* and *Concept* were handled by matching the customer preferential store with the designated *DOP* and *Concept* imputed from Cartão Continente databases. In cases where there was no information about the variables, due to the store closing down or out-of-date location codes, the observations were removed (1.01% of the observations). The code in R in Appendices C.1 and C.2 exemplify the execution of creating the variable *fave_store* and imputing missing values of the variable *DOP*, respectively. The process for the variable *concept* was similar.

In Loyalty card data, there were several variables with missing values. This was already expectable since Note's stores have a low penetration rate, meaning that many customers shop at Note but do not use their loyalty cards. As a consequence, there is no available information about the customer, their habits and transactions. As we know that the customers who were surveyed visited a Note store, the situations where there are missing values are equally important to be analysed. In the case of categorical variables, the missing values were categorized into a new segmentation - No Value. In the case of numerical variables the existence of missing values meant, for example, that the customer had no purchases in that period and these cases were replaced by zero.

Variable Construction

In order to get more significant variables to train the models with, we engineered new variables from already existing ones. This process is denominated Variable Construction. By adding derived variables, beside enriching the data, this process can potentially capture relevant and explicit relationships that would otherwise be implicit and difficult to comprehend.

The first engineered variables were created to shorten the periods from which we were extracting the data. That is, we seek to analyze the Purchase Behaviour variables (4.2) considering periods close to the customer's last transaction. With that, all variables were extracted for the Last 6 Months (L6M), the Last 3 Months (L3M) and the Last Month (LM) before the survey. So, each variable will be represented four times, one for each time period. For instance, each observation has the number of transactions of the previous 12 months (N Transactions L12M), the previous 6 months (N Transactions L6M), the previous 3 months (N Transactions L3M) and the previous month (N Transactions LM) of the survey.

Moreover, to create more variability between the data, the mean, standard deviation, maximum and minimum of some of the Purchase Behaviour variables for the four time periods were also added as variables. In particular, for the variables *Basket*, *Quantity*, *SKU*, *Price*, *Categories* and *Subcategories* (see Appendix C.3). Table 4.4 illustrates an arbitrary example of some of these variables for a single observation (a row in the data set), which for an easier understanding, is displayed in a table divided by time period. For instance, for one observation, the variable *Quantity Mean* for the previous year (L12M) will be the number of products bought (*Quantity*) divided by the number of transactions of the previous year. The final data set can be consulted in Appendix A.

Table 4.4: An illustrative example of some Purchase Behaviour variables for one observation of the data set in different time periods.

Time	Quantity	N SKU	N Transactions	Quantity Mean =	Quantity	Quantity	N SKU Mean =	...
Period				Quantity/ N Transactions	Max	Min	N SKU/ N Transactions	
L12M	18	6	3	6	12	3	2	...
L6M	12	4	1	12	12	12	4	...
L3M	0	0	0	0	0	0	0	...
LM	0	0	0	0	0	0	0	...

One-Hot Encoding for categorical variables

Although the CART decision tree algorithm handles both categorical and numerical variables, the partitioning algorithm tends to favour categorical predictors with many levels k. That is, the number of partitions grows exponentially in k, and the more choices we have, the more likely we can find a good one for the data at hand. That can lead to overfitting, and such variables should be avoided [28]. Thus, a complete search of the possible splits is only practical for categorical variables with few categories. Additionally, in [34] the authors found that the accuracy of the results of the random forest algorithm was improved by the use of the One-Hot encoding method. Therefore, as a matter of efficiency, the categorical variables were transformed into several variables through One-Hot encoding. An example of this transformation can be seen on Table 4.5.

Table 4.5: Example of one-hot-encoding: variable *DOP* is transformed into 6 dummy variables

Observation	*DOP*		Observation	*DOP_norte*	*DOP_porto*	*DOP_centro*	*DOP_lisboa*	*DOP_sul*	*DOP_insco*
1	porto		1	0	1	0	0	0	0
2	lisboa	⇒	2	0	0	0	1	0	0
3	lisboa		3	0	0	0	1	0	0
4	centro		4	0	0	1	0	0	0

Training and Testing data set

Training and test data sets are two different but important parts in machine learning. The training set is used for the algorithm to learn how to classify the customers. While, the testing set is kept isolated, to be used just at the end of the modelling procedure to estimate the performance of the chosen model, removing the overfitting effect, i.e. the increase in performance that the algorithm has on the observations it has based its learning on. There are different methods for splitting the dataset. We chose to use a common one, where 70% of the data set was randomly chosen for training (n=9222), and the remaining 30% for testing (n=3953).

Balancing data set

Considering the *Detractors* the minority class (3.40%) and the *Promoters* the majority class (70.75%), with an imbalanced ratio of 20.82, the data set is extremely imbalanced Table 4.6 presents an overview of the imbalanced data set. The presented values were multiplied by a random factor due to data privacy issues.

Table 4.6: Imbalanced data set overview.

	Observations	Variables	Classes	Majority Class (Maj)	Minority Class (Min)	Imbalance ratio ($\mathbb{R}$ = N$_{Maj}/N_{Min})$
Data set	13175	371	{D,N,P}	P (9321)	D (448)	20.81

Directly training a ML model with an imbalanced data set can result in good performance for the majority class but poor performance for the minority class [61]. To handle the imbalance problem, we decided to apply several well known resampling approaches to the training set at each iteration, such as Random Oversampling (ROS), Random Undersampling (RUS) and SMOTE, and compare their results. These methods were applied as a way to achieve a balanced data set. This decision was motivated by the results in [74], in which it is shown that when AUC is used as performance measure, the best class distribution for learning tends to be near a balanced class distribution. Table 4.7 describes the distribution of the training set per class after each technique was applied.

Table 4.7: Distribution per class of the training set after each technique.

Techniques	D	N	P	Total
Original	309	2379	6534	9222
ROS	6534	6534	6534	19602
RUS	309	309	309	927
SMOTE	3074	3073	3074	9221

These approaches were performed with the *UBL* package of R. The libraries used were *RandOverClassif* for ROS, *RandUnderClassif* for RUS and *SmoteClassif* for SMOTE.

Chapter 5

Results and Discussion

The following chapter describes the application of the proposed methodology to predict and explain a Note's customer NPS. Followed by a discussion of the business implications of the results, and lastly, an error analysis thereof.

As mentioned earlier on, the development of this project followed the steps of CRISP-DM. This framework provides an iterative approach, including frequent opportunities to evaluate the progress of the project against its original objectives. As so, when constructing the model, a Minimum Viable Product (MVP) approach was used. The MVP settles on the premise that you start with delivering minimal variables and then collect feedback that will enable you to build a better model that will resonate with future users. That being said, the final data set presented in Section 4.1.1 was built in an iterative way, meaning that multiple data sets were tested and run through the modelling phase before settling with the final one. Hence, the results will also be presented in an iterative way.

5.1 Experimental Proceedings

The aim of predicting a Note's customer NPS can be approached through supervised learning. The idea is to create a model capable of classifying a certain consumer as being a *Promoter*, *Passive* or *Detractor*, based on multiple variables that characterize the consumers, so it can:

1. Identify the most relevant variables that characterize each class,
2. Give insights on a possible strategy to understand which effects impact the customer's classification.

The first step in the classification problem is to select a supervised learning method. As previously mentioned, multiple methods have been applied in the literature to predict the NPS such as Decision Trees, Random Forests, Linear Regression, Support Vector Machines and Naive Bayes, among others (see [47] and [70]).

In a first phase, a baseline DT model with the CART algorithm was implemented due to its interpretability, robustness to outliers and capacity to handle both categorical and numerical values. The DT model was implemented in the language R with the package *rpart* [68].

The first trials were carried out in a smaller data set where the Random Oversampling (ROS), Random Undersampling (RUS) and Synthetic Minority Oversampling SMOTE methods described in Section 3.1.1 were applied to achieve a balanced data set. The performance of the models was quantified by using the performance measures described in Section 3.4. To compare results, the AUC was chosen as the main evaluation metric since it is considered a good measure of model performance for classification problems [20].

As mentioned in section 3.2.1, there is still a lot of debate on whether pruning trees grown over artificially balanced datasets can improve the classifier's performance. As an experimental procedure, the evaluation metrics were measured over pruned and unpruned decision trees for the ROS and RUS methods. The complexity parameter of a DT model has a crucial effect on its performance and it is partially controlled by the pruning method employed. Following the work of [60] and [67], we focused on optimizing this parameter by finding the CP value with the smallest prediction error in cross-validation. For the unpruned DT we worked with the default values of the parameters. Different distance metrics were also tried out for the SMOTE method, in particular the HEOM and the HVDM. The results reported a model with low AUC and accuracy, which indicates a poor predictive performance (Table 5.1). Looking at the other evaluation metrics, we can see that there is a high number of both false negatives and false positives due to the low scores of macroS and macroP, respectively, and, consequently, an overall low score of the macroF1 metric. The best test result was the ROS method with an AUC of only 0.52. On another note, the results clearly show that pruning leads to no improvement in AUC for the balanced data sets.

A common way to improve a model's performance is to simply add more data, but in this case that was not possible since we did not have more observations. Nevertheless, several studies (see [39], [59] and [5]) suggest that adding more variables can enhance the predictive performance of a model. Moreover, in [77] the authors conclude that constructing variables from already existing

Table 5.1: Comparison of the classification metrics for Decision Tree model (initial data set).

Model	AUC	BRIER	ACC	macroS	macroP	macroF1
Original	0.50	0.19	0.72	0.33	0.24	0.28
ROS	0.52	0.30	0.47	0.36	0.35	0.31
ROS (prune)	0.52	0.30	0.47	0.36	0.35	0.31
RUS	0.52	0.32	0.29	0.33	0.35	0.25
RUS (prune)	0.51	0.31	0.60	0.33	0.25	0.27
SMOTE (HEOM)	0.50	0.28	0.47	0.33	0.33	0.31
SMOTE (HVDM)	0.50	0.28	0.57	0.33	0.25	0.27

ones (Variable Construction) can significantly improve classifier performance when compared to classification learning alone .

When analysing the variable importance of the decision trees, the variables related to Purchase Behaviour (see Table 4.2) were deemed more important, and therefore, they were used for Variable Construction. In a first moment, besides all the variables collected for the L12M, the same variables were also extracted for the L6M, L3M and LM. The results showed no big improvements. In a second phase, the variables regarding the mean, standard deviation, maximum and minimum of some variables were added, in particular for the variables regarding the *Basket*, *Quantity*, *SKU*, *Price*, *Subcategory*, *Category* and *BU*. These were thought to include more variability in the data and thus help to improve the model's performance.

The results obtained for the final data set (see Appendix A) are summarized in Table 5.2 which continued to show a poor predictive performance (low AUC values), with the SMOTE (HVDM) method having the highest AUC of 0.53. In general, the AUC values suggest that the model cannot discriminate the three NPS classes.

One obvious conclusion drawn from the results is that the model is underfitting, that is, the model was not capable of learning the patterns in the training data well and is unable to generalize on new data. This is corroborated by the poor performance on the training data, which results in unreliable predictions.

Figure 5.1 shows the classification tree for customer's NPS based on the training subsample where it was applied the SMOTE method with the HVDM distance. The first line of each internal

Table 5.2: Comparison of the classification metrics for Decision Tree models (final data set).

Model	AUC	BRIER	ACC	macroS	macroP	macroF1
Original	0.50	0.20	0.70	0.33	0.24	0.28
ROS	0.52	0.30	0.40	0.38	0.35	0.28
RUS	0.51	0.31	0.37	0.36	0.34	0.29
SMOTE (HEOM)	0.52	0.27	0.51	0.34	0.34	0.33
SMOTE (HVDM)	0.53	0.26	0.57	0.37	0.36	0.35

node refers which class is more prevalent in that node. The second line indicates the predicted probability of each class. The third line indicates the percentage of observations involved in the node. And finally, below each internal node is the decision rule with a selected variable. For a node with branches, its left child node follows the decision rule in the parent node, whereas its right child follows the complement of the decision rule. The colour's shade refers to the class label, where the darker the colour, the higher the probability of the predicted class. For example, at the top, the root node starts with 100% of the training observations and shows the proportion of observations in each class. As the SMOTE technique was applied and there is a balanced training set, we start with a proportion of $\frac{1}{3}$ for each class. Its decision rule splits the observations whose variable *slifestage_4* is smaller than 1. If the answer is 'yes' then there is a 40% chance the observation is a *Detractor*. If 'no', then there is a 45% chance it is a *Promoter*. The same reasoning applies to the rest of the tree until we arrive at the leaf nodes (bottom of the tree), where it is possible to observe what variables impact the likelihood of belonging to one of the three NPS classes.

Looking at the leaf nodes of the decision tree in Figure 5.1, we can see that the model is predicting the *Detractors* with a much higher probability than the rest of the classes. Whereas when it predicts as a *Passive* or *Promoter*, there is not as much discrepancy between the probabilities, which means this particular decision tree model is having difficulties in discriminating these classes.

There are other model-driven methods to increase the model's performance such as hyper-parameter tuning and ensemble methods. The first method consists of determining the right combination of hyper-parameters that maximizes the model's performance. Whereas the parameters are estimated by the model from a given data set, the hyper-parameters are responsible for estimating the model's parameters. Yet, the authors in [46] concluded that the trees that did not obtain a

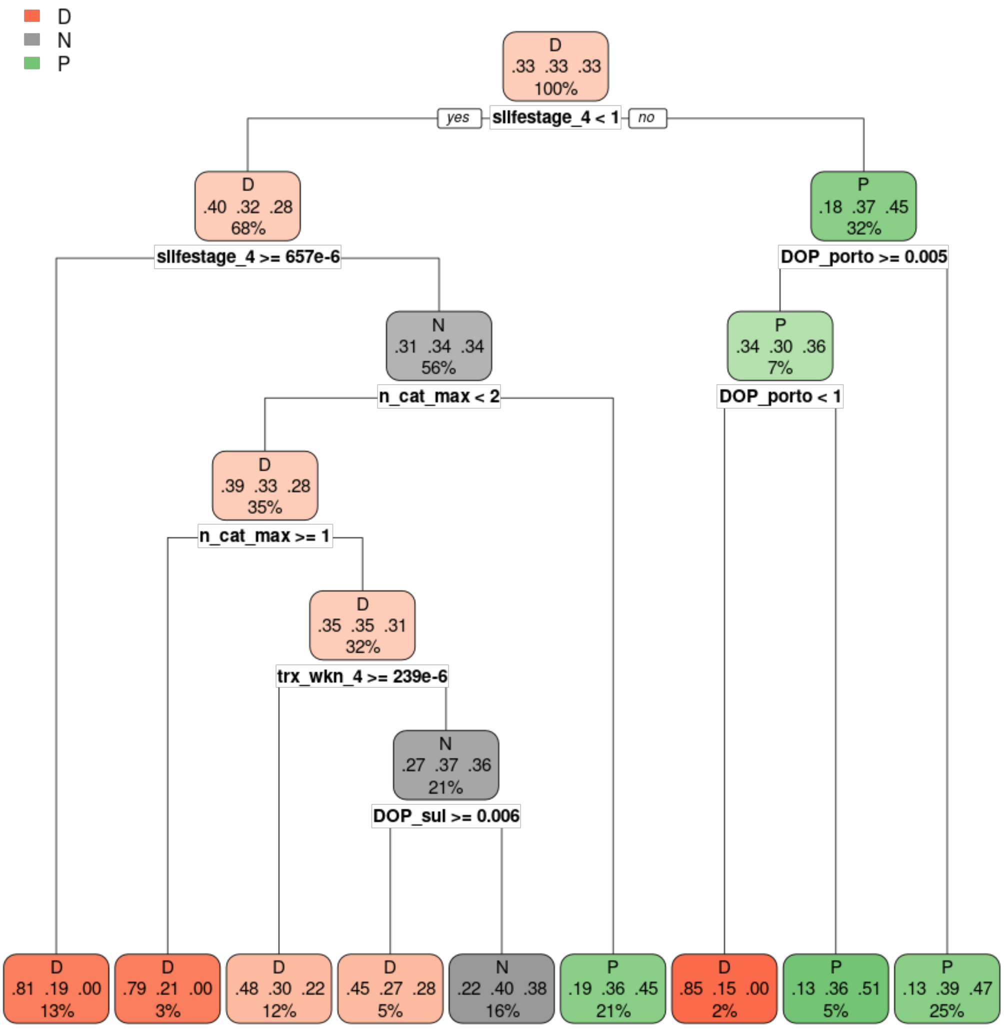


Figure 5.1: Decision Tree for classifying customer's NPS - Model with SMOTE(HDVM). Note: slifestyle_4 = Lifestyle Segmentation (level 4); DOP_porto and DOP_sul = DOP (levels = porto, sul); n_cat_max = Maximum number of categories bought in the last 12 months until the survey; trx_wkn_4 = percentage of transactions that the customer makes on its preferential store during the fourth period of the day (afternoon).

good predictive performance, when using the default values to grow the trees, the performance was similar to the optimized performance. The second method consists on applying ensemble learning. These algorithms, instead of trying to learn one super-accurate model, focus on training a large number of low-accuracy models and then combining the predictions to obtain a better model. One of the most widely used and effective ensemble learning algorithm is random forest.

On that account, in a second phase, and due to the bad performance of the DT algorithm, a RF algorithm was applied to the final data set. This choice is supported by recent studies (see [44] and [78]) that demonstrate an improvement in classification performance when using RF algorithms in comparison to DT. The RF model is also an easily interpreted model and robust to outliers, taking on the advantage of also being robust to overfitting. This model was implemented in the language R with the package *randomForest* [40].

The RF algorithm involves several hyper-parameters such as the number of variables considered as candidate splitting variables at each split ($mtry$), the minimal size a node should have to be split ($nodesize$), as well as the number of trees ($ntree$). Among those the $nodesize$ has the minor influence on the model's performance, and the default value was used. For the $mtry$ we also used the default value, which is set to $\sqrt{M}$, with M being the number of predictor variables. This decision was motivated by [7], where the authors concluded that this value is a reasonable setting to induct near optimal RF performance. As for the hyper-parameter $ntree$, the higher the number of trees, the better the results in terms of performance and precision of variable importances. However, the improvement obtained by adding trees diminishes as more and more trees are added [52]. When evaluating tree-based models such as RF, OOB error is selected as the unbiased estimation of the error. To determine the number of decision trees ($ntree$) included in the model the OOB error rate is plotted against the numbers of trees included. Based on Figure 5.2, we can observe that the OOB error rate and the error rate for the class *Detractors* is not sensitive to the number of trees included in the model, in contrast to the error rate for the classes *Passives* and *Promoters*. We chose to build the model with 400 trees, since the OOB error rate for the classes *Passives* and *Promoters* are relatively stable and less sensitive around that tree number, and no additional benefit would be gained by adding more trees.

The model's evaluation is shown in Table 5.3. There was a minimal improvement in the AUC value of the SMOTE(HVDM) model, and despite the RF algorithm presenting a better performance than the DT in most metrics, the results still indicate a low-performance model.

5.2 Discussion of Results

In tree-based models, such as Random Forest, we can analyse the importance of each variables in the prediction, which is measured by the mean decrease in the Gini index. As the Gini index is a

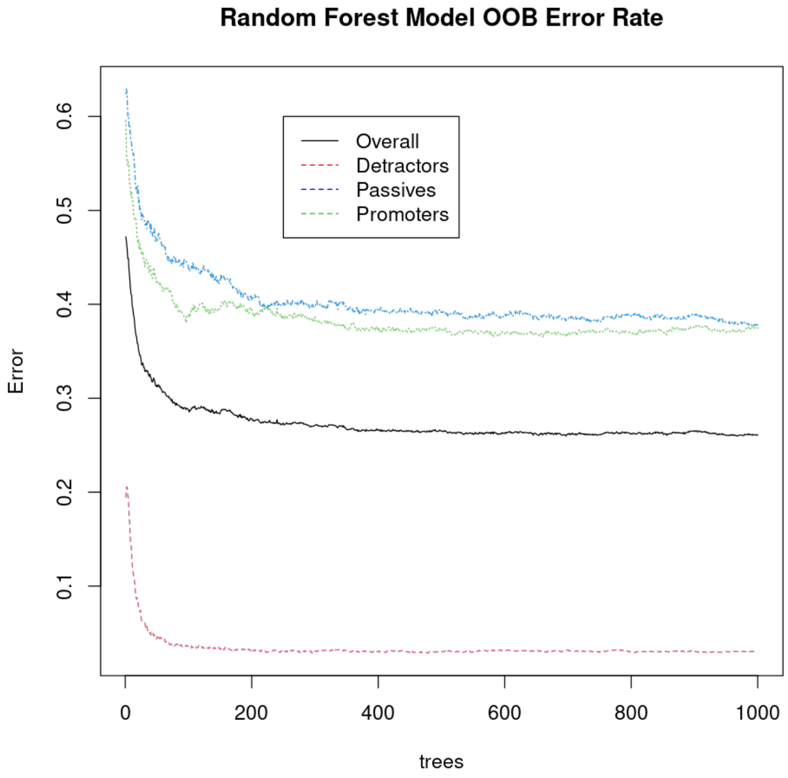


Figure 5.2: OOB error rate vs. number of trees.

Table 5.3: Comparison of the classification metrics for Random Forest model.

Model	AUC	BRIER	ACC	macroS	macroP	macroF1
Original	0.52	0.21	0.71	0.34	0.49	0.28
ROS	0.54	0.24	0.62	0.35	0.49	0.34
RUS	0.53	0.31	0.36	0.37	0.35	0.29
SMOTE (HEOM)	0.55	0.26	0.56	0.36	0.35	0.35
SMOTE (HVDM)	0.55	0.26	0.56	0.36	0.36	0.36

measure of node impurity, the highest the impurity means that each node contains only observations of a single class. Assessing the mean decrease in Gini when the variable is omitted leads to an understanding of how important that variable is to the homogeneity of the nodes, therefore, the higher the value of mean decrease Gini, the greater the importance of the variable. The relative importance for the top 10 variables of the model with the SMOTE (HVDM) method is shown in Figure 5.3.

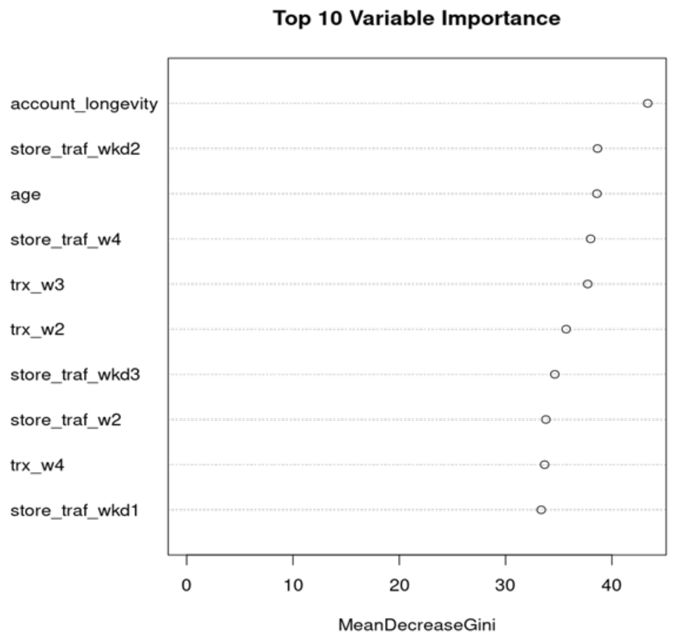


Figure 5.3: Relative importance of the top 10 variables.

From Figure 5.3 we can note that the variable regarding the customer's account longevity (*account_longevity*) contributes the most to the method's accuracy. The age of the customer also ranks high in terms of relative importance. A few variables related to the store traffic also appear to have an impact on predicting the customer's NPS, namely, the traffic on the weekends on the second (lunch), third (early afternoon) and first (morning) periods of the day, and during the week on the fourth (afternoon) and second (lunch) periods. These variables have a similar importance to the variables related to the customer's purchase habits, particularly, the percentage of transactions in the third, second and fourth period of the day during the week.

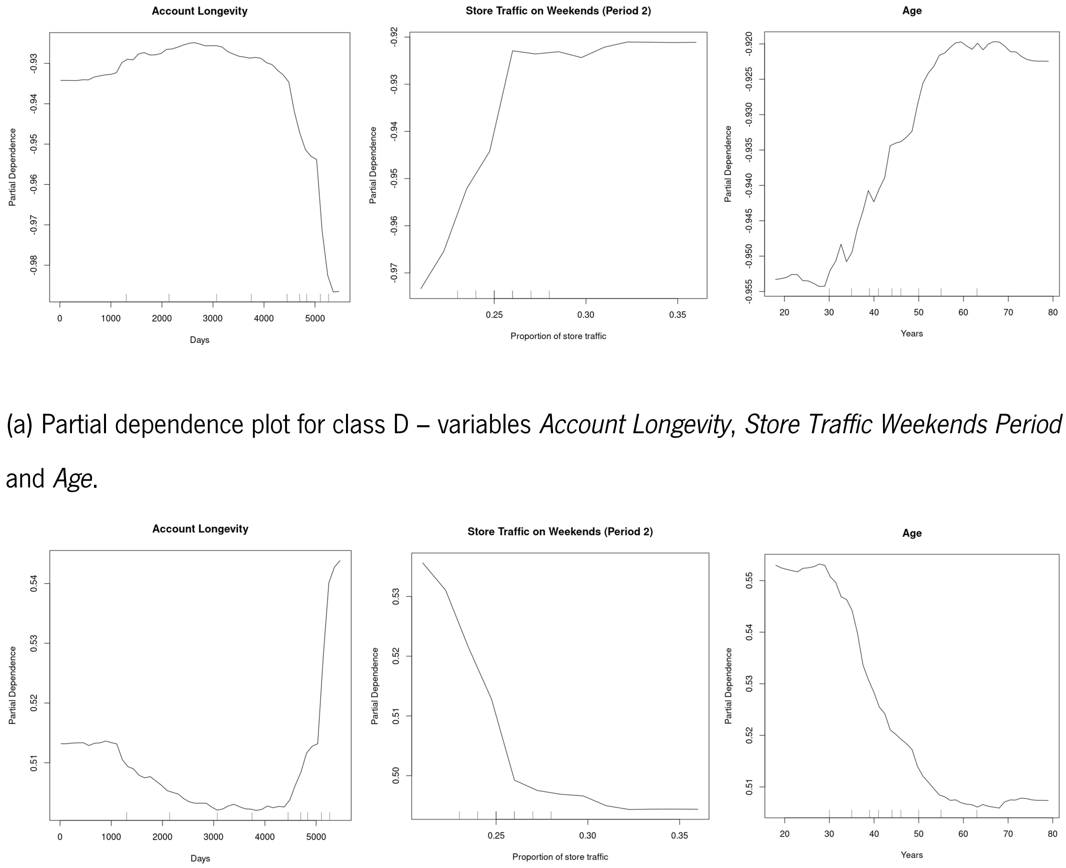
After the most relevant variables have been identified, the next step is to attempt to understand the nature of their dependence with the response variable [28]. Albeit the relative variable importance evaluates the overall impact of each variable in the method's accuracy, it does not indicate the influence of the variables values. Hence, we proceed with a further analyse of the variables through a Partial Dependence Plot (PDP). A PDP illustrates the marginal effect of the selected variables on the response after integrating out the other variables [35]. Considering the example of the PDP of the variable *Account Longevity*, it illustrates what the impact would be in the response variable, all other variables being equal, of the amount of time a consumer has had the account. On another note, a PDP can have both negative and positive values in the y-axis. The negative values illustrate that the class in question is less likely for that value of the independent variable (x-axis), according to the model, whereas positive values indicate that the class in question is more likely for that value of the independent variable. At last, zero implies no average impact on class probability.

An analysis of the top three most important variables is detailed in Figure 5.4a for class *Detractors* and Figure 5.4b for the class *Promoters*, on account of these being the key classes that influence the NPS score (Equation 2.1). For the class *Detractors*, the PDP has negative values in the y-axis, meaning that the more negative the value, the less likely it is that that variable influences the response variable. Whilst for the *Promoters*, the y-axis has positive values, indicating that the more positive the values, the higher the likelihood of that variable influencing the response variable.

By looking at the influence of the customer's account longevity on the prediction as *Detractors* 5.4a (left), we can observe that the longer a customer has had the account, it is less likely to be classified as *Detractor*. Whereas, looking at the influence on the class *Promoters* (Figure 5.4b (left)), the longer the account longevity, the more likely it is to be predicted as a *Promoter*. This result is in line with the *Promoters*' characteristics which are typically very satisfied customers and loyal to the company and, therefore, long-term customers.

The second most important variable represents the store traffic during the second period of the day on weekends. Analysing Figure 5.4a (centre) we can notice that the higher the store traffic in that period, the more chances of being predicted as a *Detractor*. In Figure 5.4b (centre) we can remark that the lower the store traffic, the more likely it is to be predicted as a *Promoter*. As observed in Figure 4.11 (right), the majority of transactions are made until the afternoon, and the stores tend to be busier during the first and third periods of the day (Figure 4.10 (right)). From that, we can deduce that the customers who buy in the second period of the day are most likely to have a less busy store and, as a consequence, a better buying experience, leaving them more satisfied and more likely to recommend the store. Therefore, the customers who shop during the second period of the day on weekends with low store traffic have a better chance of being predicted as *Promoters* and lower chance of being predicted as *Detractors*.

Lastly, the PDPs for the variable *Age* (Figure 5.4a (right) and Figure 5.4b (right)), indicate that the older the customer, the bigger the probability of being predicted as *Detractor*, whereas the younger the customer, the bigger the probability of being predicted as a *Promoter*.



(a) Partial dependence plot for class D – variables *Account Longevity*, *Store Traffic Weekends Period 2* and *Age*.

(b) Partial dependence plot for class P – variables *Account Longevity*, *Store Traffic Weekends Period 2* and *Age*.

Figure 5.4: Partial Dependence plots for the classes *Detractor* (D) and *Promoter* (P)

5.3 Error Analysis

Even though RF algorithms leverage the power of multiple DT for making decisions, the RF classifier still failed to reach AUC-measures higher than 0.55. To understand further the challenges of the task, an error analysis was performed on its results.

An error analysis helps to convey insights on how to improve the model's performance by diagnosing erroneous predictions, as well as identify if the model is behaving more erroneously for certain variables or classes. In this section the analysis is divided by NPS class, year, NPS score and misclassified Detractors.

5.3.1 By NPS Class

We start this diagnosis by analysing the errors per NPS class. Table 5.4 shows the classification error rates per class for the final data set for the different resampling methods. Along with the error rates per class, the overall error rates, the OOB error rate for the train set and the test set error rate for the random forest classifier were also added.

Table 5.4: Summary of error rates in training and testing data set.

Model	Train set Classification Error			OOB Error Rate (%)	Test set Classification Error			Test set Error Rate (%)
	D	N	P		D	N	P	
Original	1.00	1.00	0.00	29.27	1.00	0.99	0.00	29.42
ROS	0.00	0.09	0.18	9.14	0.94	0.85	0.17	37.67
RUS	0.55	0.77	0.69	66.88	0.46	0.72	0.67	63.93
SMOTE (HEOM)	0.04	0.41	0.39	28.04	0.97	0.57	0.37	44.57
SMOTE (HVDM)	0.03	0.39	0.37	26.24	0.97	0.57	0.36	43.74

The error rates in the original model reflect the importance of, not only analysing the overall error rates but also the errors per class. Especially in imbalance data sets, the overall error rate can be misleading towards the model's performance. For instance, the original data set has an imbalance class distribution, reflecting a significant majority class (*Promoters*), so, as expected, for both the train and test set, the classifier predicts with practically no error the class *Promoters*, which results in a low overall error rate. However, it misclassifies almost every observation in the classes *Passives* and *Detractors*, reflecting the model's poor performance for the minority classes.

For the method ROS, albeit with low classification errors and OOB error rate (9.14%), the classification errors in the test set for the classes *Detractors* and *Passives* is very high, 0.94 and 0.85, respectively. The same situation can be observed for the method SMOTE, for both distances, where the error rates in the train set are low, but for the test set are very high for the minority class (97% for both methods). This difference in error rates in the train versus test data set can be explained by optimistic error rates in the train set, since the model is not tested on any observations that it has not already seen. By contrast, the method RUS presents higher error rates for both the train and test set. Regardless, it gives a more feasible variation between the train and test data set classification errors. We can conjecture that the high error rates in this method are due to the small amount of observations it has available, which, in consequence, worsens the predictions and the error rates. Lastly, one obvious conclusion drawn from the results in Table 5.4 is that, overall, in unseen data, the model cannot correctly classify the minority class, the *Detractors*.

In the case of this classification problem, the RF model generates three probabilities, one for each NPS class. The predicted probabilities are pooled from different trees, and based on the majority vote in each sample, a class is predicted, namely, the class with higher predicted probability. For instance, if a given observation has a predictive probability of 0.2 for being a *Detractor*, 0.3 for *Passive* and 0.5 for *Promoter*, the observation would be classified as a *Promoter*.

For a more detailed analysis, we decided to assess the precision rate for different thresholds of the predicted probabilities for each NPS class (Figure 5.5). For that, we used the predicted scores of the model with the RUS method. In each graphic is represented the distribution of the number of observations predicted in that NPS class, i.e., the True Positive (TP) and False Positive (FP) observations. The predicted probabilities of each observation were divided into 10 classes with a range of 0.1 (e.g., probability class 1 = [0,0.1[, probability class 2 = [0.2,0.3[, etc), creating different classes of probabilities, where the higher the class, the higher the predicted probability of an observation belonging to that particular NPS class. The precision rate is represented as a line, and corresponds to the percentage of observations correctly labeled as that particular class, i.e., the percentage of TP observations.

In Figure 5.5a are represented the predicted probabilities of each observation predicted as a *Detractor*. By looking at the precision rate (orange line), we can see that the model's precision rate is very low for every probability class, not even reaching 5%, and totally failing its prediction for the observations that have an higher predictive probability of being *Detractors*. With this we can conclude that the model has great difficulty in predicting the class *Detractors*. In the case of the class *Passives* (Figure 5.5b) we can observe that the precision rate is also low, and there is not much variability between the different thresholds, indicating the difficulty of the model in predicting this class as well. On the other hand, for the class *Promoters* (Figure 5.5c) we can observe that the precision rate increases as the probabilities classes get higher, despite a small decrease in class 7. Nevertheless, the tendency for the *Promoters* is that the higher the probability, the higher the precision of the model in predicting the observations as *Promoters*.

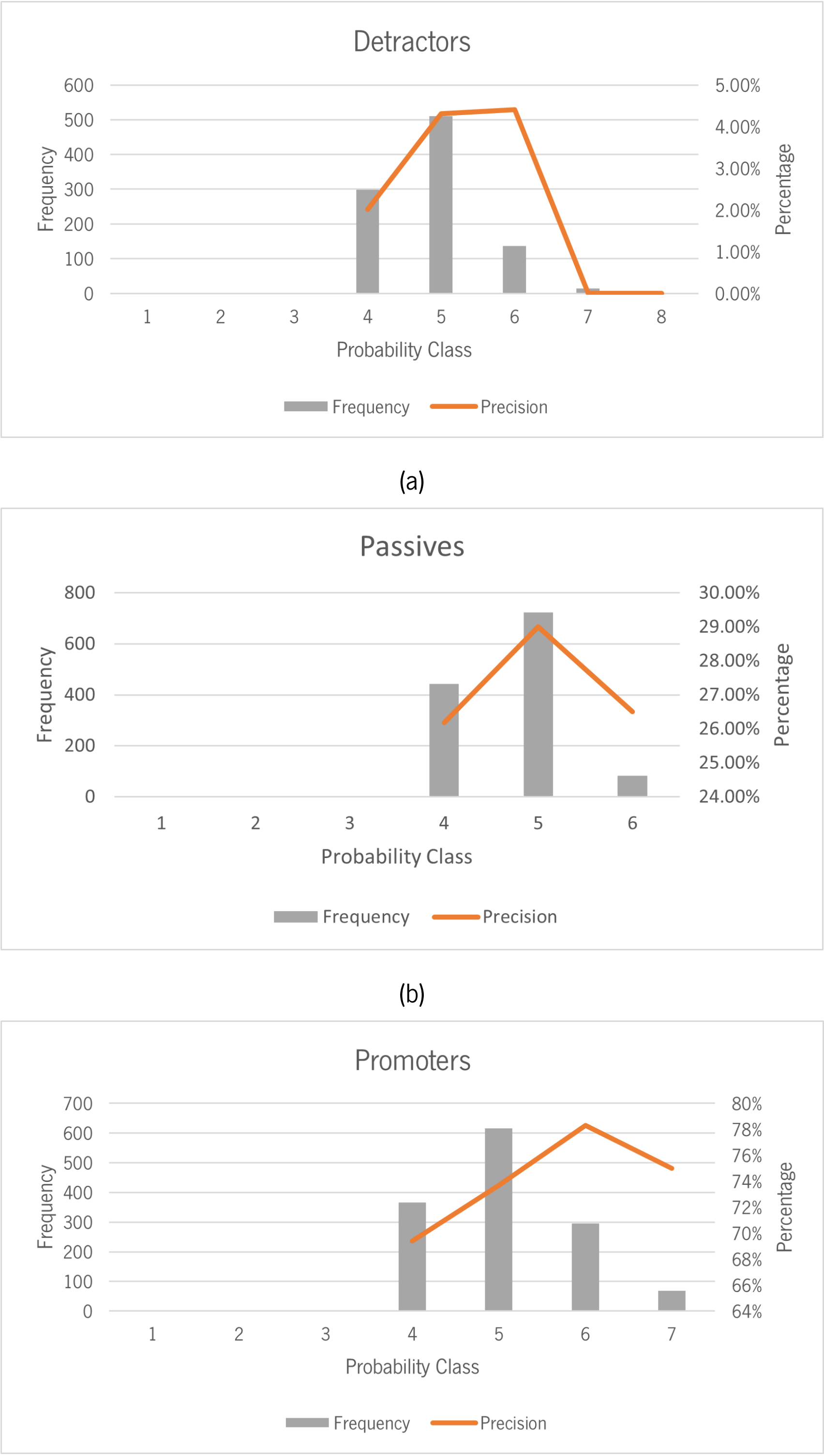
This analysis corroborates what has been concluded before, that the model has difficulty in distinguish the minority classes, the *Detractors*, followed by the *Passives*, whilst having a low difficulty in predicting the *Promoters*.

5.3.2 By Year

In a second stage, we decided to analyse the error per year. As mentioned in Section 4.1.1, the survey was conducted from september 2019 to december 2021, having 3 interrupted months in 2020. During that time the structure of the survey was also modified a few times. In order to assess if these modifications and discrepancies between the answers gathered throughout the years could influence the model's performance, a random forest classifier with random undersampling (RUS) was applied to the final data set for each year. The results in Table 5.5 for the three models show that regardless of the year, the classifier has a high classification error, on both train and test sets, for the classes *Detractors* and *Passives*, and a low classification error for the class *Promoters*, which goes in accordance with the results previously analysed. Thereupon, one can conclude that the error does not improve with any particular year.

Table 5.5: Summary of error rates in training and testing data set per year.

Year	Train set Classification Error			OOB Error Rate (%)	Test set Classification Error			Test Set Error Rate (%)
	D	N	P		D	N	P	
2019	0.84	0.63	0.26	57.57%	0.99	0.74	0.33	68.17%
2020	0.88	0.67	0.21	58.83%	0.96	0.72	0.27	64.97%
2021	0.88	0.69	0.19	58.50%	0.97	0.72	0.25	64.40%



(a)

(b)

(c)

Figure 5.5: Model's precision for each predictive probability threshold per observation for the class (a) *Detractor*, (b) *Passive* and (c) *Promoter*.

5.3.3 By NPS Score

The Net Promoter Score is measured in a scale from 0 to 10, where the customers that give out a score between 0 and 6 are classified as *Detractors*, 7 to 8 are *Passives* and 9 to 10 *Promoters*.

As previously concluded the classifiers have much difficulty in predicting the class *Detractor*. And since this class is wider than the remaining ones, we sought to analyse if the model tends to misclassify the *Detractors* similarly for all scores. Or, if on the contrary, it tends to have worst performance in the class's limits, in particular, for the higher scores close to the *Passives* scores. For this analysis the random forest classifier with the method RUS was used.

In Figure 5.6 is represented the distribution of observations per NPS score for the *Detractors*, and the respective success rate (orange line). The first thing to note is that the model misclassifies every observation with scores 0 and 1. This can be explained by the small amount of observations in those scores, which impacts the model's performance substantially. For the remainder scores, one can see that the success rate for the higher scores is much lower than for the middle scores. Applying the same analysis to the scores of the classes *Passives* and *Promoters* (Figure 5.7), it can be observed that the success rate is quite similar for all scores, not depicting the behaviour observed for the class *Detractors*. That being said, it can be concluded that for the class *Detractors*, the model has a bigger challenge in predicting the observations when the scores are 5 or 6.

For a more in-depth review of such cases, we filtered the observations with a NPS score of 5 and 6 and analysed the answers from the survey. Along with the NPS score, some customers added a note about their rating, where a few stated that their score was intermediate and even high. Thenceforth we can surmise that the scores are highly subjective, and a score of 5 and 6 for some customers can imply a 'good' score. This goes under what was observed in Figure 5.6, since some customers can be classified as *Detractors* but have a receptive attitude towards the store and its products, and so their overall behaviour can lead the model to misclassify them into another class.

5.3.4 Misclassified Detractors

Lastly, we analysed the observations belonging to the class *Detractors* that were misclassified.

Whilst answering to the survey, the customers who gave a NPS score below 7 (the *Detractors*) were also asked to name a few reasons on why they would not be so willing to recommend a Note store. In a more superficial analysis, a visual representation of the words that appeared more often

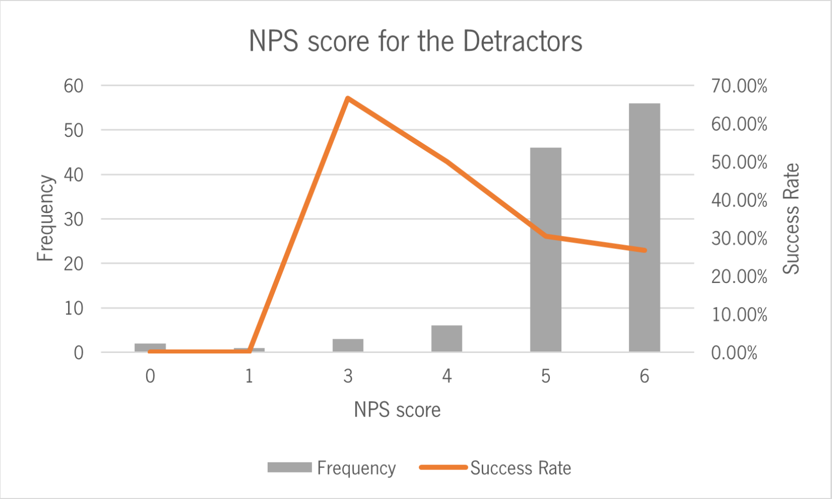


Figure 5.6: The distribution of observations per NPS score for the class *Detractors*, and the respective success rate.

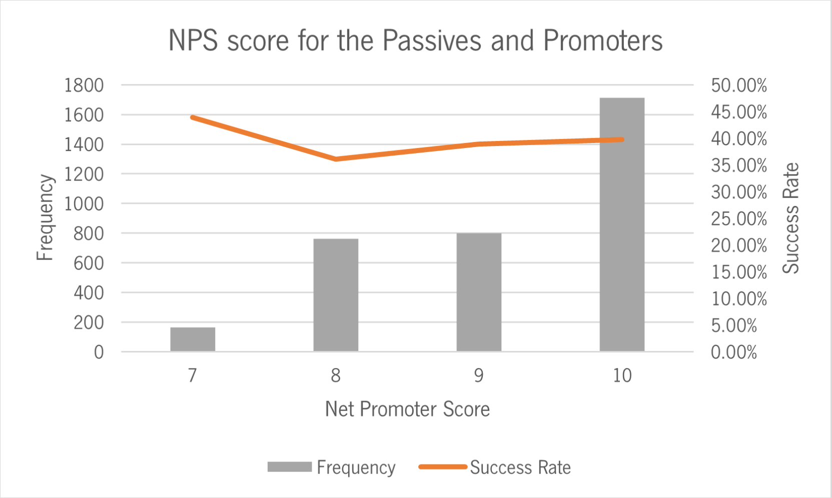


Figure 5.7: The distribution of observations per NPS score for the classes *Passives* and *Promoters*, and the respective success rate.

in the comments is represented in a word cloud in Figure 5.8.

By analysing Figure 5.8 we can observe that the most relevant words are related to the service (*atendimento*), that was bad and slow; the waiting time (*tempo de espera*) that was long; the prices for being high (*preços elevados*) and the store's limited range of products (*pouca variedade*). According to the definition in [22], these features are related to customer experience, since they originate from a set of interactions between the customer and the products and services provided by Note, and not necessarily with their loyalty. Yet, we know that positive experiences increase the

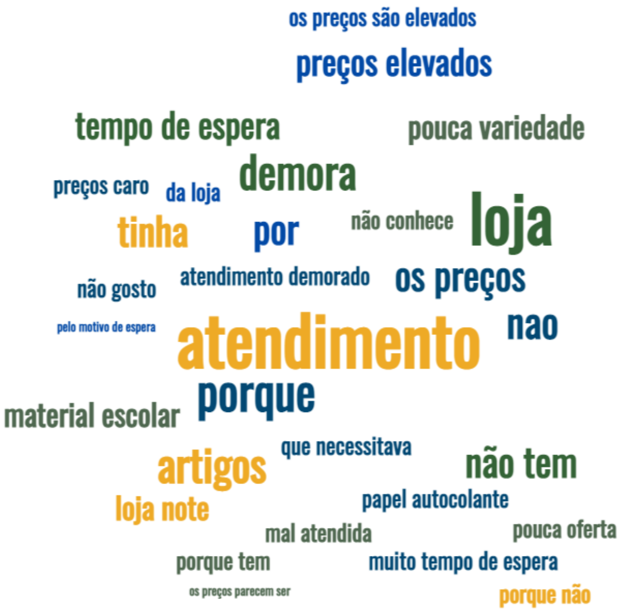


Figure 5.8: Word cloud of the most frequent words in the answers to the question: ’Can you name some reasons on why would you not be willing to recommend a Note store?’.

chances of a customer to make continued purchases and, therefore, develop brand loyalty, whereas negative experiences decrease such chances [57]. This is corroborated through the previous word cloud analysis, where the *Detractors* seem to be unsatisfied customers and thus not likely to promote a Note store nor its products. In that sense, it seems that questions regarding the customer experience should accompany the NPS analysis. Along with customer experience, it is also pivotal to measure customer satisfaction, which focuses on the attitudes or feelings that a customer forms based on their experiences [29]. And according to the authors in [4] customer satisfaction is also related to customer loyalty, which in turn is related to profitability.

While there were some questions about customer experience and customer satisfaction in the NPS survey, these were not collected throughout the entire survey period, and so were disregarded from the analysis.

In a next research step it would be important to incorporate transactional information about the customer experience and satisfaction in the different interactions, in order to have a better understanding of the customers in those aspects and potentially increase the predictive power. Along with that, the free comments that the survey customers are asked to provide should not be discarded and taken into account when analysing the data.

5.4 Business Implication

The results of this study bring some business insights with implications for the retailer. Firstly, as it was observed, Note's customers have a low penetration rate, making it difficult to have the necessary information to predict the three NPS classes. Nevertheless, the developed methodology can identify, with a reasonable error rate, the class Promoters. In a business sense, this methodology can be leveraged to distinguish the Promoters from the rest of the customers. The NPS is the result of the % *Promoters* - % *Detractors*, which means that the best way for the company to obtain better NPS is to turn Detractors and Passives into Promoters. In that sense, this prior classification alone of Promoters versus Non-Promoters is helpful for the company.

Being able to segment the customers as *Promoters* is most valuable for the retailer, since these are more likely to provide good value in long term and can benefit the company by spreading the word for attracting new customers and work as a part of their marketing department [56].

Additionally, the characterization of the *Promoters* shows that they tend to be long-term customers and to shop when there is less store traffic, which can be linked to better service and shopping experience. Also, the younger customers are most likely to rate a higher NPS score, on which we can presume that the older customers can be more demanding.

Second, by analysing the reasons for the low scores by the *Detractors*, we could identify that most answers regarded the store service and the waiting time, which are topics related to the customer experience that, consequently, reflect in customer satisfaction. These are crucial aspects to consider and to further analyse, not only as a way to reduce the *Detractors*, but also to assure and increase the *Promoters*, since loyal customers are not necessarily satisfied customers, but satisfied customers tend to be loyal, which in consequence leads to customer profitability [4]. A strategy could be to acquire feedback about the customer experience and satisfaction along with the NPS survey. One could leverage the superior knowledge to have a better understanding of which drivers influence a customer NPS.

Chapter 6

Conclusion and Future Work

The following section presents the final remarks about the project. Followed by a discussion about the project's limitations, as well as some ideas for future work.

6.1 Conclusion

As discussed in the literature review, companies are becoming more focused in understanding their customers in order to improve customer satisfaction and consequently, customer loyalty and thus increase the sales. In that sense, this project had the goal of classifying all Note's customers NPS class, that is to say, predict the customer as a Promoter, Passive or Detractor, through the use of Data Mining techniques, as well as determine the most important variables and thus understand which effects can impact the customer's classification.

Firstly, it was important to analyse the database and understand which variables could be used with the empirical experience provided by the data scientists at the case study company.

Secondly, data preparation techniques were applied to the data, in particular, resampling techniques to handle the imbalanced data set problem, since the number of Promoters was 20.81% more than the number of Detractors. Afterwards, a methodology was developed based on the three-class NPS classification problem. Two machine learning algorithms, Decision Tree (DT) and Random Forest (RF), were tested with different resampling techniques, such as Random Oversampling and Undersampling, as well as Synthetic Minority Oversampling Technique (SMOTE), which did not provide good results. Even though RF algorithms leverage the power of multiple DT for making decisions, the RF classifier still failed to reach AUC-measures higher than 0.55.

The most relevant variables turned out to be variables related to the customer account longevity, the customer's age and variables related to the store traffic and customer's transactions for different periods of the day, during the week and the weekend. We determined that the longer a customer has had the account, the more likely it is to be predicted as a Promoter. Whereas, the customers who shop during periods with more traffic have a better chance of being predicted as Detractors. Moreover, the younger the customer, the bigger the probability of being predicted as a Promoter.

In order to understand why the RF model was behaving more erroneously, an error analysis was performed. The results showed that the classifier had difficulty distinguishing the minority classes, namely, the Detractors and Passives, but it had good performance in predicting the class Promoters.

We also demonstrated that the Detractors seem to be unsatisfied customers, with bad shopping experiences. Thus our emphasis in measuring not only customer loyalty through the NPS class identification, but also measure customer experience and satisfaction. These can lead to customer loyalty, and in turn lead to profitability.

6.2 Limitations and Future Work

One limitation encountered was the retailer's low penetration rate, meaning that most of Note's customers do not use their Cartão Continente loyalty card when shopping at Note, and therefore there is not a lot of information about the customer or their transactions.

Followed by that, another limitation was that some of the customers who did use their loyalty cards had very few transactions registered. For instance, some customers had only one or two registered transactions in the previous three years, making it difficult for the models to encounter patterns and consequently predict those customers' NPS class. To potentially increase the predictive performance of the classifiers, one could include more transactional data.

In this project we demonstrated that besides analysing customer loyalty, it is also pivotal to measure the customer experience and consequently their satisfaction towards the company. And so, it would also be interesting in future studies to obtain more information about the costumer's experience, in particular, after each transaction. This way one could better pinpoint the drivers that lead to the customer's satisfaction or dissatisfaction.

Finally, in this study we aimed to determine the most important variables that impact the customer’s classification. And despite uncovering some variables with more influence to a customer being predicted into the classes Promoters and Detractors, their association might not be causal. Meaning that even if a given variable is related to an higher likelihood of being predicted into a particular NPS class, it does not mean that that variable causes the customer to be a Promoter, or Detractor or Passive. While causation and correlation can exist simultaneously, correlation does not imply causation. That way, in future studies it would be interesting to analyse the causal relationship between the variables in study and the three NPS classes. This would be helpful for leveraging the right insights to increase sales growth, such as whether certain variables cause customer retention or engagement.

Bibliography

[1] C. C. Aggarwal. *Data Classification: Algorithms and Applications*. Chapman and Hall/CRC, 2014.

[2] S. Ahmed. Applications of data mining in retail business. In *International Conference on Information Technology: Coding and Computing*, volume 2, pages 455–459, 2004.

[3] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain. Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *IEEE Access*, 4:7940–7957, 2016.

[4] E. W. Anderson, C. Fornell, and D. R. Lehmann. Customer satisfaction, market share, and profitability: Findings from sweden. *Journal of Marketing*, 58(3):53–66, 1994.

[5] H. Ba, S. Guo, Y. Wang, X. Hong, Y. Zhong, and Z. Liu. Improving ann model performance in runoff forecasting by adding soil moisture input and using data preprocessing techniques. *Hydrology Research*, 49(3):744–760, 2017.

[6] G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[7] S. Bernard, L. Heutte, and S. Adam. Influence of hyperparameters on random forest accuracy. Lecture Notes in Computer Science, pages 171–180. Springer, Springer, 2009.

[8] R. N. Bolton and J. H. Drew. A multistage model of customers' assessments of service quality and value. *Journal of Consumer Research*, 17(4):375–384, 1991.

[9] P. Branco, R. P. Ribeiro, and L. Torgo. Ubl: an r package for utility-based learning. 2016.

[10] P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modelling under imbalanced distributions. *arXiv*, 2015.

[11] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] J. Brownlee. 4 types of classification tasks in machine learning, 2020.

[13] J. Bughin, J. Doogan, and O. J. Vetvik. A new way to measure word-of-mouth marketing. *McKinsey Quarterly*, 2010.

[14] N. Chawla. C4.5 and imbalanced data sets: Investigating the eect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML*, volume 3, page 66, 2003.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[16] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: Special issue on learning from imbalanced data sets. *Association for Computing Machinery*, 6(1):1–6, 2004.

[17] J. A. Chevalier and D. Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354.

[18] R. De Oliveira, R. C. Fernandes Araújo, F. Barros, A. Segundo, R. Zampolo, W. Fonseca, V. Dmitriev, and F. Brasil. A system based on artificial neural networks for automatic classification of hydro-generator stator windings partial discharges. *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, 16:628–645, 2017.

[19] G. Douzas, F. Bacao, J. Fonseca, and M. Khudinyan. Imbalanced learning in land cover classification: Improving minority classes' prediction accuracy using the geometric smote algorithm. *Remote Sensing*, 11(24):3040, 2019.

[20] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[21] S. Garcia, J. Luengo, and F. Herrera. *Data Preprocessing in Data Mining*. Springer, 2015.

[22] C. Gentile, N. Spiller, and G. Noci. How to sustain the customer experience:: An overview of experience components that co-create value with the customer. *European Management Journal*, 25(5):395–410, 2007.

[23] M. M. Ghiasi, S. Zendehboudi, and A. A. Mohsenipour. Decision tree-based diagnosis of coronary artery disease: Cart model. *Computer Methods and Programs in Biomedicine*, 192, 2020.

[24] R. Goldschmidt and M. Passos. *Data Mining: Um Guia Prático*. Elsevier Editora Ltda., 2005.

[25] M. Grandini, E. Bagli, and G. Visani. Metrics for multi-class classification: an overview. *arXiv*, 2020.

[26] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[27] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.

[28] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[29] N. Hill, G. Roche, and R. Allen. *Customer Satisfaction: The customer experience through the customer's eyes*. Cogent Publishing, 2007.

[30] U. R. Hodeghatta and U. Nayak. *Business Analytics Using R - A Practical Approach*. Apress, Berkeley, 2017.

[31] M. B. Holbrook and E. C. Hirschman. The experiential aspects of consumption: Consumer fantasies, feelings, and fun. *Journal of Consumer Research*, 9(2):132–140, 1982.

[32] M. Hossin and S. M.N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5(2):1–11, 2015.

[33] J. Huang and C. X. Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[34] A. Y. Hussein, P. Falcarin, and A. T. Sadiq. Enhancement performance of random forest algorithm via one hot encoding for iot ids. *Periodicals of Engineering and Natural Sciences (PEN)*, 9(33):579–591, 2021.

[35] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*, volume 103. Springer, 2013.

[36] D. R. Jeske, T. P. Callanan, and L. Guo. Identification of key drivers of net promoter score using a statistical classification model.

[37] B. Jijo and A. Mohsin Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(1), 2021.

[38] V. Kumar, V. Chattaraman, C. Neghina, B. Skiera, L. Aksoy, A. Buoye, and J. Henseler. Data-driven services marketing in a connected world. *Journal of Service Management*, 24(3):330–352, 2013.

[39] V. J. Lei, E. H. Kennedy, T. Luong, X. Chen, D. E. Polsky, K. G. Volpp, M. D. Neuman, J. H. Holmes, L. A. Fleisher, and A. S. Navathe. Model performance metrics in assessing the value of adding intraoperative data for death prediction: Applications to noncardiac surgery. *Studies in Health Technology and Informatics*, 264:223–227, 2019.

[40] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[41] G. S. Linoff and M. J. A. Berry. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. Wiley, 2011.

[42] M. R. Longadge, S. S. Dongre, and D. L. Malik. Class imbalance problem in data mining: Review. 2(1), 2013.

[43] V. López, A. Fernández, and F. Herrera. On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. *Information Sciences*, 257:1–13, 2014.

[44] M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath. Loan default prediction using decision trees and random forest: A comparative study. *IOP Conference Series: Materials Science and Engineering*, 1022(1), 2021.

[45] P. C. Mandal. Net promoter score: a conceptual analysis. *International Journal of Management Concepts and Philosophy*, 8(4):209–219, 2014.

[46] R. G. Mantovani, T. Horváth, R. Cerri, S. B. Junior, J. Vanschoren, and A. C. P. d. L. F. de Carvalho. An empirical study on hyperparameter tuning of decision trees. *arXiv*, 2018.

[47] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis. A machine learning based classification method for customer experience survey analysis. *Technologies*, 8(4):76, 2020.

[48] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, J. H. Orallo, M. Kull, N. Lachiche, M. J. R. Quintana, and P. A. Flach. Crisp-dm twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8):3048–3061, 2021.

[49] G. G. Moisen. *Classification and Regression Trees*, volume 1. Academic Press, Oxford, 2008.

[50] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[51] E. Ngai, L. Xiu, and D. Chau. Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36(2):2592–2602, 2009.

[52] P. Probst, M. N. Wright, and A.-L. Boulesteix. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3), 2019.

[53] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[54] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[55] F. Reichheld and R. Markey. *The Ultimate Question 2.0: How Net Promoter Companies Thrive in a Customer-Driven World*. Harvard Business Review Press, 2011.

[56] F. F. Reichheld. The one number you need to grow. *Harvard Business Review*, 81(12):46–55, 2003.

[57] L. Ren, H. Qiu, P. Wang, and P. M. C. Lin. Exploring customer experience with budget hotels: Dimensionality and satisfaction. *International Journal of Hospitality Management*, 52:13–23, 2016.

[58] L. Rokach and O. Maimon. *Data Mining with Decision Trees Theory and Applications*. World Scientific, 2014.

[59] G. Sanson, J. Welton, E. Vellone, A. Cocchieri, M. Maurici, M. Zega, R. Alvaro, and F. D'Agostino. Enhancing the performance of predictive models for hospital mortality by adding nursing data. *International Journal of Medical Informatics*, 125:79–85, 2019.

[60] M. Schauerhuber, A. Zeileis, D. Meyer, and K. Hornik. Benchmarking open-source tree learners in r/rweka. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, 2008.

[61] F. Shakeel, A. S. Sabhitha, and S. Sharma. Exploratory review on class imbalance problem: An overview. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–8. IEEE, 2017.

[62] R. Sharda, D. Delen, and E. Turban. *Business Intelligence and Analytics: Systems for Decision Support*. Pearson Education, Inc., 2007.

[63] C. Shearer. The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4), 2000.

[64] Á. Silva, P. Cortez, M. F. Santos, L. Gomes, and J. Neves. Rating organ failure via adverse events using data mining in the intensive care unit. *Artificial intelligence in medicine*, 43(3):179–193, 2008.

[65] S. Singh and P. Gupta. Comparative study id3, cart and c4.5 decision tree algorithm: A survey. *International Journal of Advanced Information Science and Technology*, 27(27):81–106, 2014.

[66] M. Sridharan. Crisp-dm - a framework for data mining analysis. `https://thinkinsights.net/data-literacy/crisp-dm/`, 2018.

[67] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto. Automated parameter optimization of classification techniques for defect prediction models. 38th IEEE International Conference on Software Engineering. Association for Computing Machinery, 2016.

[68] T. M. Therneau and E. J. Atkinson. An introduction to recursive partitioning using the rpart routines. Technical report, Technical report Mayo Foundation, 2022.

[69] I. Tomek. Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, 1976.

[70] L. Tong, Y. Wang, F. Wen, and X. Li. The research of customer loyalty improvement in telecom industry based on nps data mining. *China Communications*, 14(11):260–268, 2017.

[71] L. Torgo. *Data Mining with R Learning with Case Studies*. Chapman and Hall/CRC, 2nd edition, 2020.

[72] B. Tuychiev. Comprehensive guide to multiclass classification metrics, 2021.

[73] T. Waheed, R. B. Bonnell, S. O. Prasher, and E. Paulet. Measuring performance in precision agriculture: Cart — a decision tree approach. *Agricultural Water Management*, 84(1):173–185, 2006.

[74] G. M. Weiss and F. Provost. Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19:315–354, 2003.

[75] D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.

[76] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[77] S. P. Yip and G. I. Webb. Empirical function attribute construction in classification learning. In *Joint conference on artificial intelligence (AI'94)*, pages 29–36, 1994.

[78] X. Zhou, P. Lu, Z. Zheng, D. Tolliver, and A. Keramati. Accident prediction accuracy assessment for highway-rail grade crossings using random forest algorithm compared with decision tree. *Reliability Engineering System Safety*, 200:106931, 2020.

[79] R. Zhu, Y. Guo, and J.-H. Xue. Adjusting the imbalance ratio by the dimensionality of imbalanced data. *Pattern Recognition Letters*, 133:217–223, 2020.

Appendix A

Variables of the Final Data set

The following tables present, in more detail, the variables that were used to build the predictive models. These variables are divided into several categories: Customer Profile, Purchase Behaviour and Store Profile.

A.1 Customer Profile Variables

Table A.1: Predictive variables of the final data set - Customer Profile

Name	Description	Variables	Values
Age	Age	age	
Account Longevity	The number of years the customer has had the account	account_longevity	
GenderF	Inferred female behaviour	genderF	0, 1
GenderM	Inferred male behaviour	genderM	0, 1
Slifestyle	Lifestyle Segmentation	slifestyle	1,2,3,4,5,6,7,8
Svcontinente	Continente Value Segmentation	svcontinente	1,2,3,4,5,6,7,8
Svnote	Note Value Segmentation	svnote	1,2,3,4,5,6,7,8,9
Slifestage	Lifestage Segmentation	slifestage	1,2,3,4,5,6

A.2 Purchase Behaviour Variables

Table A.2: Predictive variables of the final data set - Purchase Behaviour (Part I)

Name	Description	Variables
Direct Discount	Direct discount benefitted in sales and returns (%)	sales_direct_disc, returns_direct_disc, sales_direct_disc_l6m, returns_direct_disc_l6m, sales_direct_disc_l3m, returns_direct_disc_l3m sales_direct_disc_lm, returns_direct_disc_lm
Deferred Discount	Deferred discount benefitted in sales and returns (%)	sales_deferred_disc, returns_deferred_disc, sales_deferred_disc_l6m, returns_deferred_disc_l6m, sales_deferred_disc_l3m, returns_deferred_disc_l3m sales_deferred_disc_lm, returns_deferred_disc_lm
Basket	Sales basket: The gross amount spent in a given transaction Returns basket: The gross amount returned in a given transaction	cesta_sales_media, cesta_sales_max,cesta_sales_min, cesta_sales_sd cesta_returns_media, cesta_returns_max, cesta_returns_min, cesta_returns_sd, cesta_sales_media_l6m, cesta_sales_max_l6m, cesta_sales_min_l6m, cesta_sales_sd_l6m cesta_returns_media_l6m, cesta_returns_max_l6m, cesta_returns_min_l6m, cesta_returns_sd_l6m, cesta_sales_media_l3m, cesta_sales_max_l3m, cesta_sales_min_l3m, cesta_sales_sd_l3m cesta_returns_media_l3m, cesta_returns_max_l3m, cesta_returns_min_l3m, cesta_returns_sd_l3m, cesta_sales_media_lm, cesta_sales_max_lm, cesta_sales_min_lm, cesta_sales_sd_lm, cesta_returns_media_lm, cesta_returns_max_lm, cesta_returns_min_lm, cesta_returns_sd_lm
Quantity	Number of items bought	qty_sales_media, qty_sales_max, qty_sales_min, qty_sales_sd, qty_sales_total, qty_returns_media, qty_returns_max, qty_returns_min, qty_returns_sd, qty_returns_total, qty_sales_media_l6m, qty_sales_max_l6m, qty_sales_min_l6m, qty_sales_sd_l6m, qty_sales_total_l6m qty_returns_media_l6m, qty_returns_max_l6m, qty_returns_min_l6m, qty_returns_sd_l6m, qty_returns_total_l6m qty_sales_media_l3m, qty_sales_max_l3m, qty_sales_min_l3m, qty_sales_sd_l3m, qty_sales_total_l3m qty_returns_media_l3m, qty_returns_max_l3m, qty_returns_min_l3m, qty_returns_sd_l3m, qty_returns_total_l3m qty_sales_media_lm , qty_sales_max_lm , qty_sales_min_lm, qty_sales_sd_lm, qty_sales_total, qty_returns_media_lm, qty_returns_max_lm, qty_returns_min_lm, qty_returns_sd_lm, qty_returns_total_lm

Table A.3: Predictive variables of the final data set - Purchase Behaviour (Part II)

Name	Description	Variables
SKU	Number of SKUs (SKU=product identifier)	n_sku_total, n_sku_md, n_sku_max, n_sku_min ,n_sku_sd n_sku_total, n_sku_md, n_sku_max, n_sku_min ,n_sku_sd n_sku_total_l3m, n_sku_md_l3m, n_sku_max_l3m, n_sku_min_l3m ,n_sku_sd_l3m n_sku_total_lm, n_sku_md_lm, n_sku_max_lm, n_sku_min_lm ,n_sku_sd_lm
Price	Price of the product	Maximum Price: preco_max_sales_md, preco_max_sales_sd, preco_max_sales_min, preco_max_sales_max preco_max_returns_md, preco_max_returns_sd, preco_max_returns_min, preco_max_returns_max, preco_max_sales_md_l6m, preco_max_sales_sd_l6m, preco_max_sales_min_l6m, preco_max_sales_max_l6m preco_max_returns_md_l6m, preco_max_returns_sd_l6m, preco_max_returns_min_l6m, preco_max_returns_max_l6m preco_max_sales_md_l3m, preco_max_sales_sd_l3m, preco_max_sales_min_l3m, preco_max_sales_max_l3m preco_max_returns_md_l3m, preco_max_returns_sd_l3m, preco_max_returns_min_l3m, preco_max_returns_max_l3m Minimum Price: preco_min_sales_md, preco_min_sales_sd, preco_min_sales_min, preco_min_sales_max preco_min_returns_md, preco_min_returns_sd, preco_min_returns_min, preco_min_returns_max (...) Average Price: preco_medio_sales_md, preco_medio_sales_sd, preco_medio_sales_max, preco_medio_sales_min preco_medio_returns_md, preco_medio_returns_sd, preco_medio_returns_max, preco_medio_returns_min(...) Standard Deviation Price: preco_sd_sales_md, preco_sd_sales_sd, preco_sd_sales_max, preco_sd_sales_min preco_sd_returns_md, preco_sd_returns_sd, preco_sd_returns_max, preco_sd_returns_min (...)
Subcategory	Number of subcategories of the product	n_subcat_total, n_subcat_md, n_subcat_max, n_subcat_min, n_subcat_sd

Table A.4: Predictive variables of the final data set - Purchase Behaviour (Part III)

Name	Description	Variables
Category	Number of categories of the product	n_cat_total, n_cat_md, n_cat_max, n_cat_min, n_cat_sd n_cat_total_l6m, n_cat_md_l6m, n_cat_max_l6m, n_cat_min_l6m, n_cat_sd_l6m n_cat_total_l3m, n_cat_md_l3m, n_cat_max_l3m, n_cat_min_l3m, n_cat_sd_l3m n_cat_total_lm, n_cat_md_lm, n_cat_max_lm, n_cat_min_lm, n_cat_sd_lm
BU	Number of business units of the product	n_bu_total, n_bu_md, n_bu_max , n_bu_min , n_bu_sd n_bu_total_l6m, n_bu_md_ulm, n_bu_max_l6m , n_bu_min_l6m , n_bu_sd_l6m n_bu_total_l3m, n_bu_md_l3m, n_bu_max_l3m , n_bu_min_l3m , n_bu_sd_l3m n_bu_total_lm, n_bu_md_lm, n_bu_max_lm , n_bu_min_lm , n_bu_sd_lm
Transactions	Number of transactions	n_trx_sales, n_trx_returns, n_trx_total n_trx_sales_l6m, n_trx_returns_l6m, n_trx_total_l6m n_trx_sales_l3m, n_trx_returns_l3m, n_trx_total_l3m n_trx_sales_lm, n_trx_returns_lm, n_trx_total_lm
App Transactions	Number of transactions on the app	n_trx_app, n_trx_app_l6m, n_trx_app_l3m, n_trx_app_lm
Recency	Number of days since the last transaction up until the survey	
Mission	The percentage of transactions in each mission	livros, universo_infantil, material_escritorio, null, quiosque, presentes, material_papelaria_escolar, apoio_escolar, mochilas, material_funcional, servicos

A.3 Store Profile Variables

Table A.5: Predictive variables of the final data set - Store Profile

Name	Description	Variables	Values
DOP	Operacional Direction	DOP	centro, porto, insco, lisboa, norte, sul
Concept	Store antiquity concept	concept	old,intermidiate, new
Store type	Type of store based on its area and the range of products it sells	type_store	tendencia, cultura, express, insco tendencia, tendencia (outlier), online
Store traffic	The percentage of traffic during the week and weekend for each period of the day	Week traffic variables: store_traf_w1,store_traf_w2, store_traf_w3, store_traf_w4,store_traf_w5 Weekend traffic variables: store_traf_wkd1, store_traf_wkd2, store_traf_wkd3,store_traf_wkd4, store_traf_wkd5	
Transactions Distribution	The percentage of transactions for each period of the day during the week and weekend	Transactions during the week variables: trx_w1, trx_w2, trx_w3, trx_w4,trx_w5 Transactions during the weekend variables: trx_wkn_1, trx_wkn_2, trx_wkn_3, trx_wkn_4, trx_wkn_5	

Appendix B

Exploratory Analysis on the Data

The following table presents the results of the Kolmogorov-Smirnov normality test.

B.1 Kolmogorov-Smirnov Normality Test

Table B.1: Results of the Kolmogorov-Smirnov normality test

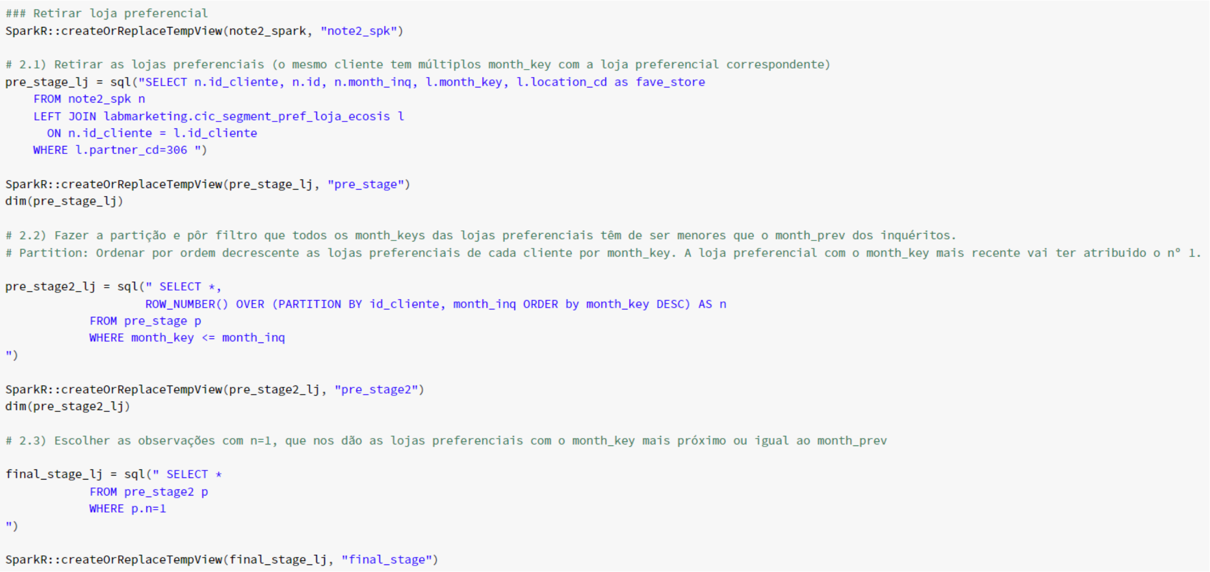
Name	D	P Value
Age	0.075	< 0.0001
Account Longetivity	0.181	
Gross Amount	0.324	
Basket	0.199	
Quantity	0.431	
SKU	0.306	
Category	0.337	
Transactions	0.358	
App Transactions	0.407	
Recency	0.231	

Appendix C

Data Preparation

In the following figures there is a description of some of the R code that was implemented to create, extract and construct variables.

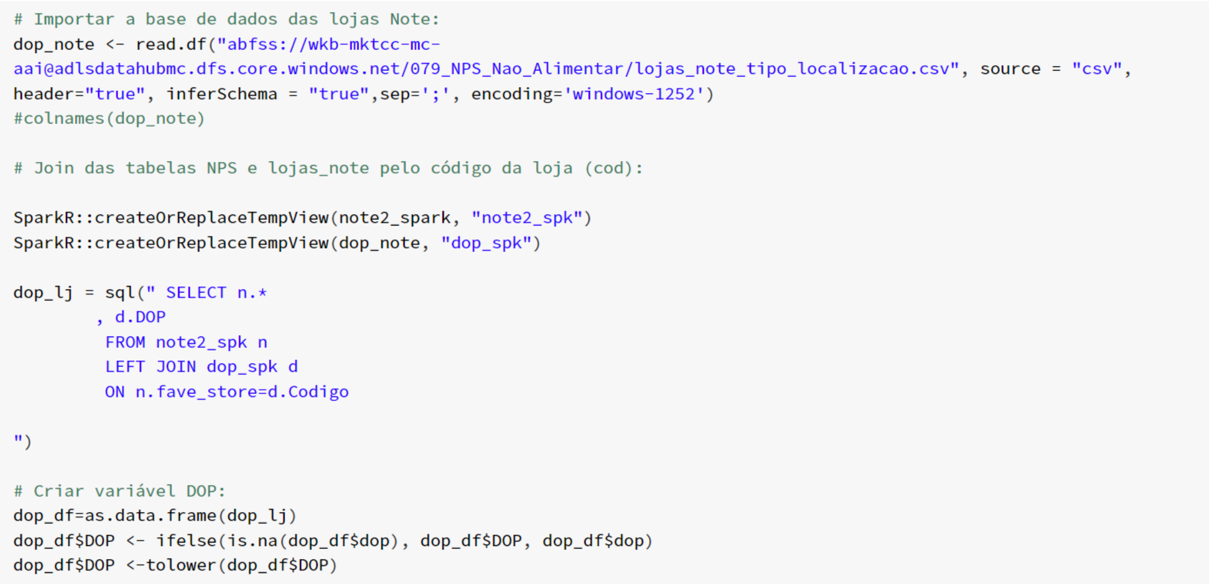
C.1 Creating the Variable fave_store



```
### Retirar loja preferencial
SparkR::createOrReplaceTempView(note2_spark, "note2_spk")

# 2.1) Retirar as lojas preferenciais (o mesmo cliente tem múltiplos month_key com a loja preferencial correspondente)
pre_stage_lj = sql("SELECT n.id_cliente, n.id, n.month_inq, l.month_key, l.location_cd as fave_store
    FROM note2_spk n
    LEFT JOIN labmarketing.cic_segment_pref_loja_ecosis l
      ON n.id_cliente = l.id_cliente
    WHERE l.partner_cd=306 ")

SparkR::createOrReplaceTempView(pre_stage_lj, "pre_stage")
dim(pre_stage_lj)

# 2.2) Fazer a partição e pôr filtro que todos os month_keys das lojas preferenciais têm de ser menores que o month_prev dos inquéritos.
# Partition: Ordenar por ordem decrescente as lojas preferenciais de cada cliente por month_key. A loja preferencial com o month_key mais recente vai ter atribuido o nº 1.

pre_stage2_lj = sql(" SELECT *,
                    ROW_NUMBER() OVER (PARTITION BY id_cliente, month_inq ORDER by month_key DESC) AS n
            FROM pre_stage p
            WHERE month_key <= month_inq
")

SparkR::createOrReplaceTempView(pre_stage2_lj, "pre_stage2")
dim(pre_stage2_lj)

# 2.3) Escolher as observações com n=1, que nos dão as lojas preferenciais com o month_key mais próximo ou igual ao month_prev

final_stage_lj = sql(" SELECT *
            FROM pre_stage2 p
            WHERE p.n=1
")

SparkR::createOrReplaceTempView(final_stage_lj, "final_stage")
```

Figure C.1: Creating the variable *fave_store*.

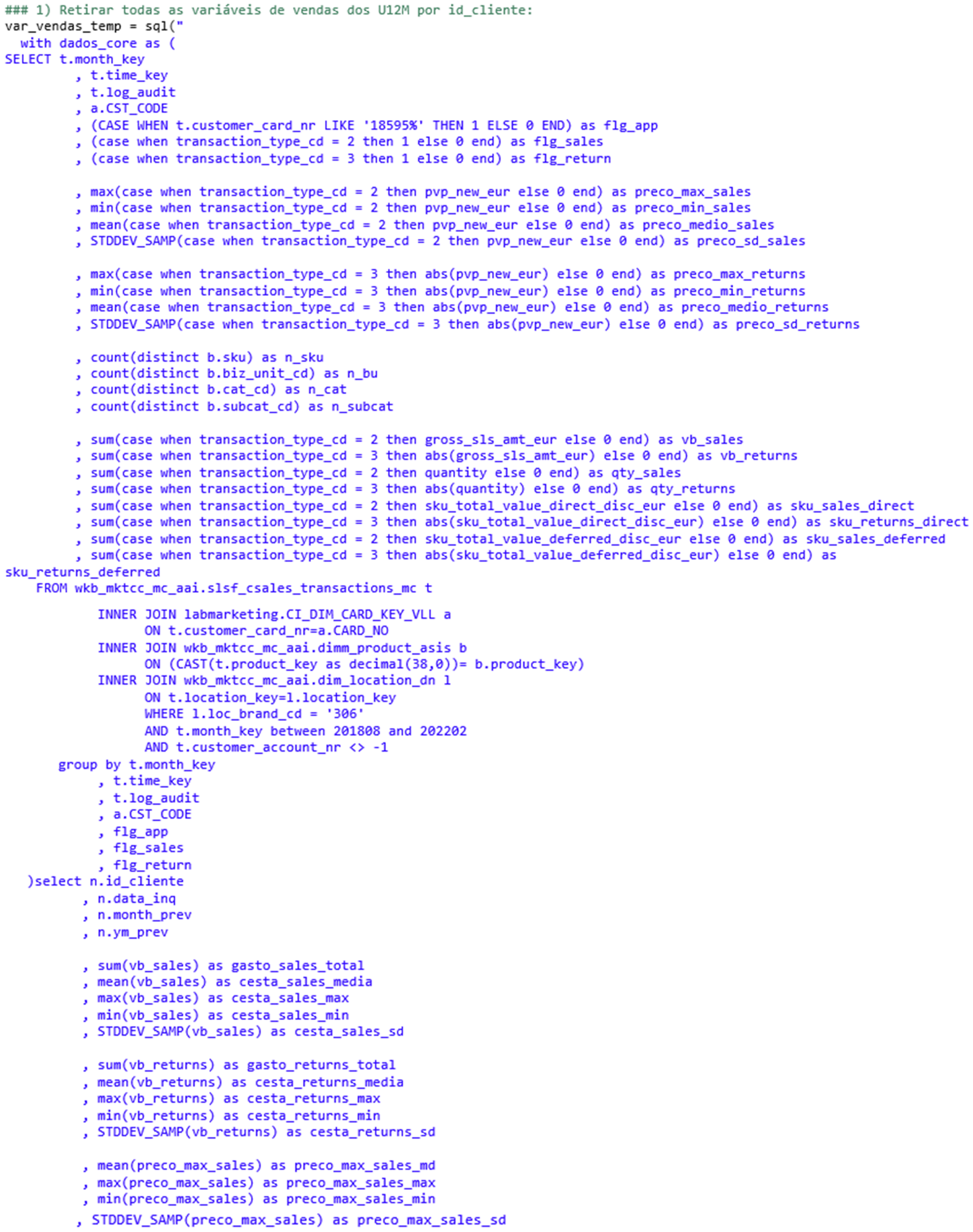
C.2 Handling the Missing Values of the Variable DOP



```
# Importar a base de dados das lojas Note:
dop_note <- read.df("abfss://wkb-mktcc-mc-
aai@adlsdatahubmc.dfs.core.windows.net/079_NPS_Nao_Alimentar/lojas_note_tipo_localizacao.csv", source = "csv",
header="true", inferSchema = "true",sep=';', encoding='windows-1252')
#colnames(dop_note)

# Join das tabelas NPS e lojas_note pelo código da loja (cod):

SparkR::createOrReplaceTempView(note2_spark, "note2_spk")
SparkR::createOrReplaceTempView(dop_note, "dop_spk")

dop_lj = sql(" SELECT n.*
        , d.DOP
         FROM note2_spk n
         LEFT JOIN dop_spk d
         ON n.fave_store=d.Codigo

")

# Criar variável DOP:
dop_df=as.data.frame(dop_lj)
dop_df$DOP <- ifelse(is.na(dop_df$dop), dop_df$DOP, dop_df$dop)
dop_df$DOP <-tolower(dop_df$DOP)
```

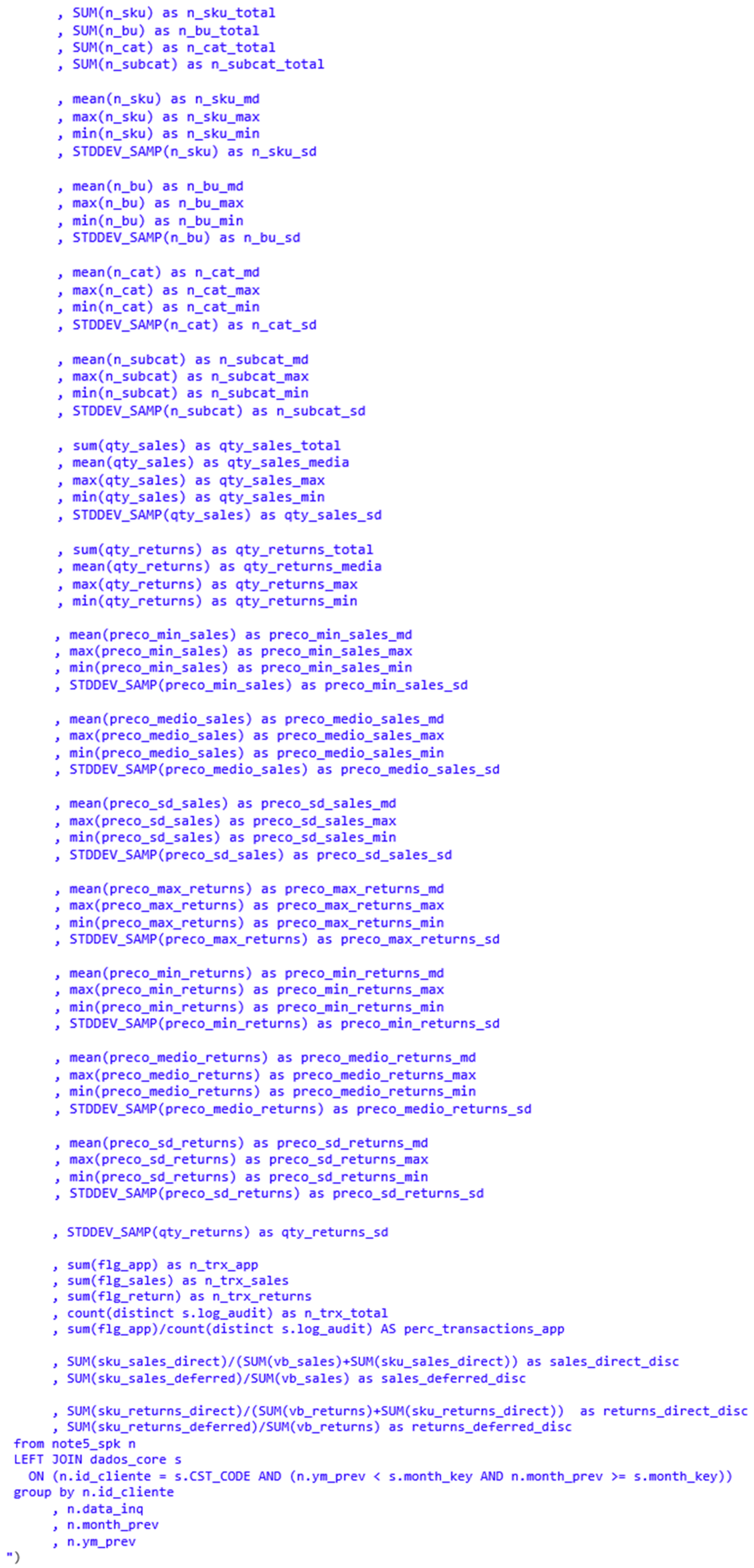
Figure C.2: Imputation of missing values for the variable DOP.

C.3 Variable Extraction and Construction



```
### 1) Retirar todas as variáveis de vendas dos U12M por id_cliente:
var_vendas_temp = sql("
  with dados_core as (
SELECT t.month_key
         , t.time_key
         , t.log_audit
         , a.CST_CODE
         , (CASE WHEN t.customer_card_nr LIKE '18595%' THEN 1 ELSE 0 END) as flg_app
         , (case when transaction_type_cd = 2 then 1 else 0 end) as flg_sales
         , (case when transaction_type_cd = 3 then 1 else 0 end) as flg_return

         , max(case when transaction_type_cd = 2 then pvp_new_eur else 0 end) as preco_max_sales
         , min(case when transaction_type_cd = 2 then pvp_new_eur else 0 end) as preco_min_sales
         , mean(case when transaction_type_cd = 2 then pvp_new_eur else 0 end) as preco_medio_sales
         , STDDEV_SAMP(case when transaction_type_cd = 2 then pvp_new_eur else 0 end) as preco_sd_sales

         , max(case when transaction_type_cd = 3 then abs(pvp_new_eur) else 0 end) as preco_max_returns
         , min(case when transaction_type_cd = 3 then abs(pvp_new_eur) else 0 end) as preco_min_returns
         , mean(case when transaction_type_cd = 3 then abs(pvp_new_eur) else 0 end) as preco_medio_returns
         , STDDEV_SAMP(case when transaction_type_cd = 3 then abs(pvp_new_eur) else 0 end) as preco_sd_returns

         , count(distinct b.sku) as n_sku
         , count(distinct b.biz_unit_cd) as n_bu
         , count(distinct b.cat_cd) as n_cat
         , count(distinct b.subcat_cd) as n_subcat

         , sum(case when transaction_type_cd = 2 then gross_sls_amt_eur else 0 end) as vb_sales
         , sum(case when transaction_type_cd = 3 then abs(gross_sls_amt_eur) else 0 end) as vb_returns
         , sum(case when transaction_type_cd = 2 then quantity else 0 end) as qty_sales
         , sum(case when transaction_type_cd = 3 then abs(quantity) else 0 end) as qty_returns
         , sum(case when transaction_type_cd = 2 then sku_total_value_direct_disc_eur else 0 end) as sku_sales_direct
         , sum(case when transaction_type_cd = 3 then abs(sku_total_value_direct_disc_eur) else 0 end) as sku_returns_direct
         , sum(case when transaction_type_cd = 2 then sku_total_value_deferred_disc_eur else 0 end) as sku_sales_deferred
         , sum(case when transaction_type_cd = 3 then abs(sku_total_value_deferred_disc_eur) else 0 end) as
sku_returns_deferred
    FROM wkb_mktcc_mc_aai.slsf_csales_transactions_mc t

            INNER JOIN labmarketing.CI_DIM_CARD_KEY_VLL a
                  ON t.customer_card_nr=a.CARD_NO
            INNER JOIN wkb_mktcc_mc_aai.dimm_product_asis b
                  ON (CAST(t.product_key as decimal(38,0))= b.product_key)
            INNER JOIN wkb_mktcc_mc_aai.dim_location_dn l
                  ON t.location_key=l.location_key
                  WHERE l.loc_brand_cd = '306'
                  AND t.month_key between 201808 and 202202
                  AND t.customer_account_nr <> -1
       group by t.month_key
            , t.time_key
            , t.log_audit
            , a.CST_CODE
            , flg_app
            , flg_sales
            , flg_return
   )select n.id_cliente
          , n.data_inq
          , n.month_prev
          , n.ym_prev

          , sum(vb_sales) as gasto_sales_total
          , mean(vb_sales) as cesta_sales_media
          , max(vb_sales) as cesta_sales_max
          , min(vb_sales) as cesta_sales_min
          , STDDEV_SAMP(vb_sales) as cesta_sales_sd

          , sum(vb_returns) as gasto_returns_total
          , mean(vb_returns) as cesta_returns_media
          , max(vb_returns) as cesta_returns_max
          , min(vb_returns) as cesta_returns_min
          , STDDEV_SAMP(vb_returns) as cesta_returns_sd

          , mean(preco_max_sales) as preco_max_sales_md
          , max(preco_max_sales) as preco_max_sales_max
          , min(preco_max_sales) as preco_max_sales_min
         , STDDEV_SAMP(preco_max_sales) as preco_max_sales_sd
```

Figure C.3: Extraction and variable construction of some the variables of Purchase Behaviour - Part I



```
      , SUM(n_sku) as n_sku_total
      , SUM(n_bu) as n_bu_total
      , SUM(n_cat) as n_cat_total
      , SUM(n_subcat) as n_subcat_total

      , mean(n_sku) as n_sku_md
      , max(n_sku) as n_sku_max
      , min(n_sku) as n_sku_min
      , STDDEV_SAMP(n_sku) as n_sku_sd

      , mean(n_bu) as n_bu_md
      , max(n_bu) as n_bu_max
      , min(n_bu) as n_bu_min
      , STDDEV_SAMP(n_bu) as n_bu_sd

      , mean(n_cat) as n_cat_md
      , max(n_cat) as n_cat_max
      , min(n_cat) as n_cat_min
      , STDDEV_SAMP(n_cat) as n_cat_sd

      , mean(n_subcat) as n_subcat_md
      , max(n_subcat) as n_subcat_max
      , min(n_subcat) as n_subcat_min
      , STDDEV_SAMP(n_subcat) as n_subcat_sd

      , sum(qty_sales) as qty_sales_total
      , mean(qty_sales) as qty_sales_media
      , max(qty_sales) as qty_sales_max
      , min(qty_sales) as qty_sales_min
      , STDDEV_SAMP(qty_sales) as qty_sales_sd

      , sum(qty_returns) as qty_returns_total
      , mean(qty_returns) as qty_returns_media
      , max(qty_returns) as qty_returns_max
      , min(qty_returns) as qty_returns_min

      , mean(preco_min_sales) as preco_min_sales_md
      , max(preco_min_sales) as preco_min_sales_max
      , min(preco_min_sales) as preco_min_sales_min
      , STDDEV_SAMP(preco_min_sales) as preco_min_sales_sd

      , mean(preco_medio_sales) as preco_medio_sales_md
      , max(preco_medio_sales) as preco_medio_sales_max
      , min(preco_medio_sales) as preco_medio_sales_min
      , STDDEV_SAMP(preco_medio_sales) as preco_medio_sales_sd

      , mean(preco_sd_sales) as preco_sd_sales_md
      , max(preco_sd_sales) as preco_sd_sales_max
      , min(preco_sd_sales) as preco_sd_sales_min
      , STDDEV_SAMP(preco_sd_sales) as preco_sd_sales_sd

      , mean(preco_max_returns) as preco_max_returns_md
      , max(preco_max_returns) as preco_max_returns_max
      , min(preco_max_returns) as preco_max_returns_min
      , STDDEV_SAMP(preco_max_returns) as preco_max_returns_sd

      , mean(preco_min_returns) as preco_min_returns_md
      , max(preco_min_returns) as preco_min_returns_max
      , min(preco_min_returns) as preco_min_returns_min
      , STDDEV_SAMP(preco_min_returns) as preco_min_returns_sd

      , mean(preco_medio_returns) as preco_medio_returns_md
      , max(preco_medio_returns) as preco_medio_returns_max
      , min(preco_medio_returns) as preco_medio_returns_min
      , STDDEV_SAMP(preco_medio_returns) as preco_medio_returns_sd

      , mean(preco_sd_returns) as preco_sd_returns_md
      , max(preco_sd_returns) as preco_sd_returns_max
      , min(preco_sd_returns) as preco_sd_returns_min
      , STDDEV_SAMP(preco_sd_returns) as preco_sd_returns_sd

      , STDDEV_SAMP(qty_returns) as qty_returns_sd

      , sum(flg_app) as n_trx_app
      , sum(flg_sales) as n_trx_sales
      , sum(flg_return) as n_trx_returns
      , count(distinct s.log_audit) as n_trx_total
      , sum(flg_app)/count(distinct s.log_audit) AS perc_transactions_app

      , SUM(sku_sales_direct)/(SUM(vb_sales)+SUM(sku_sales_direct)) as sales_direct_disc
      , SUM(sku_sales_deferred)/SUM(vb_sales) as sales_deferred_disc

      , SUM(sku_returns_direct)/(SUM(vb_returns)+SUM(sku_returns_direct))  as returns_direct_disc
      , SUM(sku_returns_deferred)/SUM(vb_returns) as returns_deferred_disc
 from note5_spk n
 LEFT JOIN dados_core s
   ON (n.id_cliente = s.CST_CODE AND (n.ym_prev < s.month_key AND n.month_prev >= s.month_key))
 group by n.id_cliente
      , n.data_inq
      , n.month_prev
      , n.ym_prev
")
```

Figure C.4: Extraction and variable construction of some the variables of Purchase Behaviour - Part II