UNIVERSIDADE CATÓLICA PORTUGUESA

Building a predictive lead scoring model for contact prioritization

The case of HUUB

Rita Mafalda Magalhães Pereira

Católica Porto Business School
2021

UNIVERSIDADE CATÓLICA PORTUGUESA

Building a predictive lead scoring model for contact prioritization

The case of HUUB

Trabalho Final na modalidade de Relatório de Estágio
apresentado à Universidade Católica Portuguesa
para obtenção do grau de mestre em Gestão

por

Rita Mafalda Magalhães Pereira

sob orientação de
Professor Doutor António Andrade

Católica Porto Business School
Maio de 2021

Acknowledgements

This dissertation represents the final accomplishment of my Masters' degree path. Hence, this section is dedicated to acknowledging the people who have contributed to this project in some way.

Firstly, I would like to thank HUUB's team for the immense support and constant share of knowledge throughout this journey.

Secondly, I would like to express my gratitude to my supervisor, Professor António Andrade, who has guided and supported me during these months, providing me with great feedback such that I could further improve my work.

Lastly, I want to express my deepest appreciation to my family, boyfriend and friends, who have encouraged me and kept me optimistic on this journey.

Abstract

In the last decades, machine learning has become quite popular for solving business problems, as it often delivers high-quality and efficient solutions. Moreover, the amount of data collected by companies has grown substantially, which has contributed to this trend.

Companies do not have enough resources to contact every lead, so contact prioritization is essential. Lead scoring supports this task, by assigning a value to each lead based on his actions or characteristics. Even though it is expected that lead scoring contributes to higher conversion rates, there is still very few literature on how to use machine learning to automate this process.

This dissertation shows how to combine historical data from Customer Relationship Management platforms and supervised learning to develop a lead scoring model for companies. The approach followed is based on the CRISP-DM method, where several tools were used, such as HubSpot, Microsoft Power BI and RStudio. The classification model proposed is a decision tree that predicts the leads' conversion outcome (Won or Postpone), developed using the CART algorithm and data from a logistics company – HUUB.

The main findings of this project conclude that machine learning can be used to develop a lead scoring model to perform contact prioritization. However, there are several factors, especially data-related, that should be taken into consideration, since they may impact the model's performance.

Lastly, a suggestion for future research is to develop an experiment to compare the results of manual and automated lead scoring, to assess if machine learning actually provides a superior alternative to the manual approach.

Keywords: Marketing automation; Lead scoring; Machine learning; Contact prioritization.

Resumo

Nas últimas décadas, o *machine learning* tornou-se bastante popular para resolver problemas organizacionais, já que tende a produzir soluções eficientes e de alta qualidade. Adicionalmente, a quantidade de dados colecionados pelas empresas cresceu substancialmente, o que contribuiu para esta tendência.

As empresas não têm recursos suficientes para contactar todos os *leads*, pelo que é essencial priorizá-los. O *lead scoring* apoia esta tarefa, ao atribuir um valor para cada *lead* baseado nas suas ações ou características. Embora seja expectável que o *lead scoring* contribua para melhores taxas de conversão, ainda é escassa a literatura acerca da automatização deste processo através do *machine learning*.

Esta dissertação expõe como combinar *supervised learning* e dados históricos de sistemas de *Customer Relationship Management* para desenvolver um modelo de *lead scoring* para empresas. A abordagem baseia-se no método CRISP-DM, onde diversas ferramentas foram usadas, nomeadamente o *HubSpot*, o *Microsoft Power BI* e o *RStudio.* O modelo de classificação proposto é uma árvore de decisão que prevê o desfecho de conversão dos *leads*, desenvolvido com o algoritmo CART e dados de uma empresa de logística – a HUUB.

As principais descobertas deste projeto concluem que é viável utilizar o *machine learning* para desenvolver um modelo de *lead scoring* para priorizar os contactos. Contudo, há fatores que devem ser tidos em conta, especialmente relacionados com os dados, já que podem impactar o desempenho do modelo.

Por fim, sugere-se para pesquisa futura o desenvolvimento de um estudo experimental que compare os resultados do *lead scoring* automatizado e manual, de forma a avaliar se o *machine learning* é de facto a melhor alternativa.

Palavras-chave: Automatização do *Marketing*; *Lead scoring*; *Machine learning*; Priorização de contactos.

List of Abbreviations

AUC – Area Under the Curve

B2B – Business-to-Business

B2C – Business-to-Consumer

CEO – Chief Executive Officer

CRISP-DM – Cross-Industry Standard Process for Data Mining

CRM – Customer Relationship Management

DAG – Directed Acyclic Graph

DQM – Direct Qualification Model

FFM – Full Funnel Modeling

MQL – Marketing Qualified Lead

SQL – Sales Qualified Lead

Table of Contents

Acknowledgements ..... *iv*

Abstract ..... *vi*

Resumo ..... *viii*

List of Abbreviations ..... *x*

List of Figures ..... *xvi*

List of Tables ..... *xviii*

1. Introduction *21*

1.1 Research Question and Goals **21**

1.2 Research Methodology **23**

1.3 Structure of the Dissertation **24**

2. Literature Review: Lead Management *26*

2.1 Leads **26**

2.2 Lead Management and Sales Funnel **27**

2.3 Lead Scoring **30**

2.4 Research Gap **32**

3. Literature Review: Models and Algorithms for Lead Scoring *34*

3.1 Bayesian Network **35**

3.1.1 Bayesian Networks 35

3.1.2 Contextualization of the study and data used 37

3.1.3 Results 40

3.2 Gradient Boosting Trees **40**

3.2.1 Data used 41

3.2.2 Direct Qualification Model 41

3.2.3 Full Funnel Modeling 43

3.2.4 Results 44

3.3 Logistic Regression, Decision Trees, Random Forests and Neural Networks **45**

3.3.1 Contextualization of the study and data used 46

3.3.2 Logistic Regression 47

3.3.3 Decision Trees 48

3.3.4 Random Forests 49
3.3.5 Neural Networks 50
3.3.6 Results 51
3.4 Models' Comparison 52
4. Research Methodology 53
4.1 Goals of the research 53
4.2 Contextualization of the research – HUUB 54
4.3 Method – CRISP-DM 56
5. Case Study – HUUB 60
5.1 Business Understanding 60
5.1.1 HUUB's Inside Sales 2.0 Framework 61
5.1.2 HUUB's Need for a Lead Scoring Model 63
5.2 Data Understanding and Data Preparation 64
5.2.1 HubSpot's Organization 64
5.2.2 HubSpot Deal Properties 67
5.2.3 Feature Selection 69
5.2.4 Preliminary Data Analysis and Data Cleaning 71
5.2.5 Data Sampling and Attribute Construction 88
5.3 Modeling 90
5.3.1 Supervised and Unsupervised Learning 90
5.3.2 Decision Trees 91
5.3.3 Model Construction 93
5.3.4 Generated Tree Interpretation 97
5.3.5 Model Assessment 99
5.4 Evaluation 104
5.4.1 Evaluation of the Results 104
5.4.2 Process Review 106
5.4.3 Next Steps 107
5.5 Deployment 108
6. Conclusion 110
6.1 Final Considerations 110
6.2 Main Contributions 112
6.3 Future Work 113

Bibliographic References *114*

Attachments *121*

Attachment I – Manual lead scoring matrix **121**

Attachment II – Lead scoring automation. **121**

Appendixes *122*

Appendix I – Methodology overview. **122**

Appendix II – HubSpot deal properties. **123**

Appendix III – Data warehouse star schema. **125**

Appendix IV – Data import and understanding in R **125**

Appendix V – Stratified random sampling in R. **126**

Appendix VI – Balancing the training sample with three different methods in R. **126**

List of Figures

Figure 1 *– Sales funnel.* *28*
Figure 2 *– Stages of the sales process.* *30*
Figure 3 *– Bayes' theorem.* *35*
Figure 4 *– DAG.* *36*
Figure 5 *– Distribution of joint probability over U.* *37*
Figure 6 *– Bayesian network structure.* *39*
Figure 7 *– Logistic regression.* *48*
Figure 8 *– Decision tree.* *48*
Figure 9 *– CRISP-DM method.* *57*
Figure 10 *– HUUB's Inside Sales 2.0 framework.* *61*
Figure 11 *– HUUB's digital scoring.* *63*
Figure 12 *– HubSpot's Inside Sales pipeline layout.* *65*
Figure 13 *– HubSpot's Conversion pipeline layout.* *65*
Figure 14 *– Inside Sales pipeline.* *66*
Figure 15 *– Conversion pipeline.* *67*
Figure 16 *– Sales data report in Microsoft Power BI.* *87*
Figure 17 *– Summary of the final dataset.* *89*
Figure 18 *– Original training sample.* *94*
Figure 19 *– Training sample after over and under-sampling.* *95*
Figure 20 *– R code for building the decision tree model.* *95*
Figure 21 *– Optimal complexity parameter analysis.* *97*
Figure 22 *– R code and output for the decision tree plot.* *98*
Figure 23 *– Confusion matrix and performance metrics for the decision tree model.* *100*

List of Tables

Table 1 *– Performance metrics for Bayesian network.* *40*
Table 2 *– Companies A and B's details.* *41*
Table 3 *– Training set details for companies A and B.* *41*
Table 4 *- AUC for DQM.* *45*
Table 5 *- AUC for FFM.* *45*
Table 6 *- Sources of data used.* *46*
Table 7 *- Model performance comparison.* *52*
Table 8 *– Absolute and relative frequencies of deals for variable Brand Segment.* *72*
Table 9 *– Absolute and relative frequencies of deals won for variable Brand Segment.* *72*
Table 10 *– Absolute and relative frequencies of deals for variable E-commerce Platform.* *73*
Table 11 *– Absolute and relative frequencies of deals won for variable E-commerce Platform.* *73*
Table 12 *– Absolute and relative frequencies of deals for variable ECO.* *74*
Table 13 *– Absolute and relative frequencies of deals won for variable ECO.* *74*
Table 14 *– Absolute and relative frequencies of deals for variable Country.* *75*
Table 15 *- Absolute and relative frequencies of deals won for variable Country.* *76*
Table 16 *- Absolute and relative frequencies of deals for variable Instagram Followers.* *77*
Table 17 *- Absolute and relative frequencies of deals won for variable Instagram Followers.* *77*
Table 18 *– Absolute and relative frequencies of deals for variable Instagram Engagement.* *77*
Table 19 *– Absolute and relative frequencies of deals won for variable Instagram Engagement.* *78*
Table 20 *– Absolute frequencies of deals for variable Instagram Influencers.* *78*
Table 21 *– Absolute frequencies of deals won for variable Instagram Influencers.* *79*
Table 22 *– Absolute frequencies of deals for variable Product's Category.* *79*
Table 23 *– Absolute frequencies of deals won for variable Product's Category.* *80*
Table 24 *– Absolute frequencies of deals for variable Sales Channels.* *80*
Table 25 *– Absolute frequencies of deals won for variable Sales Channels.* *80*
Table 26 *– Absolute frequencies of deals for variable Brand Type.* *81*
Table 27 *– Absolute frequencies of deals won for variable Brand Type.* *81*
Table 28 *– Absolute frequencies of deals for variable Production.* *83*
Table 29 *– Absolute frequencies of deals won for variable Production.* *83*
Table 30 *– Descriptive statistic measures for variable Number of Contacts.* *84*
Table 31 *– Descriptive statistic measures of deals won for variable Number of Contacts.* *85*
Table 32 *– Absolute and relative frequencies for variable Number of Contacts.* *85*
Table 33 *– Absolute and relative frequencies of deals won for variable Number of Contacts.* *85*
Table 34 *– Descriptive statistic measures and missing values for variable Aesthetic Rank.* *86*

Table 35 *– Descriptive statistic measures and missing values of deals won for variable Aesthetic Rank. 86*
Table 36 *– Absolute and relative frequencies for variable Aesthetic Rank. 86*
Table 37 *– Absolute and relative frequencies of deals won for variable Aesthetic Rank. 86*
Table 38 *– Performance metrics for different sampling methods. 94*
Table 39 *– Rules behind the decision tree algorithm. 99*
Table 40 *– Comparison of the models' performance. 103*

First Chapter

1. Introduction

"*Modern technology has become a total phenomenon for civilization, the defining force of a new social order in which efficiency is no longer an option but a necessity imposed on all human activity.*" (Jacques Ellul)

This dissertation was developed during an internship at the tech start-up HUUB, which proposed the development of a lead scoring model for its potential clients leveraged by machine learning. The goal is for the model's results to allow the Sales team to adequately perform contact prioritization, which may lead to more efficiency and better results of the sales performance indicators, such as higher conversion rates.

HUUB is a logistics company for independent fashion brands. The start-up manages the supply chain of its clients from end-to-end and gives them visibility over their business through an integrated digital platform, Spoke. Currently, the company is in its expansion phase; hence, a big focus is on technology and automation to scale up the business.

This internship took place in the Data Insights & Analytics team. However, the project required collaboration with other teams, namely Sales and Data Engineering, which provided great contributions for this work.

1.1 Research Question and Goals

The research question this dissertation proposes to answer can be formalized as: *How can a Business-to-Business (B2B) company automate the process of lead scoring?*, which relates to three main goals:

- Identify relevant trends and insights from past converted leads.

- Identify which attributes are relevant to predict lead conversion at an early stage of the sales process.
- Develop a lead scoring model that allows the Sales team to adequately perform contact prioritization.

There is a great motivation to study this theme, as it is quite relevant for several reasons. Firstly, companies have limited resources, such as time and money. Hence, efficiently managing them is important for companies' success, which is why organizations are constantly trying to improve the efficiency of their processes, to achieve better outcomes with less resources. This applies to many processes, such as acquiring new customers – generally, companies do not have enough time and money to contact every lead, which reinforces the relevance of a lead scoring model that helps perform contact prioritization.

Secondly, there is a research gap in the literature, identified by Nygård and Mezei (2020), surrounding this theme – there are still very few studies concerning the use of machine learning models for automating the lead scoring process. Manual lead scoring is the dominant approach used in practice; however, it has several disadvantages and is perceived as inefficient. This dissertation diminishes this gap, by presenting a practical study of how a company can use machine learning to develop a lead scoring model.

Thirdly, and focusing on the company in which this project took place, HUUB's Sales team contacts 60 brands each week through campaigns, which consist of e-mails and cold calls. For that, Sales associates search for brands on the Customer Relationship Management (CRM) platform and pick 60 to enter the campaigns. Noticeably, this process could be more efficient with a lead scoring model that predicts the conversion outcome of each brand (Won or Postpone) based on its characteristics, which are gathered during profiling. Based on the model's predictions, the team can prioritize the brands that obtain a positive prediction (Won).

1.2 Research Methodology

The practical component of this dissertation, the case study, follows a popular data mining method, CRIPSP-DM, which provides a set of guidelines that conduct the researcher throughout the knowledge discovery process.

Nowadays, many companies gather data, which can be used for training machine learning models, resulting in more automated mechanisms inspired by trends detected in real data. The lead scoring model developed in this project is a classification model that predicts the conversion outcome of each lead. To build the model, historical data from HUUB's CRM platform was extracted and used as training data for a predictive algorithm – the CART algorithm.

The case study starts with a business understanding phase, in which the problem and main tasks of the project are presented. During this stage, there was a meeting with the Sales team to gather information on the company's sales process and to identify which needs the lead scoring model should satisfy.

In the second and third phases, the data is extracted, analyzed and prepared for the final dataset. This starts with a study of HUUB's CRM platform, HubSpot, and the information fields it stores to characterize the leads. Then, feature selection is performed, along with a preliminary analysis of the data, to extract insights, detect data quality issues and perform data cleaning. Lastly, data sampling and attribute construction are performed to obtain the final dataset.

In the fourth stage, the dataset is imported into RStudio and split into two samples, one for training the model and another to test its performance, using a stratified random sampling. Then, the chosen method is a decision tree, due to its interpretability to the user of the model's results – the Sales team. Since the dataset is imbalanced, three random sampling methods are tested to compare the models' performance. Then, the tree is built using the CART algorithm and

pruned with an optimal complexity parameter. Lastly, the model is assessed using the confusion matrix-based performance metrics.

The last stages, evaluation and deployment, evaluate the project's results, review the process, identify future work that could be valuable to the project and present the maintenance and monitoring strategy of the data mining results.

1.3 Structure of the Dissertation

This dissertation is divided into six chapters. The current chapter, Chapter 1, presents an overview of the research question and goals, the methodology followed and the structure of this paper. Then, Chapter 2 acts as a theoretical framework for this dissertation, as it presents the main concepts surrounding lead management, as well as the research gap identified in the literature.

Chapter 3 is a literature review of models and algorithms that have been proposed in the literature to automate the lead scoring process in companies. It reviews several studies that used historical data and Artificial Intelligence to build lead scoring models, presents the data used and the results obtained. It concludes with a comparison of the models presented.

Chapter 4 focuses on the research methodology of this dissertation. It discusses the main goals of this project and the research question, presents HUUB's business in more detail, and describes the six phases of the CRISP-DM method.

Chapter 5 presents the practical component of this project – the case study. This chapter is organized according to the CRISP-DM's phases: it starts with the business understanding of the problem and current situation, followed by the data understanding and preparation phase, in which the data is analyzed and the final dataset is prepared; then, there is the modeling phase, where the lead scoring model is built using the CART algorithm and assessed through common performance metrics; finally, there are the evaluation and deployment

stages, which review the project and its results, identify future work and present the maintenance and monitoring strategy of the data mining results.

Lastly, Chapter 6 is the conclusion of this dissertation. It summarizes the project and its main contributions, reinforces the conclusions obtained, and suggests future work for this thematic.

Second Chapter

2. Literature Review: Lead Management

This chapter presents a literature review regarding leads, and acts as a theoretical framework for this dissertation. Some fundamental concepts will be presented, as well as the main research gap found in the literature.

In terms of macrostructure, this literature review follows a thematic approach. Thus, this chapter is divided into four sections: Subchapter 2.1, which introduces the concept of leads, Subchapter 2.2, which presents the lead generation process and the sales funnel, Subchapter 2.3, where lead scoring is exposed, and Subchapter 2.4, where the research gap is summarized.

2.1 Leads

A lead can be defined as "a contact generated by a marketing campaign to existing or potential new customers that express interest in the company's offering or inquire into products or services" (Todor, 2016, p.90). However, the concept of leads can slightly fluctuate according to different authors and practitioners.

The term lead is frequently used in the Marketing and Sales departments, which focus on moving leads through the sales funnel, with the goal of converting them into customers. However, this is not an easy nor fast process, especially when many leads are generated, which reinforces the relevance of the research question presented in this dissertation.

2.2 Lead Management and Sales Funnel

The process of developing prospects into customers is called lead management (Bradford et al., 2016). The first step in this process is lead generation, which is a critical phase (Peterson et al., 2011). Lead generation has a great impact over companies' profitability, however it is consistently considered as a major challenge for organizations (Niemi, 2017).

Lead generation intends to attract potential customers or to obtain their details (Doyle, 2016). Throughout this process, companies use marketing tactics to create consumer interest in their products or services (Todor, 2016). There are two types of marketing strategies: inbound and outbound. Although marketing approaches have evolved with the digital evolution, both strategies are still relevant.

Outbound marketing is the traditional product or service promotion, such as television and radio advertising, that interrupts someone's course of action (Dakouan et al., 2019). The audience is forced to deal with the message, even though it was not searching for it. The main goal is to attract a large audience without previously studying the target (Dakouan et al., 2019).

According to Dakouan et al. (2019), outbound marketing is no longer effective enough, which is why a more recent approach, inbound marketing, has become so important. Inbound marketing occurs when the consumers search for information regarding the company, such as visiting its website. This strategy pulls the public towards the business, and is usually based on creative content. Since it is a more authentic approach, its use has been growing throughout the years. Nonetheless, Dakouan et al. (2019) consider that both strategies are complementary and should be used together whenever it is possible, since both have advantages.

Another important Marketing concept is the sales funnel, which represents the journey of a customer from awareness through purchasing. This funnel

represents "the pool of leads or potential customers in the various stages of the sales process" (Gao et al., 2019, p.1), and is used to manage the flow of leads throughout the phases (Bradford et al., 2016). When analyzing a sales funnel (Figure 1), it is visible that the first stages, on top, represent the biggest pools of leads, and the volume keeps decreasing as we move down the funnel.

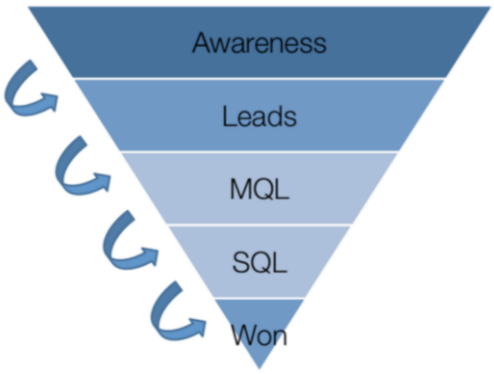


Figure 1 *– Sales funnel.*
Source: Duncan & Elkan (2015, p.1751)

Accordingly, in the first stages, more automation is required to target a wide audience. Hence, for the first interactions, companies tend to use automated strategies, such as mass e-mails. Then, as the leads flow towards the end of the funnel, more human interaction is required to ensure successful closures. Nonetheless, marketing automation can operate on all stages of the funnel, as it contributes to efficiency gains and higher return on marketing investments (Lindahl, 2017).

Regarding the types of prospect customers, leads are the initial prospects that have not been evaluated; then, MQLs (Figure 1) refer to Marketing Qualified Leads – leads that were qualified by the Marketing team according to their behavior and demographics, and passed along to Sales (Duncan & Elkan, 2015). After that, there are SQLs, which refer to Sales Qualified Leads – leads that passed Sales' criteria, such as having the budget to complete a purchase (Duncan & Elkan, 2015). Most SQLs are pursued by Sales, and can result in a

closed-won, if the ending is successful, or a closed-lost, otherwise (Duncan & Elkan, 2015).

According to Rosenbröijer (2014), the discipline of Marketing has been through a major perspective change in the last decades. In the past, marketeers used to focus on the product, however, nowadays, their focus is more customer centric. The traditional transactional approach has subsided and shifted towards a customer relationship approach: the customer is now seen as the central component of the business, and companies prioritize this perspective when making decisions. Hence, CRM – the devices, operations and support to optimize the quality of the customer relationship (Benhaddou & Leray, 2017) – has become a very relevant topic.

There must be a great bond between Sales and Marketing to generate qualified leads and maximize sales. Studies show that companies where Sales and Marketing have a strong collaboration, achieve better sales effectiveness (Patterson, 2007). Past research showed a lack of alignment between these teams, partly due to the dominant transactional approach (Patterson, 2007). Patterson (2007) suggests that switching towards a more customer centric strategy, which is what has been occurring (Rosenbröijer, 2014), contributes to a better alignment between these teams.

Figure 2 shows that, between the processes of lead generation and lead scoring, there is another task – lead nurturing. Conceptually, it can be defined as the process of building and nurturing relationships through on-going and valuable communication, whether the prospect becomes a customer or not (Niemi, 2017).

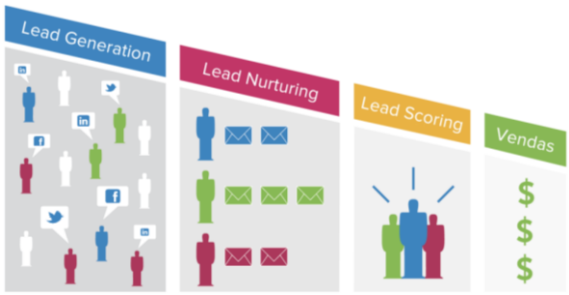


Figure 2 *– Stages of the sales process.*
Source: Giacomele (2015)

A study conducted by McGlaughlin et al. (2012) showed that organizations that nurture their leads, experience, on average, a lift of 45% of lead generation return on investment over those that do not practice lead nurturing. Generating leads and passing them along to Sales is no longer enough, since the quality of the leads is determined by finding "the right buyers at the right time" (McGlaughlin et al., 2012, p.6). If, on one hand, lead generation strategies contribute to finding the right buyers, on the other hand, lead scoring and nurturing are still crucial to identify the right time to pass them to Sales.

Even though companies are increasingly trying to become data driven, the processes of lead generation and lead nurturing are still largely based on guesswork, which often leads to a poor allocation of resources (Niemi, 2017).

2.3 Lead Scoring

Once the lead generation process has been completed, businesses face another challenge. If many leads are generated, companies need to figure out which ones are most likely to become customers. This is fundamental for efficiency, because if these leads are prioritized and contacted first, Sales will save a lot of time on leads that are not ready to purchase yet.

The relevant leads are those that have the need and the budget now and lead scoring is crucial because it helps identifying those ready for sale leads (Marion, 2016). Furthermore, data shows that, on average, only 6% of MQLs

convert into customers, and lead scoring may improve these results (Duncan & Elkan, 2015).

Lead scoring is a subtask of CRM, that consists of calculating and assigning a value to each lead (Benhaddou & Leray, 2017), based on its actions and attributes (Salesforce, 2014). Its main goal is to prioritize the leads, respond to them adequately, and increase the conversion rate (Cox, 2019). Lead scoring allows Sales representatives to identify leads that are ready for immediate follow-up (Salesforce, 2014), contributing to lower costs per lead and higher conversion rates (Lindahl, 2017).

Lead scoring can be based on a wide range of features. According to Nygård and Mezei (2020), one of the tasks that most influences the quality of lead scoring is the selection of variables to use. The data is usually behavioral data, such as reactions to marketing campaigns and visits to the company's website, or demographic information. Furthermore, the data collected by companies can be classified as explicit data, if it is directly obtained from leads' input, or implicit, if it is obtained by data collected on the leads' actions (Nygård & Mezei, 2020).

It is important to distinguish between manual and automated lead scoring. Manual lead scoring is the dominant approach used in practice (Nygård & Mezei, 2020), even though many authors believe it has problems. The main issue presented by Marion (2016) is that attributing scores to leads should be handled by statisticians. However, this task is traditionally assigned to marketeers, who act based on their instincts and assumptions. Usually, manual lead scoring starts by developing a scoring matrix that includes information regarding lead qualification criteria (Attachment I). However, without support from statisticians, it is hard for marketeers to know how to weight the relative importance of each signal to predict a buyer's purchase probability. Furthermore, manual lead scoring requires continuous testing and measuring, in order to incorporate improvements into the process, which can be extremely

time consuming and, consequently, inefficient. Moreover, traditional lead scoring is unable to handle the amount of data needed for the scoring to reach statistical significance and it focuses on data captured by marketeers, leading many predictive signals of buying behavior to be missed, such as current technology usage (Marion, 2016).

Duncan and Elkan (2015) reinforce some disadvantages of manual lead scoring, namely that the scores are hand-selected by Marketing or Sales teams, and it heavily relies on behavioral data, which does not allow the company to discover leads quickly, as the data has to be collected over a period of time.

Marion (2016) believes there are better ways to generate qualified leads then through manual lead scoring, such as using predictive lead scoring platforms that use machine learning to identify conversion signals and rank the leads accordingly. Based on historical data, they predict the likelihood of conversion of leads with similar criteria (Marion, 2016). Nygård and Mezei (2020) also state that nowadays, many companies gather huge amounts of data, which, along with predictive analytics, can be the source of competitive advantage for companies, since past data can be used as training data for many algorithms. Currently, most sales agents focus their time on leads without any predictive data, which results in fewer conversions of leads into customers (Venkatesan et al., 2018).

2.4 Research Gap

Nygård and Mezei (2020) have recently identified a research gap regarding this topic, namely, that there is still very few literature concerning the use of machine learning models for automating the lead scoring process.

Actually, the authors' recent study (Nygård & Mezei, 2020) is a contribution to surpass this gap, as it consists in an experiment where different algorithms are applied to real data to assess their performance in automating lead scoring.

Targeting this research gap is relevant because companies should start considering automated lead scoring as an alternative to the still broadly used manual lead scoring, to become more efficient (Nygård & Mezei, 2020). The practical relevance of this problem is huge, however literature is not yet giving it the deserved attention. The aim of this dissertation is to provide a contribution that helps to tackle this gap.

Third Chapter

3. Literature Review: Models and Algorithms for Lead Scoring

This chapter will present some models and machine learning algorithms that have been proposed in the literature to automate the lead scoring process in companies.

Since the decade of 1970, Artificial Intelligence techniques have gained popularity in solving real life problems (Gama et al., 2017). Although, in the past, this process often required interviews with experts in the field of study, for knowledge to be acquired and coded through rules using software. This process had major limitations, namely being highly subjective (Gama et al., 2017). In the last decades, however, problems grew in complexity, and the amount of data collected got more substantial. This emphasized the need for more autonomous tools, that required less human intervention. To accomplish this, these techniques would have to learn on their own from past experience.

The models that will be presented took advantage of historical data and of Artificial Intelligence to automate the lead scoring process in companies (Attachment II). Subchapter 3.1 presents Bayesian networks, Subchapter 3.2 focuses on gradient boosting trees and Subchapter 3.3 portrays four methods: logistic regression, decision trees, random forests and neural networks. In each subchapter, the algorithm is introduced, along with a contextualization of the study and data used, and a review of the results obtained. Lastly, Subchapter 3.4 performs a comparison of the aforementioned models.

3.1 Bayesian Network

Benhaddou and Leray (2017) attempted a Bayesian network for building a lead scoring model for a B2B company, with a small amount of data. The authors attempted this method since some CRM tools use probabilistic graphical models for other tasks (Benhaddou & Leray, 2017).

3.1.1 Bayesian Networks

Benhaddou and Leray (2017) highlight some of the advantages of Bayesian networks, such as their ability to handle uncertainty. Moreover, an important feature of these models is that they can be built from the knowledge of experts.

Bayesian networks (Pearl, 1988) refer to a probabilistic method based on the Bayes' theorem (Figure 3). This theorem assumes that the probability of an event A occurring given the occurrence of an event B does not only depend on the relation between A and B, but also on the probability of observing A independently of observing B (Duda et al., 2001). The probability of an event occurring can be estimated through the frequency in which it is observed (Gama et al., 2017). In classification problems, A represents the class and B the observed values of the features of an observation.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Figure 3 *– Bayes' theorem.*
Source: Prog (2011)

A Bayesian network is as "an annotated directed acyclic graph that encodes a joint probability distribution over a set of random variables" (Benhaddou & Leray, 2017, p.251). It relies on the concept of conditional independence between variables – considering the variables X, Y and Z, X is conditionally independent of Y given Z if $P(X|Y,Z)=P(X|Z)$ (Gama et al., 2017).

Considering U as a set of random variables: $U=(X_1, X_2, \ldots, X_n)$, a Bayesian network for U can be written as $B=(G,\Theta)$ (Benhaddou & Leray, 2017). The first component, which encodes the structure of the G network, is a directed acyclic graph (DAG) (Gama et al., 2017). The vertices or nodes of the DAG represent the random variables, whilst the edges represent the direct dependencies between them (Benhaddou & Leray, 2017) (Figure 4).

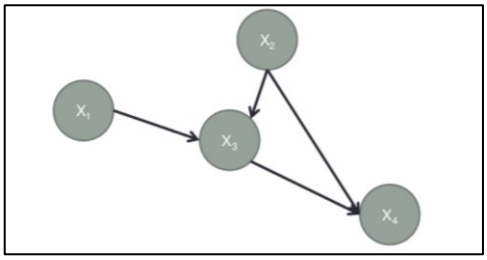


Figure 4 *– DAG.*
Source: Silva (2015)

The second component, Θ, symbolizes the tables of conditional probability (Gama et al., 2017), hence, representing the parameters that quantify the network (Benhaddou & Leray, 2017). There is a parameter $\Theta_{xi|\Pi xi} = P_B(xi|\Pi xi)$ for each possible value xi of Xi and Πxi of ΠXi, where ΠXi refers to the parents of Xi in G (Benhaddou & Leray, 2017). Conceptually, the random variables that influence a variable X_i are the parents of X_i (Gama et al., 2017).

In summary, the DAG represents the qualitative model of the network, whilst the conditional probability tables act as the quantitative model (Gama et al., 2017).

Furthermore, the DAG satisfies the Markov condition, which assumes that each node is independent of all its non-descendants, given its parents in G (Gama et al., 2017). Given this condition, the distribution of joint probability over U can be represented in the factorized form (Figure 5).

$$P_B(X_1, ..., X_n) = \prod_{i=1}^{n} P_B(X_i|\Pi X_i) = \prod_{i=1}^{n} \Theta_{X_i|\Pi_{X_i}}$$

Figure 5 *– Distribution of joint probability over U.*
Source: Benhaddou & Leray (2017, p.252)

The complexity of a Bayesian network corresponds to the number of parameters used to define the conditional probability distributions (Benhaddou & Leray, 2017). The complexity is high if the variables have many parents: given a node and its n parents, the size of the tables of conditional probability grows exponentially with n (Fenton & Neil, 2019).

To reduce this complexity, some generic Bayesian network structures have been identified, such as the Noisy-OR model (Fenton & Neil, 2019). In this case, Benhaddou and Leray (2017) used an extension of the Noisy-OR model, called Leaky Noisy-OR, which is a popular method used to simplify the conditional probability tables in Bayesian networks that involve boolean variables, since it only requires (n+1) parameters to encode the tables of conditional probability (Fenton & Neil, 2019).

3.1.2 Contextualization of the study and data used

In this project, Benhaddou and Leray (2017) worked with Uneek, a company that developed a CRM tool called Kosmopolead. To generate leads that may be interested in Kosmopolead, Uneek's commercials search for online news regarding companies that may want to purchase that tool, and then contact them. The lead scoring model built by the authors imitates this work, by computing a score based on if the company present in the news is interesting for Uneek and should be contacted or not.

The main source of data used in the experiment was online news about different companies. Since the data available was not enough to define the structure, the authors used knowledge-based methods.

The factors (variables) used by Benhaddou and Leray (2017) are boolean. For each decision factor (DF_i) some keywords that model the experts' criteria were defined, such that if a keyword is present on the news text, that DF_i will be true, and false otherwise. For example, one of the factors used was "Reorganization", for which there were associated keywords such as "Moving" and "Fusion"; hence, if one of those words was present on the news, the DF_i "Reorganization" would be true. The target variable, "Kept", is a binary variable that expresses if a company should be contacted or not. The scoring for each lead was given by P(Kept=true|evidence).

There are two types of decision factors: mandatory and secondary. Mandatory decision factors (MF_i) have to be true for a company to be kept. "Headquarter Location" was the only MF_i defined, such that if a lead's headquarters is not the same as Uneek's, that MF_i is false and its scoring is zero. The secondary decision factors are the usual attributes included in lead scoring – the more positive they are, the higher the scoring will be; the secondary factors chosen were "Interest", "Activity Sector", "Company Type", "Growing Company" and "Reorganization".

Figure 6 presents the structure of the Bayesian network constructed by Benhaddou and Leray (2017). Heuristics were used to reduce the complexity of the conditional probability distributions between the final scoring and the decision factors. The authors executed a decomposition, using parent divorcing, to achieve a hierarchical structure; parent divorcing reduces the complexity of the model by adding intermediate decision factors (Benhaddou & Leray, 2017).

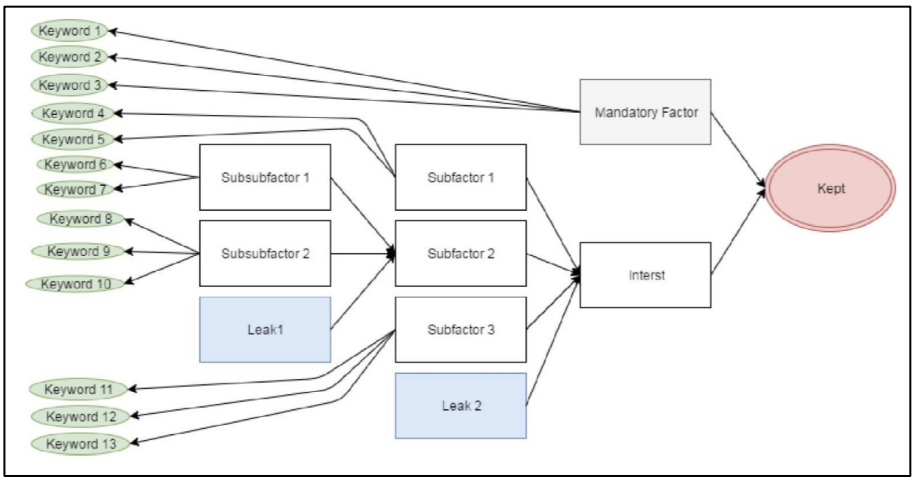


Figure 6 *– Bayesian network structure.*
Source: Benhaddou & Leray (2017, p.254)

The probability distribution of each keyword given the factor it is connected to was defined as:

- P(keyword = true | factor = true) = 0.8
- P(keyword = true | factor = false) = 0.3.

As previously mentioned, the authors used an approximation method to capture the preferences of the experts regarding the decision factors, where each conditional probability distribution association to a decision factor was modeled by a Leaky Noisy-OR model. Here, one factor has n parents and (n+1) parameters need to be identified, and each parameter defines the interest of the news when factor Xi is true and the others are false. To obtain these parameters, Benhaddou and Leray (2017) used an Analytic Hierarchy Process, which ranks the parent factors and finds their weights. Firstly, the experts were asked to state their relative importance for each pair of factors by using a scale. Then, the pairwise comparisons were encoded in a matrix, where each row and column represents a decision factor, and each cell represents the relative importance between a pair. The columns were then normalized, and the weight of each factor corresponded to the mean of the column. Lastly, the weights were re-calibrated, so that the probability of the best factors would not be overestimated (Benhaddou & Leray, 2017).

3.1.3 Results

Uneek provided Benhaddou and Leray (2017) with 17 positive and six negative news. The results obtained on the model's first trial are presented in Table 1.

Table 1 *– Performance metrics for Bayesian network.*

Precision	0.80
Recall	0.88
Accuracy	0.75

Source: Benhaddou & Leray (2017)

The first metric, precision, shows that 80% of the cases classified as positive were correctly identified (Hale, 2020b). Then, the recall (or sensitivity) shows that 88% of the actual positive cases were correctly identified (Hale, 2020b). Lastly, the accuracy, which is an overall performance measure, shows that 75% of the total predictions made by the model were correct.

These results reveal that the model exhibited a good performance in classifying the news. However, these were only the results of the first trial, since the model is still being updated as the amount of data available increases, so its performance could change in the future. Furthermore, the authors state in their study (Benhaddou & Leray, 2017) that the main issue of this trial was actually regarding the negative cases, since they only had a few examples.

3.2 Gradient Boosting Trees

Duncan and Elkan (2015) studied how probabilistic classifiers can be used to estimate the probability of a prospect becoming a customer and the expected revenue from a lead. The main goal is to adequately rank the leads, so that Sales and Marketing can prioritize their efforts accordingly.

Duncan and Elkan (2015) proposed two models: i) Direct Qualification Model and ii) Full Funnel Modeling.

3.2.1 Data used

CRM systems are very useful for machine learning as they store large amounts of historical data. The data used by Duncan and Elkan (2015), provided by the company Fliptop, was stored in CRM platforms and included demographic and behavioral leads' data.

The demographic features intend to estimate the fit of the individual with the product and specify information about the individual and the company he works for, namely: industry code, number of employees, market value, income, location, number of openings, technologies used in the company and job title. The behavioral data is represented by the number of times the individual has done a certain action, so it consists of numerical variables. This data is tracked by marketing automation software and includes website visits, marketing e-mails opened, unsubscribe forms filled out, among other events.

The historical data used was based on two companies, A and B, whose details are presented in Table 2. Furthermore, the information regarding the datasets that were used to train the models is presented in Table 3.

Table 2 *– Companies A and B's details.*

Company	Business	Employees	Annual Revenue
A	Software as service	≈ 200	≈ $20 Million
B	Software	> 500	≈ $100 Million

Source: Duncan & Elkan (2015)

Table 3 *– Training set details for companies A and B.*

Company	Unconverted Leads	Closed-lost Leads	Closed-won Leads	Demographic Features	Behavioral Features
A	5925	1320	1469	243	350
B	25904	956	1097	242	20

Source: Duncan & Elkan (2015)

3.2.2 Direct Qualification Model

The Direct Qualification Model (DQM) suggested by Duncan and Elkan (2015) uses a multi-class classifier, such that, depending on how far through the sales funnel a lead advances, a different class is assigned. This multi-class

classifier overcomes the cons of a traditional binary classifier for predicting if a lead will convert or not, namely that a lead that goes deeper in the sales funnel is more valuable that another that does not go so far. Hence, the leads can be classified into three different classes:

1. NoCON, for leads that never convert.
2. LOST, for leads that convert to opportunities that are lost in the end (closed-lost).
3. WON, for leads that convert to opportunities that have a successful ending (closed-won).

Regarding the data used for the DQM, for both companies A and B, 75% of the data was used for training purposes and 25% was used to test the performance of the model. To label the training data, the classes LOST and WON represent leads that have closed within the previous year, and the class NoCON includes leads that have not converted yet.

The algorithm used by Duncan and Elkan (2015) was the gradient boosting tree, which was implemented using the *scikit-learn* library in Python. This algorithm is included in ensemble learning and in boosting methods since it constructs a strong model through iteratively learning from an ensemble of weaker models (Kurama, 2020). The gradient boosting "minimizes a loss function by iteratively choosing a function that points towards the negative gradient" (Kurama, 2020). It has three main components: i) the loss function, ii) the weak learner and iii) the additive model. The loss function estimates how accurate the model is at making classifications by computing its residuals, the weak learners are usually decision trees and, lastly, the additive model is the sequence of adding the weak learners one at a time. After each iteration, the value of the loss function should decrease. This method generates "competitive, highly robust, interpretable procedures" for classification (Friedman, 2001, p.1189).

This algorithm was applied to the training set, to build the model, which was then put to test in predicting the outcome for the observations in the test set. The predictions obtained were probabilistic values.

For each lead x, the model computes three probabilities:

- $p_1(x)=P(NoCON|x)$.
- $p_2(x)= P(LOST|x)$.
- $p_3(x)= P(WON|x)$.

To convert these probabilities into a score, s(x), the authors used linear combinations of p_2 and p_3: $s(x) = \alpha p_2(x) + \beta p_3(x)$, where the possibilities $(\alpha,\beta)=(0,1)$ and $(\alpha,\beta)=(1,1)$ were considered, since they maximize the closed-won probability and the probability of conversion.

3.2.3 Full Funnel Modeling

As previously stated, a prospect passes through various stages as he moves along the sales funnel. For Duncan and Elkan (2015), the most relevant passages occur from lead to SQL and from SQL to closed-won. These transitions were represented by two models, and a third model was included for the last stage of the funnel, which quantifies the closed deal. In these models, x expresses the values of the leads' attributes:

- $P(SQL|x, lead)$.
- $P(won|x, SQL)$.
- $E(revenue|x, won)$.

The probability that a lead with the characteristics x will become a successful sale and his expected revenue are represented by:

- $P(won|x, lead) = P(SQL|x, lead) \cdot P(won|x, SQL)$.
- $E(revenue|x, lead) = P(won|x, lead) \cdot E(revenue|x, won)$.

The last formula allows the company to know the expected revenue from a prospect at the stage where he is a lead, which allows Sales to anticipate the budget that should be invested in pursuing that lead.

For the Full Funnel Modeling (FFM), a training set is required for each transition: one of leads for modeling the probability P(SQL|x, lead), one of opportunities for modeling P(won|x, SQL) and one of closed-won customers for modeling E(revenue|x, won). Regarding the choice of method, gradient boosting trees were also used in the FFM, but with a binary classifier.

For the FFM, the score of a lead can be calculated as:

- s(x) = P(won|x, lead), or
- s(x) = E(revenue|x, lead).

3.2.4 Results

Duncan and Elkan (2015) used two metrics to assess the DQM's performance:

- AUC_1 – the area under the Receiver Operating Characteristics curve for classification of NoCON leads versus classification of WON or LOST leads, which corresponds to ranking the leads with $(\alpha,\beta)=(1,1)$;
- AUC_2 – the area under the Receiver Operating Characteristics curve for classification of WON opportunities versus classification of NoCON or LOST leads, which corresponds to ranking the leads with $(\alpha,\beta)=(0,1)$.

To assess the FFM's performance, the AUC was computed for the classifier that predicts conversions and for the classifier that predicts closed-won.

As seen in Tables 4 and 5, both models achieved high AUC values, which denotes the good performance of the classifiers. Furthermore, Table 4 shows that, when both demographic and behavioral features are included, the performance of the model is better, so both types of data are relevant to classify the leads. It can be concluded that these models, when applied to data from real companies, are good performers at classifying lead conversion and at predicting future sales.

Table 4 *- AUC for DQM.*

Company	Features	AUC_1	AUC2
A	All	0.992	0.960
A	Only demographic	0.988	0.940
A	Only behavioral	0.927	0.867
B	All	0.956	0.969
B	Only demographic	0.928	0.964
B	Only behavioral	0.906	0.922

Source: Duncan & Elkan (2015)

Table 5 *- AUC for FFM.*

Company	Stage	AUC
A	Lead conversion	0.991
A	Closed-won	0.788
B	Lead conversion	0.952
B	Closed-won	0.912

Source: Duncan & Elkan (2015)

Furthermore, the DQM was put to test during three months at a company, and the end results showed that the company's conversion rate increased from 8% to 17% and the average lead qualifying time reduced from 20 to 7 days (Duncan & Elkan, 2015).

3.3 Logistic Regression, Decision Trees, Random Forests and Neural Networks

Nygård and Mezei (2020) tested four methods to evaluate the feasibility of machine learning algorithms for automating the process of lead scoring, namely: logistic regression, decision trees, random forests and neural networks. According to these authors, the behavioral data that many companies collect, along with predictive analytics, can be used to estimate the probability of a contact purchasing.

The algorithms attempted by Nygård and Mezei (2020) are supervised learning methods, since the models are trained using classified historical data. The training data includes the characteristics of previous leads and the observed conversion outcome.

Nygård and Mezei (2020) used RapidMiner software for this experiment: to pre-process the data, to build the models and to assess their performance.

3.3.1 Contextualization of the study and data used

Nygård and Mezei (2020) used real data from an international company, focusing on its business-to-consumer (B2C) leads in Finland. Table 6 presents the two main sources of data used.

Table 6 *- Sources of data used.*

Contact-level Data	**Activity Data**
Customer's name	Website visits
Customer's country	E-mails sent
Customer's location	E-mails opened
Source of the lead	E-mail clickthroughs
Whether the lead has purchased or not	Forms submitted
Etc.	Etc.

Source: Nygård & Mezei (2020)

From the contact-level dataset, the variables selected were:

1. An identifier – links the contacts in different datasets.
2. The location – the region of the lead.
3. The location of the marketing unit.
4. The dates created and modified – timestamps for lead related events.
5. The e-mail address.
6. The contacts' status and time – identifies customers and the moment of their purchase.

Regarding the activity data, it was transformed into a table with three columns: i) contact, ii) activity type and iii) time of activity, and went through a process of aggregation, where a count for various events was computed for each lead. For that, the end date for lead conversion had to be specified. For converted leads, the end date was defined as the time of the first purchase, so data collected on that customer after that was not included, and for unconverted leads, the lead's last activity was used as the end date.

The variables from the activity dataset that were included in the model were:

- Contact – the lead identifier.
- daysToEnd.max – the number of days between the first activity and the end date.
- daysToEnd.avg – the average number of days between all activities and the end date.
- Sum – the total number of activities.
- 1daySum – the number of activities within 1 day of the end date.
- 3daySum –the number of activities within 3 days of the end date.
- 1weekSum – the number of activities within 1 week of the end date.
- 2weekSum – the number of activities within 2 weeks of the end date.
- 4weekSum –the number of activities within 4 weeks of the end date.
- 10percentSum – the number of activities within 10 percent of the total time prior to the end date.
- 40percentSum – the number of activities within 40 percent of the total time prior to the end date.
- 80percentSum – the number of activities within 80 percent of the total time prior to the end date.

As previously stated, these authors tested four different algorithms, which are among the most used in CRM (Nygård & Mezei, 2020), and will be presented in the next subchapters.

3.3.2 Logistic Regression

A logistic regression is a technique used to model dichotomous outcome variables (Peng et al., 2002), that is widely employed for building classification methods (Ranganathan et al., 2019). It is included in generalized linear models (Taktak & Fisher, 2007).

The goal of a linear regression model is to classify an input vector $x = (x_1, x_2, \ldots, x_D)$ into one of two classes. It considers that the logarithm of the odds

of belonging to one class is a linear function of the features used for classification (Figure 7). In the expression below, p is the probability of belonging to one class, p/(1–p) is the odds ratio, and α, β_1, β_2, …, β_D are the estimated coefficients (Taktak & Fisher, 2007).

$$ln\,(p/1-p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_D x_D$$

Figure 7 *– Logistic regression.*
Source: Taktak & Fisher (2007)

The output consists of a number between 0 and 1 (Ranganathan et al., 2019), and 0.5 is considered the threshold to define if the class attributed to an observation is 0 or 1 – an instance with an output over 0.5 belongs to class 1. One assumption of logistic regression that does not always verify is the independence among variables; nonetheless, this method tends to achieve very good results in practice (DiGangi & Moore, 2013).

3.3.3 Decision Trees

A decision tree is a DAG where each node is either a split node or a leaf node (Gama et al., 2017). The split nodes are labeled with attributes chosen to split the data, whilst the leaf nodes are labeled with classifications. Moreover, the arcs are labeled with the possible values for the attributes. Figure 8 presents an example of a decision tree, where the green circles represent the split nodes and the blue circles represent the leaf nodes, which contain the final classifications.

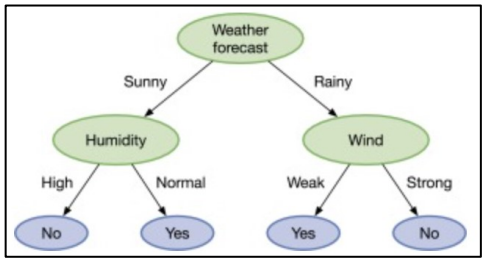


Figure 8 *– Decision tree.*

Source: Reinders et al. (2019)

Decision trees use a recursive partitioning algorithm for partitioning the observations of a dataset into different subsets. There are several algorithms for building decision trees, such as CART, ID3 and C4.5, which are the most popular (Bouchon-Meunier et al., 2006).

The classification tree obtained can be used for decision making, since it encodes rules for classification. Decision trees are very popular because their results are very clear for any decision maker, whilst other algorithms can be hard to understand. They have the advantage of offering intuitive explanations on how it arrives at the predicted class (Nygård & Mezei, 2020).

Regarding the process of building a decision tree, it can be divided into two phases: tree construction, which starts with all of the training observations at the root and then they are partitioned based on certain attributes, and tree pruning, which intends to remove branches that reflect noise or outliers (Gama et al., 2017).

3.3.4 Random Forests

One of the main problems of decision trees is that a single tree may develop overfitting problems as it grows deeper (Kotu & Deshpande, 2019). If a tree is too overfitted to the training data, it will not be able to generalize beyond the training examples, meaning it will have poor accuracy for unseen samples. This problem can be minimized through pruning the tree, as previously stated. However, a better solution can be to use random forests instead. Random forests tackle this issue by constructing multiple decision trees, where each tree selects a random sample of the training data and of the original features (Kotu & Deshpande, 2019). In the end, the class assigned to each observation is computed through majority voting.

3.3.5 Neural Networks

In the last years, neural networks have become a popular choice for supervised learning (Nygård & Mezei, 2020). They are based on deep learning, a subfield of machine learning where the algorithms are inspired on the human brain (Duggal, 2020).

A neural network is composed by thousands of interconnected nodes – the processing units of the network (Hardesty, 2017). The networks train themselves to recognize patterns, through the input data that is fed to them, and to predict the outputs for unseen data (Duggal, 2020).

Neural networks are organized into several layers of nodes (Hardesty, 2017): the input layer, which receives the input, the output layer, which predicts the final output, and the hidden layers, which are located between the previous two and perform the computations (Duggal, 2020). The neurons of a layer connect with neurons of the following layers through channels.

A node assigns a numerical value called weight to each channel (Hardesty, 2017). The inputs are multiplied by the corresponding weights, and the sum is sent to the neurons in the hidden layers. The resulting value is passed through a threshold function – the activation function – which determines if the neuron will be activated and transmit data or not. If the number is below the threshold, it will not pass data to the next layer (Hardesty, 2017). Otherwise, if it surpasses the threshold, the node will fire, which means it will send the sum of the weighted inputs through its outgoing connections (Hardesty, 2017). In the output layer, the neuron with the highest value fires and determines the output.

In the process of training the network, initially, the weights and the thresholds are set to random values, which are then continually adjusted, through iterative backward and forward propagation, such that the model is able to predict the right label for most of the data (Hardesty, 2017).

3.3.6 Results

Nygård and Mezei (2020) computed the usual performance metrics based on the confusion matrix to assess the performance of the attempted models, in which a positive case refers to a lead that has converted into a customer. As previously seen, the common metrics are the accuracy, the precision, the sensitivity and the specificity. Additionally, the authors considered the Area Under the Curve (AUC), obtained by computing the area under the Receiver Operating Characteristics curve, which plots the true positive rate against the false positive rate for various probability thresholds (Nygård & Mezei, 2020).

Table 7 presents the performance metrics obtained for each model. In terms of accuracy, the neural network showed the best results, however this model treated the classes very differently, as noted by Nygård and Mezei (2020) – the sensitivity obtained was 36% and the specificity was 90%. Hence, this model correctly predicted 90% of the negative cases, but only guessed right 36% of the positive cases. Since in this case the positive cases are the most important ones to detect, this model would not be the most adequate choice. Looking at the sensitivity, the model with the highest score was the logistic regression, however it achieved the worse specificity, meaning it is the best at predicting positive cases, but the worse at identifying the negative ones. Lastly, the decision tree and the random forest obtained similar results, however the Random Forest was a slightly better performer, achieving a higher sensitivity and AUC than the decision tree; this may be justified by the fact that, as explained in Subchapter 3.3.4, random forests perform a random data and feature selection, resulting in less overfitted models. Nygård and Mezei (2020) considered that the random forest was the best model, as it achieved the highest AUC.

***Table 7** - Model performance comparison.*

Model	Accuracy	Sensitivity	Specificity	AUC
Logistic Regression	0.59	0.77	0.58	0.70
Decision Tree	0.69	0.66	0.69	0.72
Random Forest	0.69	0.69	0.69	0.76
Neural Network	0.86	0.36	0.90	0.75

Source: Nygård & Mezei (2020)

3.4 Models' Comparison

To conclude this chapter, a brief comparison of the models will be presented.

Regarding the first model, the Bayesian network, even though it achieved good results, it is a highly complex model to build, especially if there is not enough data available to define the structure of the network. It requires a lot of time, skill and close communication with experts in the field. Hence, due to these requirements, it might not be the best model to use in practice, especially if there is only a small amount of data available and a short deadline to meet.

The remaining models, presented in Subchapters 3.2 and 3.3, appear to be better options, as they do not require as many resources. However, it can be argued that the models proposed by Nygård and Mezei (2020) are less complete, as they only predict the leads' conversion outcome; the DQM allows for better contact prioritization as the outputs are objective scores and not classifications. Moreover, it can be argued that the FFM surpasses the other models, as it computes the expected revenue of a lead in monetary units. This allows Sales to rank the leads according to their expected revenue and to anticipate the budget invested to pursue that lead. Nonetheless, this model requires additional data to estimate the values of E(revenue|x, won).

In conclusion, among the models presented in Chapter 3, the FFM can be considered as the most complete one, as it allows for further contact prioritization in comparison to the remaining models. However, the FFM requires additional data, which may not always be available, especially in the early stages of the sales process.

Fourth Chapter

4. Research Methodology

The previous chapters were fundamental to present the important concepts regarding the theme of this dissertation, some of the models proposed in the literature and the research gap it intends to tackle. Now, Chapter 4 will present the research methodology adopted in this study.

There are three sections in this chapter: Subchapter 4.1, where the main goals of the research are discussed, Subchapter 4.2, where a contextualization of the study is presented, and, lastly, Subchapter 4.3, where the chosen data mining method is described.

4.1 Goals of the research

As previously discussed, the main goal of this dissertation is to present a possible approach for companies to efficiently perform lead scoring and contact prioritization. Even though this study was built for a specific company, the approach followed can be adapted and applied to businesses in other sectors, as this dissertation presents the steps involved in building an automated lead scoring model, through machine learning, and assessing its performance.

The research question underlying this dissertation can be summarized as: *How can a B2B company automate the process of lead scoring?.* This research question can be decomposed into three articulated objectives, that describe what we want to achieve with this project:

- Identify relevant trends and insights using data from past leads that converted into customers.
- Identify which attributes are relevant to predict lead conversion at an early stage of the sales process.

- Develop a lead scoring model that allows the Sales team to adequately perform contact prioritization.

The subject of this analysis is a company called HUUB, which will be presented in Subchapter 4.2. Given the digital evolution that has been occurring in the last decades, namely the advances in Artificial Intelligence, it is reasonable to consider the possibility of automating the lead scoring process in companies. However, due to lack of literature in this topic, it is not very clear how this can be achieved in practice.

HUUB proposed the topic of this dissertation as the company believed that the development of an automated lead scoring mechanism would have a positive impact in terms of prioritizing the contacts. The start-up supplied the data needed for building the model, which was extracted from its CRM system, HubSpot. Moreover, HUUB's Sales team was also crucial to this project, as it provided relevant information regarding how the sales processes were currently being handled and which aspects needed improvement.

4.2 Contextualization of the research – HUUB

The research conducted during this dissertation took place at a Portuguese start-up called HUUB, a logistics company that operates in the fashion field, which initiated its activity in 2015.

HUUB was created by four founders: Luís Roque, the former Chief Executive Officer (CEO), Tiago Paiva, the current CEO and former Chief Marketing Officer, Pedro Santos, the Chief Operating Officer, and Tiago Craveiro, the Chief Product Officer. Together, and combining years of academic and professional knowledge in various backgrounds, these men were able to convert an innovative business idea into a profitable reality. As expected from a start-up, the team has grown considerably throughout the years, counting with over 60 employees in December of 2020.

Even though HUUB is still recent in the market, it has conquered some major achievements in its lifespan. In 2018, the start-up received 2.500.000€ from the investment company Pathena, made its first appearance at Web Summit, and was nominated as one of the "sexiest start-ups in Europe". In 2019, HUUB received 1.500.000€ at an investment round from the logistics company Maersk and made its second appearance at Web Summit. Currently, HUUB is going through its expansion phase – the company is focused in acquiring new customers and leveraging on technology and automation to grow and scale up the business.

HUUB is a tech fashion start-up that entered the market with the goal of being disruptive in the logistics industry, by managing the supply chain of its clients from end-to-end and giving them full visibility over the entire process. HUUB's clients are mainly small or medium-sized independent fashion brands, such as NU-IN and Être Cécile. These brands rely on HUUB to manage their entire supply chain, from suppliers to final customers, which may be B2B or B2C. HUUB is in charge of all processes: receiving purchase orders from the brands' suppliers, shipping orders to customers and handling returns. As opposed to the traditional rules of the logistics sector, HUUB defines a price per item for the entire operation, making the process very transparent to its clients.

Regarding the business model, HUUB is a Software as a Service company. HUUB's Tech team developed an integrated digital platform, Spoke, which provides the clients with full visibility over their business and supply chain. Spoke stores real-time information regarding suppliers, customers, orders, products and stocks, being its main goal to leverage the brands' growth, as it grants full visibility over the current operations, allowing the brands to monitor and optimize their supply chains.

Structure-wise, HUUB is organized into various departments, namely: Marketing & Communication, People, Finance, Product, Global Operations, Brand Success, Sales, Data Insights & Analytics, Artificial Intelligence, Tech,

and Data Engineering. Even though each department is in charge of certain processes and tasks, the day-to-day chores still require a lot of communication and cooperation among the teams to ensure that the Board's goals are achieved.

4.3 Method – CRISP-DM

Regarding the method adopted to answer the research question proposed in this dissertation, CRISP-DM was chosen as the most appropriate option. It should be noted, however, that other data mining methodologies, such as KDD and SEMMA, could have been used to achieve the same results. The choice of CRISP-DM is justified by its business-oriented nature, as it includes the stages of business understanding and deployment, unlike the other methods.

This method, formally known as Cross-Industry Standard Process for Data Mining, was assembled by DaimlerChrysler (then Daimler-Benz), SPSS (then ISL) and NCR in 1996 (Shafique & Qaiser, 2014). However, it was continuously refined and improved throughout the years, until it was officially published as CRISP-DM 1.0 in 2000 (Pete et al., 2000).

This methodology is useful for both novices and experts in data mining, as it provides a set of clear guidelines that conduct the researcher throughout the knowledge discovery process. The developers of this methodology wanted it to be "industry-, tool-, and application-neutral" (Pete et al., 2000, p.1), which is why they discussed the method with practitioners from various backgrounds to ensure it could be applied to most data mining projects.

Figure 9 presents a schema of the CRISP-DM method. It should be noted that the sequence of phases does not have to be followed rigidly (Azevedo & Santos, 2008), as the output of each step influences which direction to go next (Bošnjak et al., 2009). Hence, the researcher is recommended to go back and forth between stages during his study. The arrows shown in Figure 9 merely indicate the most frequent dependencies between stages (Wirth & Hipp, 2000). The outer circle represents the cyclic nature of data mining – it often does not

end at deployment, as new business questions that need to be explored tend to arise throughout the process (Wirth & Hipp, 2000).

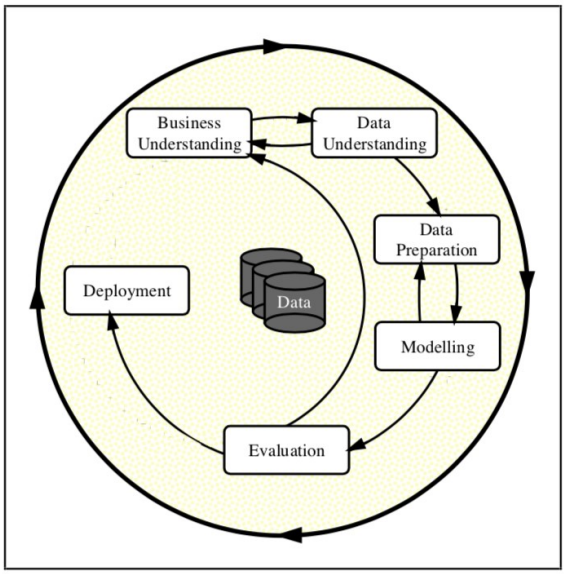


Figure 9 *– CRISP-DM method.*
Source: Wirth & Hipp (2000, p.5)

CRISP-DM includes six stages:

1) Business understanding – The first step, which is very business-oriented, consists of setting the goals the analyst wants to achieve with the project. It is important to fully understand the problem faced and the outcome desired, as well as uncover factors that may influence the final outcome. Furthermore, at this stage, the business goal should be translated into a data mining problem (Wirth & Hipp, 2000). Some examples of business goals include reducing costs, increasing sales, and searching for trends in the data (Bošnjak et al., 2009). According to Pete et al. (2000), there are four main tasks in this stage: determining business objectives, assessing the situation, determining data mining goals and producing a project plan.

2) Data understanding – This stage includes the collection of data for the project, as well as a preliminary analysis to allow the researcher to become familiarized with the data. The goal is to extract relevant insights, identify data quality issues, and detect relevant trends (Wirth & Hipp, 2000). At this stage, an

overview of the instances and variables of the dataset should be conducted (Bošnjak et al., 2009). The data exploration is helpful to support the analyst's hypotheses and provide further insight over the data. For this task, simple queries can be used, as well as visualization techniques, which can reveal trends in data and help to easily detect data quality issues, such as outliers. This stage can be summarized in four tasks: collecting the data, describing the data, exploring the data and verifying the quality of the data (Pete et al., 2000).

3) Data preparation – This step intends to select and prepare the data for the final dataset (Pete et al., 2000). This includes the usual tasks, such as data sampling (selecting the instances), feature selection (selecting the variables), data cleaning, attribute construction, among others (Wirth & Hipp, 2000). If any quality issues were detected in the previous stage, this is when they should be handled, to ensure data integrity for applying the modeling techniques in the next phase. Additionally, data integration may be required at this stage, if the original data was extracted from multiple sources (Pete et al., 2000). Moreover, this is usually when the data is split into two sets: the training dataset, which is used to train the model, and the test dataset, which is used to assess the model's performance. This stage can be resumed in five tasks: selecting data, cleaning data, constructing data, integrating data and formatting data (Pete et al., 2000).

4) Modeling – At this stage, one or more modeling techniques are selected and applied to the data (Pete et al., 2000). Since a method does not dominate the others at all times, this stage is often repeated for different algorithms, and the results are compared in the end (Wirth & Hipp, 2000). Here, it is important to document any assumptions of the model(s). Then, to build the model(s), the chosen algorithm(s) is run through the training dataset, using software, and the parameters of the model(s) are tuned to guarantee optimal results. To conclude this stage, the performance of the model(s) is assessed. This is usually done by computing metrics based on the confusion matrix, the AUC or error rates.

According to Pete et al. (2000), this stage includes four tasks: selecting the modeling technique, generating a test design, building and assessing the model.

5) Evaluation – The evaluation phase assesses the degree to which the model was able to achieve the business goals (Pete et al., 2000). Furthermore, at this stage, the researcher analyzes if there are goals the model has not been able to meet and what may be the reason behind it (Wirth & Hipp, 2000). Following this assessment, there is a chance to review the process, to identify if there are relevant factors that have been overlooked and to determine further actions that may be required (Pete et al., 2000). Pete et al. (2000) divide this stage in three tasks: evaluating the results, reviewing the process and determining the next steps.

6) Deployment – The sixth and final stage of CRISP-DM is the deployment. It allows to understand what needs to be done for the model to eventually be used in practice (Wirth & Hipp, 2000). Pete et al. (2000) divide this stage in four tasks: a plan of the deployment, a plan of the monitoring and maintenance, producing the final report and reviewing the project. Planning the deployment consists of determining a deployment strategy and the steps it includes. Defining a maintenance strategy helps to prevent incorrect usage of the data mining results. In some cases, a final report is written at the end of the project, with a summary of the results (Pete et al., 2000). Lastly, the project can be reviewed, to determine areas with potential for future improvement.

The six stages presented summarize the CRISP-DM method, providing a summary of its tasks and objectives. It should be noted, however, that not all of the tasks aforementioned were performed, due to the nature of this work.

Lastly, an overview of the methodology followed in this project, including the tasks performed and tools used, is presented in Appendix I.

Fifth Chapter

5. Case Study – HUUB

Whilst the previous chapter consisted of a theoretical explanation of the methodology used in this research, Chapter 5 will now present the CRISP-DM method being applied to this project at HUUB.

The subchapters of Chapter 5 correspond to the phases of the CRISP-DM method. Hence, Subchapter 5.1 focuses on assessing the current situation and setting the goals of the project, Subchapter 5.2 corresponds to the phases of data understanding and data preparation, Subchapter 5.3 comprises of the modeling phase, Subchapter 5.4 consists of the evaluation stage and, lastly, Subchapter 5.5, presents the deployment of the project.

5.1 Business Understanding

This first stage aims at understanding the problem we are trying to solve and setting the main goals of this project, along with studying the current situation.

The business objective of this study is to develop an automated lead scoring model that allows the Sales team to prioritize contacts. If we translate this project into data mining tasks, this objective requires: extracting the available data regarding leads from the CRM platform, conducting a preliminary analysis to get familiarized with the data and assess its quality, preparing the final dataset, choosing a machine learning algorithm for this classification problem and, lastly, using a software to build the model and assess its performance.

These steps will lead us to the final goal of this project, which is to be able to attribute a classification for each lead that predicts its conversion outcome. This will be helpful for the Sales team decision making process, as the members will

be able to define, among the brands that are entering the CRM platform, which ones should be contacted first.

At this stage, it is crucial to understand HUUB's sales process to have a full picture of the current situation and define the next steps. Hence, a meeting with the Sales team was held to gather information on this process.

5.1.1 HUUB's Inside Sales 2.0 Framework

HUUB's Sales team has designed a framework that provides an overview of the components of the sales process. The current version is called Inside Sales 2.0 framework (Figure 10).

In the past, this framework was focused on two sections: inside sales and field sales. Inside sales refers to the process of discovering brands, mainly through social media and paid databases, and cold contacting them, whilst field sales consists of meeting potential clients at fashion fairs or events. However, the COVID-19 pandemic had a major impact on this process, leading the Sales team to focus their efforts 100% on the Inside Sales component.

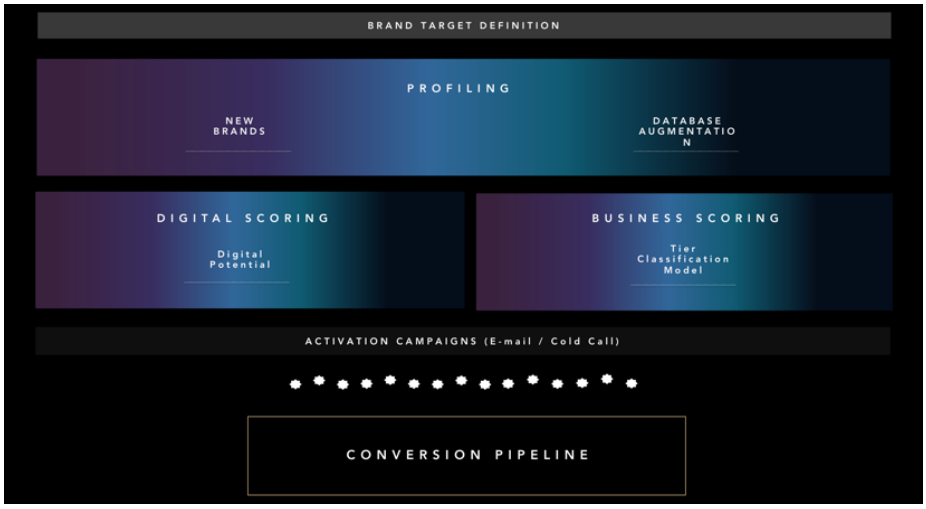


Figure 10 *– HUUB's Inside Sales 2.0 framework.*
Source: HUUB

The aforementioned process starts by developing a clear definition of the target brands. By relying on past experience and knowledge of the industry, the Sales team determined the characteristics of the brands they want to target.

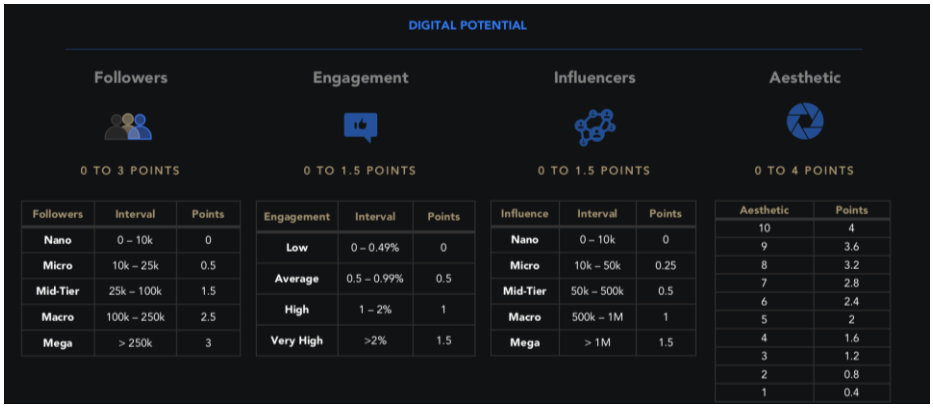
Then, there is the profiling stage, which is divided into two areas: new brands and database augmentation. New brands refer to brands that are not yet stored on HUUB's CRM platform, HubSpot, and that are found over time by the team and added to HubSpot along with their information. The second component, database augmentation, consists of updating the information fields of brands that are already in HubSpot but have not been in touch with HUUB in the last year, so that they can be contacted again.

Conceptually, profiling means identifying the relevant characteristics of a brand, to understand its limitations and competitive edge. Moreover, profiling is important because some profiling characteristics later result in the different campaign segmentations. Some characteristics are publicly known, so they are easier to obtain at an early stage, whilst others can only be obtained much later in the process. Some examples of attributes that HUUB uses for profiling are the country of the brand, where its production is located, the sales channels used, among many others.

As seen in Figure 10, after the profiling stage, it would be relevant to have a scoring model to prioritize the leads before the campaigns were activated. The schema shows two different scorings: digital scoring, which has been created by the Sales team, and business scoring, which has yet to be developed, and is the main goal of this dissertation.

The digital scoring/potential of a lead is a value that can range between 0 and 10 points. It is based on four components: Instagram followers, Instagram engagement, Instagram influencers associated to the brand and the brand's aesthetic. Each component has various categories with corresponding intervals to which a number of points has been assigned (Figure 11). Then, a calculation is made, where each component is assigned a specific weight, that has been

determined by the Sales team, based on literature reviews and empirical experience.



Followers	Interval	Points
Nano	0 – 10k	0
Micro	10k – 25k	0.5
Mid-Tier	25k – 100k	1.5
Macro	100k – 250k	2.5
Mega	> 250k	3

Engagement	Interval	Points
Low	0 – 0.49%	0
Average	0.5 – 0.99%	0.5
High	1 – 2%	1
Very High	>2%	1.5

Influence	Interval	Points
Nano	0 – 10k	0
Micro	10k – 50k	0.25
Mid-Tier	50k – 500k	0.5
Macro	500k – 1M	1
Mega	> 1M	1.5

Aesthetic	Points
10	4
9	3.6
8	3.2
7	2.8
6	2.4
5	2
4	1.6
3	1.2
2	0.8
1	0.4

Figure 11 *– HUUB's digital scoring.*
Source: HUUB

Even though the digital scoring does not have a practical component yet, as decisions are not being made based on those results, it nonetheless allows the team to quickly assess the fitting accuracy of each brand.

The second part of scoring, business scoring, is what HUUB does not have yet, but would greatly benefit from. According to the Sales team, a lead scoring model must be part of this framework, as with those results the campaigns activated would be a lot smarter, leading us to expect higher conversion rates.

5.1.2 HUUB's Need for a Lead Scoring Model

Currently, the Sales team activates two campaigns per week. Each campaign consists of four e-mails and two cold calls and is sent to approximately 30 brands. Hence, around 60 brands are contacted each week.

These campaigns are segmented by five main themes: e-commerce brands, brands with production made in Portugal, Portuguese brands, Spanish brands and sustainable brands. The team filters the leads in HubSpot according to the campaign theme criteria and selects 30 brands for each campaign. Hence, the lead scoring model would be used at this moment, so that the selection of the 60

brands would no longer be random, but instead based on the leads' predicted conversion outcome.

Among the several brands that enter the CRM platform daily, it can be challenging to choose which ones should be contacted first in the Inside Sales campaigns. Therefore, the lead scoring model would allow the Sales team to prioritize the leads in an adequate and efficient manner.

5.2 Data Understanding and Data Preparation

The stage of data understanding includes the collection of the data and its description, as well as a preliminary analysis that aims at identifying data quality issues and detecting relevant trends. Then, the data preparation stage intends to prepare the final dataset, by performing tasks such as data sampling, feature selection, data cleaning and attribute construction.

As noted in Chapter 4, the sequence of phases of the CRISP-DM method does not have to be followed rigidly. Hence, in Subchapter 5.2, tasks comprised in both the second and third stages of CRISP-DM will be performed.

5.2.1 HubSpot's Organization

The data used in this project was stored in HUUB's CRM platform – HubSpot. HubSpot has several pipelines through which the deals[1] can move along. Each pipeline has various stages, and as a lead progresses in the process, the Sales team updates its stage and/or pipeline in HubSpot.

Currently, the most important pipelines for the Sales team are the Inside Sales and the Conversion pipelines. The Inside Sales pipeline stores leads that have been discovered by the Sales team and that are being profiled and contacted through campaigns, whilst the Conversion pipeline refers to brands that have already been qualified and are in close contact with HUUB.

[1] HubSpot uses the term deal to define a lead, so both terms will be used interchangeably.

HubSpot's layout for the Inside Sales pipeline is presented in Figure 12. We can visualize the brands[2] that are currently on that pipeline, as well as the stage of the pipeline each brand is currently in. For instance, brand A is currently in the profiling stage of the Inside Sales pipeline, whilst brand B has already been in profiling and has been moved to the in campaign stage. The same logic applies to the Conversion pipeline (Figure 13).

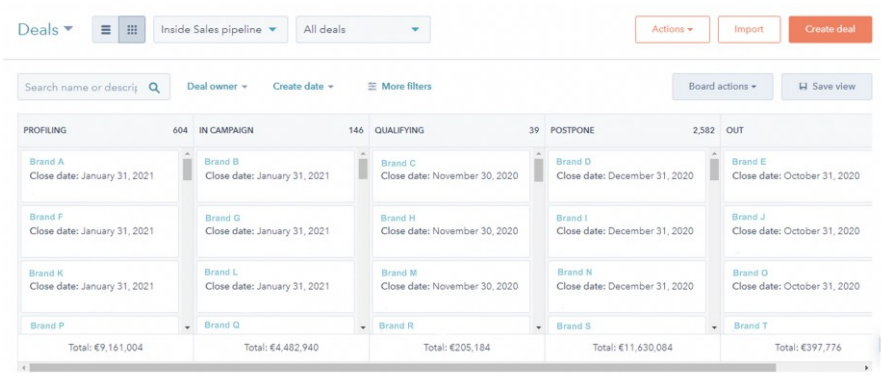


Figure 12 *– HubSpot's Inside Sales pipeline layout.*

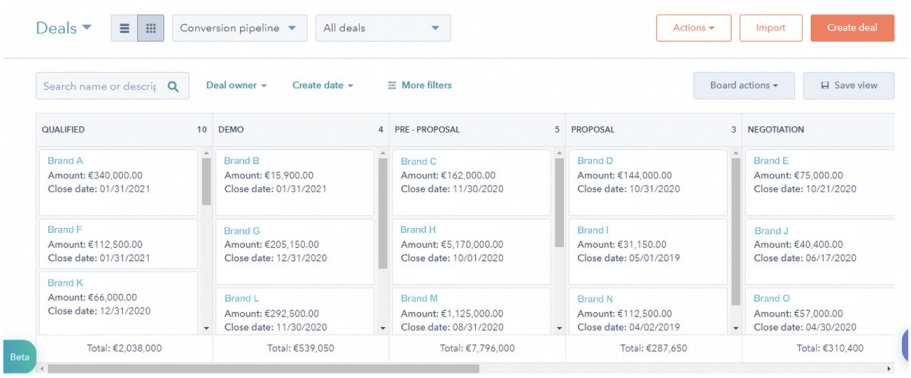


Figure 13 *– HubSpot's Conversion pipeline layout.*

Figure 14 presents a more detailed schema of the Inside Sales pipeline, which comprises of five stages:

- Profiling – when a brand that fits HUUB's criteria has been found by Sales associates and is added to HubSpot along with the respective information.
- In campaign – when a deal enters a campaign, which consists of e-mails and cold calls that present HUUB's value proposal.

[2] The brands' names were omitted for confidentiality purposes.

- Qualifying – when a brand has shown interest in HUUB, the Sales team gathers more information to determine if there is a good fit between HUUB and the brand. After this stage, the deal is either moved to the Conversion pipeline, if there is a fit with HUUB, or to the stages Postpone or Out.
- Postpone – when a deal is not yet a perfect match with HUUB, it stays in the stage Postpone until it is ready to re-enter a new campaign.
- Out – when a brand asks not to be contacted again by HUUB (for example, if it asks to be removed from the e-mailing list).

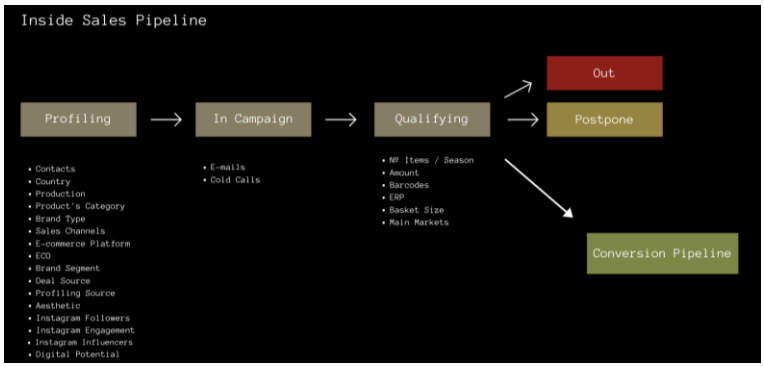


Figure 14 *– Inside Sales pipeline.*

The same logic applies to the Conversion pipeline (Figure 15), which includes the stages:

- Qualified – when all of the requirements have been met, a deal is qualified. At this point, the Sales team will discuss with the Operation and Product teams if the partnership is viable.
- Demo – at this stage, a demonstration of Spoke is provided to the brands, along with a presentation of HUUB's cost structure.
- Pre-proposal – when the Sales team is working on the proposal documents to present to the brand.
- Proposal – when the proposal documents are sent to the brand and the team waits for feedback.
- Negotiation – when there is a negotiation of costs.
- Won – when HUUB has gained a new client.

- Postpone – when a deal is not closed because it is not yet the perfect match.
- Lost – when a brand has closed or bankrupted.

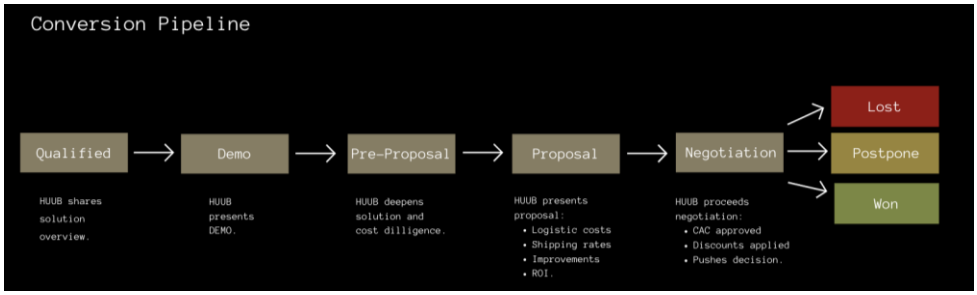


Figure 15 *– Conversion pipeline.*

The further down the sales funnel a deal is, the more information there is available and the more accurate it is.

Even though there are various pipelines and stages through which the deals move along, the information fields associated to them remain the same. HubSpot defines these information fields as "deal properties", which represent the characteristics of the brands.

5.2.2 HubSpot Deal Properties

An analysis of HubSpot's deal properties was conducted at this point. Since there were 85 properties, to promote the efficiency of this analysis, a meeting was held with the Sales team to understand the relevance of each property. It was concluded that many properties are not used by the team and could be disregarded. Hence, 51 properties were explored.

Appendix II presents an overview of these 51 fields; the first column indicates the name of the deal property and the second defines the HubSpot field type, namely:

- Date picker – stores date values automatically generated by HubSpot or inserted by a user (HubSpot, 2021).
- Calculation – stores equations computed by HubSpot based on other properties (HubSpot, 2021).

- Dropdown select and Radio select – store a maximum of 1000 options; only one can be selected as a value (HubSpot, 2021).
- HubSpot user – stores a list of the HubSpot users on the account; only one can be selected as a value (HubSpot, 2021).
- Single-line text and Multi-line text – store one or multiple strings of alphanumeric characters, respectively (HubSpot, 2021).
- Single checkbox – stores two options; only one can be selected as a value (HubSpot, 2021).
- Multiple checkbox – stores checkboxes with several options; multiple values can be selected (HubSpot, 2021).
- Number field – stores a string of numbers (HubSpot, 2021).

The third column of Appendix II specifies if the property is an open, closed or automatically generated field. This is relevant to understand which values each property can assume. In an open field, any value of that field type is accepted, whilst a closed field can only assume as values the pre-defined categories; lastly, there are fields that are automatically generated by HubSpot.

The fourth column of Appendix II contains a brief description of each property; lastly, the fifth column shows the percentage of missing values[3] for each property, which will be important for feature selection.

The second column of Appendix II suggests that further action is needed to ensure data quality and integrity, not only for the next steps of this project, but also for future analyses the company may develop. This initial analysis indicated some issues regarding the format in which the properties were being stored in HubSpot, namely:

- Brand Segment should be a radio or dropdown select field, such that only one value can be selected for a deal. Currently, it consists of multiple checkboxes, so several values can be assigned to a deal. Instagram Engagement, Instagram Followers and Instagram Influencers should also be

[3] Calculated for each property as: Number of missing values for the property/Number of Deals in HubSpot*100.

a radio or dropdown select field, such that only one value can be selected per deal.

- Country should also be a radio or dropdown select field. This property is currently a text field, which means that the same country can be written in different ways and that any text inserted is accepted. This makes the trends in the data harder to identify.
- Production is currently a text field but should also be stored as a radio or dropdown select, to avoid issues such as misspelling errors.
- Followers and Number of items per season should be stored as number fields, instead of text.

These recommendations were discussed with the Sales team, which agreed that these properties' format should be changed to improve data integrity. Hence, these format alterations were conducted by the Sales and Data Insights & Analytics teams and the properties are currently stored in the correct formats.

5.2.3 Feature Selection

After the analysis of the deal properties, feature selection was performed. This consists of selecting the relevant variables for building the model. It should be noted, nonetheless, that the algorithm used in this project has built-in feature selection, so the final classifier may actually disregard some of these variables.

This task was based on the following characteristics of the properties: the descriptions, the percentages of missing values and the stage in which that information is available; since the model will be used after the profiling stage, the selected variables must refer to data that is available to the Sales team at that stage.

Based on these criteria, out of the 51 deal properties presented on Appendix II, 13 were selected: Brand Segment, E-commerce Platform, ECO, Country, Instagram Followers, Instagram Engagement, Instagram Influencers, Product's

Category, Sales Channels, Brand Type, Production, Number of Contacts and Aesthetic Rank.

It is important to justify why the remaining properties were disregarded:

- The date properties do not provide any relevant information for predicting the conversion outcome of a deal, especially in the early stages of the funnel.
- Properties such as Number of items per season, Price per Item, Barcodes, Interest, Main Markets, Brand Tier, Amount and Amount in Company Currency would be relevant, but they are only available later in the process.
- Deal Source, Referral, Original Source Type, Deal Stage and Pipeline were disregarded since they always assume the same value for deals in the profiling stage of the Inside Sales pipeline.
- Properties such as Deal ID, Deal Name, Deal Description, Campaign Name, Profiling Source, Deal Owner and Sales Owner were excluded since they do not hold any relevant information for the classifier.
- Tradeshow was disregarded, since the model will focus on Inside Sales deals, and not on Fairs deals (which are stored in the Fairs pipeline).
- Basket Size and ERP were disregarded, since they have a percentage of missing values of almost 100%.
- Won Reason, Loss Reason and Postpone Reason were excluded since they represent the underlying reasons for the conversion outcomes.
- Fulfillment Center was disregarded as it is only used for already won deals.
- Original Source Data 1 and 2 are open fields, where any information can be written, which makes them less interesting to include in the model since there are no pre-defined categories.
- Followers and Digital Potential were excluded because other properties that represent the same thing have been selected (Instagram Followers and the four components of the digital scoring).

5.2.4 Preliminary Data Analysis and Data Cleaning

The data stored in HubSpot was integrated with HUUB's data warehouse. The Data Insights & Analytics team developed the data necessities and designed the data model (fact and dimension tables), whilst the Data Engineering team was in charge of integrating the data. Once this process was completed, a dataflow was created in Microsoft Power BI, using HUUB's PRO account; this ensures faster data updates once it is being used for analyses.

The following paragraphs present a preliminary data analysis for the variables chosen during feature selection. This is crucial to get familiarized with the data, identify data cleaning requirements and extract insights. This analysis was performed with Microsoft Power BI, by using the dataflow as the data source and establishing the relations between the dimension and fact tables (Appendix III).

1. Brand Segment

Brand Segment is a categorical variable with four classes: Luxury (>500€), Premium (150-500), Value (50-150) and Low Cost (<50€). This property segments the brands by the average price of their products.

As previously noted, this property was being stored in HubSpot as a multiple checkbox field, which allowed for multiple categories to be selected. The data showed that this happened for 11 deals, for which more than one brand segments had wrongly been selected. These deals were detected and passed to the Sales team, which corrected them in HubSpot. The dataflow and the Power BI file were then refreshed for this analysis.

Table 8 shows that many deals have a missing value[4] for this property, amounting to approximately 38% of the instances. Excluding those cases, the most frequent class is Premium, with 2106 deals. On the other hand, the least frequent class is Low Cost, with only 142 deals.

[4] The number of missing values appears on the frequency tables in the category "Blank".

By looking at past won deals (Table 9), it is visible that the majority is of brand segment Premium, corresponding to approximately 85% of the won brands. On the other hand, only 1 deal Low Cost has become a client – this is justified by the increased difficulty these brands have in assuming the logistics price in their operation.

Table 8 *– Absolute and relative frequencies of deals for variable Brand Segment.*

Brand Segment	Absolute Frequency	Relative Frequency
Blank	2455	37,93%
Low Cost (<50€)	142	2,19%
Value (50-150)	1517	23,44%
Premium (150-500)	2106	32,54%
Luxury (>500€)	253	3,91%
Total	**6473**	**100,00%**

Table 9 *– Absolute and relative frequencies of deals won for variable Brand Segment.*

Brand Segment	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Blank	3	1,91%
Low Cost (<50€)	1	0,64%
Value (50-150)	18	11,46%
Premium (150-500)	133	84,71%
Luxury (>500€)	2	1,27%
Total	**157**	**100,00%**

2. E-commerce Platform

E-commerce Platform is a categorical variable, for which 58 different categories appear in HubSpot. Most of the categories are e-commerce platforms' names, however there are some extra categories such as None (brand does not use any e-commerce platform), Other (platform used by the brand is none of the possible categories) or Not Identified (HUUB has not been able to identify the platform used by the brand).

Table 10 shows that there are approximately 30% missing values for this property. Excluding those, the most frequent platform is Shopify, with 1580 deals, followed by WooCommerce, with 820 deals. The category Other accounts for 526 deals and 148 deals do not use any e-commerce platform.

Table 11 shows the distribution of won deals by the different e-commerce platforms. It is visible that almost half of those leads use Shopify (77 deals). This was expected since HUUB's platform, Spoke, can be integrated with Shopify,

which may impact the brands' decision to work with HUUB. The second most frequent platform is WooCommerce, which is also one of HUUB's integrations.

Table 10 *– Absolute and relative frequencies of deals for variable E-commerce Platform.*

E-commerce Platform	Absolute Frequency	Relative Frequency
Blank	1931	29,83%
Shopify	1580	24,41%
WooCommerce	820	12,67%
Other	526	8,13%
Prestashop	295	4,56%
Magento	290	4,48%
Not Identified	277	4,28%
Cart Functionality	273	4,22%
None	148	2,29%
Squarespace	62	0,96%
Salesforce Commerce Cloud	56	0,87%
Wix	54	0,83%
Shopware	33	0,51%
OpenCart	20	0,31%
OXID Eshop	13	0,20%
BigCommerce	7	0,11%
Drupal Commerce	6	0,09%
Big Cartel	5	0,08%
Shoplo	5	0,08%
Ecwid	4	0,06%
Hybris	4	0,06%
LogiCommerce	4	0,06%
Visualsoft	4	0,06%
CS Cart	3	0,05%
Demandware	3	0,05%
Edrone	3	0,05%
Intershop	3	0,05%
JetShop	3	0,05%
Shoper	3	0,05%
Gambio	2	0,03%
IBM Websphere Commerce	2	0,03%
JTL-Shop	2	0,03%
PlentyMarkets	2	0,03%
SmartWeb	2	0,03%
textalk	2	0,03%
T-Soft	2	0,03%
VirtueMart	2	0,03%
blubolt	1	0,02%
DynamicWeb	1	0,02%
GetShopped	1	0,02%
Global-e	1	0,02%
IAI SHOP	1	0,02%
In Development	1	0,02%
Jumpseller	1	0,02%
MyStore NO	1	0,02%
NET2B	1	0,02%
Net-a-Porter	1	0,02%
nopCommerce	1	0,02%
OneStop Internet	1	0,02%
Shoprocket	1	0,02%
Shopsoftware	1	0,02%
Snipcart	1	0,02%
Stylehub	1	0,02%
SupaDupa	1	0,02%
Sylius	1	0,02%
Tiendy	1	0,02%
Vendd	1	0,02%
Versa Commerce	1	0,02%
xt Commerce	1	0,02%
Total	**6473**	**100,00%**

Table 11 *– Absolute and relative frequencies of deals won for variable E-commerce Platform.*

E-commerce Platform	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Blank	1	0,64%
Not Identified	1	0,64%
Pretashop	4	2,55%
None	7	4,46%
Other	28	17,83%
WooCommerce	39	24,84%
Shopify	77	49,04%
Total	**157**	**100,00%**

3. ECO

ECO is a binary variable with two possible options, True and False. This property determines if the brand is environmentally sustainable, in which case the value assigned is True, or False otherwise.

ECO is a relevant property because most of HUUB's clients are "slow fashion" independent brands. Moreover, one of HUUB's on-going projects is

related to giving its clients visibility over their carbon footprint, by measuring the impact they have in terms of carbon emission.

Table 12 shows that the vast majority of leads in HubSpot have a blank value for this property, amounting to approximately 81% of the leads database. However, for the remaining brands, it is clear that the dominating category is True, with 1215 deals, whilst the False category only has 28 deals. This shows that many brands have the mission to be environmentally sustainable, which may influence their level of interest in developing a partnership with HUUB.

By looking at past won deals (Table 13), and disregarding the missing values, the most frequent category is also True, with 26 deals.

Table 12 *– Absolute and relative frequencies of deals for variable ECO.*

ECO	Absolute Frequency	Relative Frequency
Blank	5230	80,80%
True	1215	18,77%
False	28	0,43%
Total	**6473**	**100,00%**

Table 13 *– Absolute and relative frequencies of deals won for variable ECO.*

ECO	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Blank	119	75,80%
True	26	16,56%
False	12	7,64%
Total	**157**	**100,00%**

4. Country

Country is a categorical variable which identifies the origin country of each deal. Hence, the possible categories are countries' names. Currently, there are 73 different countries being used.

As previously noted, this property was being stored in HubSpot as a text field. Hence, the original data showed that 52 deals had a country name that needed to be corrected. Some examples included misspelling errors (such as "Grecee" instead of Greece), continents or cities instead of countries (such as "Europe" and "Dubai"), and country names that were not standardized (such

as “United Kingdom” and “UK”). These deals were corrected in HubSpot by the Sales team and the data was refreshed for this analysis.

Table 14 shows that the most frequent country is the United Kingdom, with 939 deals, followed by Spain, with 705 deals, Portugal, with 684 deals, and France, with 662 deals. Together, these four countries account for around 46% of the leads. Table 15 shows that these countries are also dominant for past won deals. However, for won deals, Portugal is the country with the highest absolute frequency (52 deals).

Due to HUUB’s fulfillment centers being located in Europe, it is expected that most brands that HUUB profiles and conquers are also from Europe, which justifies why most of the least frequent countries are located outside of Europe (Table 14).

Table 14 *– Absolute and relative frequencies of deals for variable Country.*

Country	Absolute Frequency	Relative Frequency
Blank	156	2,41%
United Kingdom	939	14,51%
Spain	705	10,89%
Portugal	684	10,57%
France	662	10,23%
Italy	553	8,54%
Germany	477	7,37%
Denmark	344	5,31%
The Netherlands	301	4,65%
USA	267	4,12%
Sweden	208	3,21%
Belgium	139	2,15%
Poland	116	1,79%
Greece	104	1,61%
Finland	93	1,44%
Norway	62	0,96%
Switzerland	59	0,91%
Australia	55	0,85%
Turkey	48	0,74%
Japan	45	0,70%
Austria	41	0,63%
Canada	39	0,60%
Ukraine	33	0,51%
Ireland	27	0,42%
Lithuania	27	0,42%
Russia	25	0,39%
Latvia	23	0,36%
Romania	17	0,26%
South Korea	16	0,25%
Brazil	14	0,22%
Hungary	14	0,22%
New Zealand	14	0,22%
Estonia	12	0,19%
Czech Republic	11	0,17%
Bulgaria	10	0,15%
Israel	10	0,15%
Moldova	9	0,14%
Scotland	8	0,12%

Country	Absolute Frequency	Relative Frequency
Serbia	8	0,12%
Iceland	7	0,11%
Slovakia	7	0,11%
United Arab Emirates	7	0,11%
Peru	6	0,09%
Singapore	6	0,09%
China	5	0,08%
India	5	0,08%
Luxembourg	5	0,08%
Argentina	4	0,06%
Croatia	4	0,06%
Cyprus	4	0,06%
Hong Kong	4	0,06%
Taiwan	4	0,06%
Mexico	3	0,05%
Slovenia	3	0,05%
Belarus	2	0,03%
Colombia	2	0,03%
Malta	2	0,03%
South Africa	2	0,03%
Andorra	1	0,02%
Bahrain	1	0,02%
Bangladesh	1	0,02%
Cape Town	1	0,02%
Chile	1	0,02%
Egypt	1	0,02%
Faroe Islands	1	0,02%
Ghana	1	0,02%
Indonesia	1	0,02%
Malaysia	1	0,02%
Monaco	1	0,02%
Nepal	1	0,02%
Saudi Arabia	1	0,02%
Thailand	1	0,02%
Tunisia	1	0,02%
Venezuela	1	0,02%
Total	**6473**	**100,00%**

Table 15 *- Absolute and relative frequencies of deals won for variable Country.*

Country	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Portugal	52	33,12%
United Kingdom	28	17,83%
France	20	12,74%
Spain	18	11,46%
Germany	9	5,73%
Denmark	5	3,18%
Sweden	4	2,55%
Belgium	3	1,91%
Switzerland	3	1,91%
USA	3	1,91%
Greece	2	1,27%
Latvia	2	1,27%
Russia	2	1,27%
The Netherlands	2	1,27%
Austria	1	0,64%
Estonia	1	0,64%
Finland	1	0,64%
Norway	1	0,64%
Total	**157**	**100,00%**

5. Instagram Followers

Instagram Followers is a categorical variable, which converts the number of Instagram followers of a brand into one of five categories: Nano, Micro, Mid-Tier, Macro and Mega.

As previously noted, this property was being stored in HubSpot as a multiple checkbox field, which allowed for many categories to be selected for a deal. The original data showed that this had happened for 3 deals, which were corrected in HubSpot by the Sales team, and the data was then refreshed.

Table 16 shows that 3804 deals have a blank value for this property, which may be related to the fact that many brands do not have an Instagram account. Excluding those, the most frequent category is Micro, with 1032 deals, followed by Mid-Tier, with 943 deals. Approximately 31% of the deals are concentrated on these two classes. The least frequent class is Mega, with 144 deals.

Table 17 shows similar trends for deals that have been won: many deals have missing values (119 deals), the most frequent categories are Micro and Nano, and Mega is still the least frequent category.

Table 16 *- Absolute and relative frequencies of deals for variable Instagram Followers.*

Instagram Followers	Absolute Frequency	Relative Frequency
Blank	3804	58,77%
Nano	297	4,59%
Micro	1032	15,94%
Mid-Tier	943	14,57%
Mega	144	2,22%
Macro	253	3,91%
Total	**6473**	**100,00%**

Table 17 *- Absolute and relative frequencies of deals won for variable Instagram Followers.*

Instagram Followers	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Blank	119	75,80%
Nano	9	5,73%
Micro	17	10,83%
Mid-Tier	7	4,46%
Mega	1	0,64%
Macro	4	2,55%
Total	**157**	**100,00%**

6. Instagram Engagement

Instagram Engagement is a categorical variable with four categories: Low, Average, High and Very High. This property defines the level of engagement of the brands' Instagram account.

Table 18 shows that the majority of deals have a missing value for this property, amounting to approximately 59% of the brands. Besides those, the most frequent category is Average, with 969 deals, followed by High, with 678 deals. The least frequent category is Very High, with 429 deals.

By analyzing the won deals (Table 19), it is visible that, excluding the blanks, most of the brands have an Average, High or Very High engagement on Instagram; only approximately 6% of the deals won have a Low engagement.

Table 18 *– Absolute and relative frequencies of deals for variable Instagram Engagement.*

Instagram Engagement	Absolute Frequency	Relative Frequency
Blank	3803	58,75%
Low	594	9,18%
Average	969	14,97%
High	678	10,47%
Very High	429	6,63%
Total	**6473**	**100,00%**

Table 19 *– Absolute and relative frequencies of deals won for variable Instagram Engagement.*

Instagram Engagement	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Blank	77	49,04%
Low	9	5,73%
Average	27	17,20%
High	22	14,01%
Very High	22	14,01%
Total	**157**	**100,00%**

7. Instagram Influencers

Instagram Influencers is a categorical variable with six possible categories: Nano, Micro, Mid-Tier, Mega, Macro, None. This property defines the level of followers of the most popular influencer that collaborates with the brand on Instagram, and includes an additional category, "None", for brands that do not have partnerships with Instagram influencers.

Table 20 shows that there are 3703 missing values for this property. Excluding those, the most frequent class is Mid-Tier, with 1485 deals. On the other hand, the least frequent class is None – only 51 brands do not have partnerships with Instagram influencers.

Table 21 exhibits that 78 won deals have a blank value for this property. Aside from those, the most frequent class remains Mid-Tier and the least frequent remains None.

Table 20 *– Absolute frequencies of deals for variable Instagram Influencers.*

Instagram Influencers	Absolute Frequency	Relative Frequency
Blank	3703	57,21%
Nano	105	1,62%
Micro	378	5,84%
Mid-Tier	1485	22,94%
Mega	352	5,44%
Macro	399	6,16%
None	51	0,79%
Total	**6473**	**100,00%**

Table 21 *– Absolute frequencies of deals won for variable Instagram Influencers.*

Instagram Influencers	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Blank	78	49,68%
Nano	8	5,10%
Micro	26	16,56%
Mid-Tier	35	22,29%
Mega	5	3,18%
Macro	3	1,91%
None	2	1,27%
Total	**157**	**100,00%**

The next four variables have the particularity that multiple categories can be selected in HubSpot for each deal. Hence, the frequency tables will only present the absolute frequencies of each single class (instead of combinations of classes, which would harden the task of identifying trends in the data). Aggregated totals and relative frequencies were not computed because the frequencies are not mutually exclusive, so those values could not be correctly interpreted here.

8. Product´s Category

Product's Category is a categorical variable that details the types of products sold by the brands. There are seven product categories: Apparel, Footwear, Swimwear, Homewear, Underwear, Accessories and Other (residual category).

Table 22 shows that there are 138 missing values for this property. The most frequent class is Apparel – of the 6473 deals in HubSpot, 4302 sell apparel items. The other categories that stand out are Accessories, with 1811 deals, and Footwear, with 1085 deals. Table 23 shows similar trends for deals that have been won: categories Apparel, Accessories and Footwear are the most frequent, and Underwear is the least frequent product category.

Table 22 *– Absolute frequencies of deals for variable Product's Category.*

Product's Category	Absolute Frequency
Blank	138
Accessories	1811
Apparel	4302
Swimwear	329
Footwear	1085
Homewear	150
Underwear	173
Other	587

Table 23 *– Absolute frequencies of deals won for variable Product's Category.*

Product's Category	Absolute Frequency (deals won)
Blank	1
Accessories	30
Apparel	112
Swimwear	9
Footwear	29
Homewear	6
Underwear	2
Other	10

9. Sales Channels

Sales Channels is a categorical variable that details the sales channels used by the brands, such as: Wholesale, Ecommerce, Own Stores and Marketplace.

Table 24 shows that there are 394 deals with no value assigned. The most frequent class is Ecommerce, with 4469 deals, closely followed by Wholesale, with 4040 deals. These two categories are clearly the predominant classes. On the other hand, the least frequent channel is Marketplace, which is used by only 186 brands.

Table 25 shows the same trends for deals that have been won. Ecommerce and Wholesale are the dominating classes, with 132 and 101 deals, respectively, and Marketplace is the least frequent with only 3 brands.

Table 24 *– Absolute frequencies of deals for variable Sales Channels.*

Sales Channel	Absolute Frequency
Blank	394
Ecommerce	4469
Marketplace	186
Own Stores	1185
Wholesale	4040

Table 25 *– Absolute frequencies of deals won for variable Sales Channels.*

Sales Channel	Absolute Frequency (deals won)
Blank	1
Ecommerce	132
Marketplace	3
Own Stores	12
Wholesale	101

10. Brand Type

Brand Type is a categorical variable that defines the gender of the products sold by the brand, Men or Women, and the age category of the consumers, Adult or Kids. Furthermore, it has two additional categories: Home, if the products are home decoration items, and Other, which is a residual category for brand types that do not fit in other classes.

Table 26 shows that, among Adult and Kids, the most frequent age class is Kids, with 1688 deals. Also, between Men and Women, Women shows a significantly higher absolute frequency (3143 brands). Then, 35 brands sell Home products and 49 deals have been assigned the category Other. The data also shows that there are 386 missing values for this property.

Table 27 shows slightly different trends for the deals won: the category Kids still has a higher absolute frequency then Adult, with 66 versus 17 deals, respectively, but now Men is slightly more frequent than Women, with 55 versus 51 deals, respectively. Home is still the least frequent category, with only 5 deals.

Table 26 *– Absolute frequencies of deals for variable Brand Type.*

Brand Type	Absolute Frequency
Blank	386
Adult	1596
Home	35
Kids	1688
Men	1845
Other	49
Women	3143

Table 27 *– Absolute frequencies of deals won for variable Brand Type.*

Brand Type	Absolute Frequency (deals won)
Blank	2
Adult	17
Home	5
Kids	66
Men	55
Other	8
Women	51

11. Production

Production is a categorical variable that details the countries or continents[5] in which the brand produces its products.

Similarly to the Country field, this property had to go through data cleaning as it contained errors, such as city names instead of countries/continents, misspelling errors and not standardized terms (such as "United States of America", "USA" and "US"). These deals were corrected by the Sales team and the data was then refreshed.

Table 28 shows that there are 2582 missing values for this property. Excluding those, Portugal is the most frequent location – 1362 brands have Portuguese suppliers. Then, Italy follows, with 653 deals, and Spain with 532. Similarly to Country, many of the least frequent locations are non-European countries, such as Argentina, United Arab Emirates and Senegal. Table 29 shows that the most frequent production locations for won deals are Portugal, with 128 brands, and China, with 22 deals.

[5] The continents are only selected if the Sales team does not know the specific countries in which the brand produces.

Table 28 *– Absolute frequencies of deals for variable Production.*

Production	Absolute Frequency
Blank	2582
Portugal	1362
Italy	653
Spain	532
China	238
United Kingdom	237
France	233
India	153
Turkey	147
Germany	120
Europe	103
Poland	90
USA	89
Greece	63
Peru	45
Romania	42
Japan	38
Finland	35
Lithuania	34
Netherlands	32
Denmark	31
Bulgaria	26
Indonesia	26
Sweden	26
Vietnam	26
Brazil	26
Estonia	25
Nepal	24
Ukraine	24
Latvia	20
Belgium	19
Bangladesh	17
Canada	16
Morocco	16
Tunisia	14
Australia	13
Czech Republic	13
Russia	13
Switzerland	13
Hungary	12
Thailand	11
Mexico	10
Scotland	10
Serbia	10
South Korea	10
Austria	9
Ireland	9
Mongolia	9
Pakistan	9
Slovakia	9
New Zealand	7
Asia/Pacific Region	6
Madagascar	5
Moldova	5
Norway	5
Philippines	5
South Africa	5
Bolivia	4
Israel	4
Kenya	4
Sri Lanka	4
Egypt	3
Ghana	3
Mauritius	3
Slovenia	3
Bali	2
Cambodia	2
Colombia	2
Croatia	2
Georgia	2
Africa	1
Albania	1
Argentina	1
Belarus	1
Chile	1
Ethiopia	1
Guatemala	1
Hong Kong	1
Iceland	1
Laos	1
Macedonia	1
Malawi	1
Nicaragua	1
Scandinavia	1
Senegal	1
Singapore	1
Taiwan	1
Tanzania	1
United Arab Emirates	1

Table 29 *– Absolute frequencies of deals won for variable Production.*

Production	Absolute Frequency (deals won)
Blank	1
Portugal	128
China	22
India	9
Spain	9
Turkey	8
France	5
Italy	5
Bulgaria	3
Romania	2
Nepal	2
Germany	2
Poland	2
United Kingdom	2
Thailand	2
Peru	2
Austria	1
Brazil	1
Europe	1
Japan	1
South Africa	1

12. Number of Contacts

Number of Contacts is a discrete numerical variable that represents the number of contacts (phone numbers and e-mail addresses) associated to each deal in HubSpot. Since it is automatically defined by HubSpot, there are no missing values for this property.

The contact is a relevant information field, because even if a brand is well profiled with the important characteristics, Sales representatives need the contact of someone within the brand in order to reach them. Currently, the Sales team uses a platform called Apollo, which is a huge database of contact records, that helps perform this task.

The data shows that the average number of contacts per deal is 1.85 (Table 30). Table 32 shows that the most frequent value, the mode, corresponds to having 1 contact, which happens for approximately 58% of the deals. On the other hand, the least frequent values are the highest ones – there are only 2 deals with 12 contacts associated and 1 deal with 10 contacts.

Furthermore, it is visible that most of the data is concentrated on 1 and 2 contacts, amounting to around 85% of the leads database (Table 32).

Tables 31 and 33 perform the same analysis but for deals won in the past. For these deals, the most frequent value remains having 1 contact, however the average number of contacts is slightly lower – approximately 1.30 contacts per deal (Table 31). In terms of standard deviation, Table 31 shows a smaller value than Table 30, meaning that among the deals that were won there is less variability in the data for this property. Table 33 shows that the most frequent values for deals won are 1 and 2 contacts – together, they represent approximately 75% of the brands won.

Table 30 *– Descriptive statistic measures for variable Number of Contacts.*

	Mean	Mode	Standard Deviation
Number of Contacts	1,85	1,00	1,16

Table 31 *– Descriptive statistic measures of deals won for variable Number of Contacts.*

	Mean	Mode	Standard Deviation
Number of Contacts - Deals Won	1,30	1,00	0,93

Table 32 *– Absolute and relative frequencies for variable Number of Contacts.*

Number of Contacts	Absolute Frequency	Relative Frequency
0	121	1,87%
1	3724	57,53%
2	1802	27,84%
3	560	8,65%
4	177	2,73%
5	54	0,83%
6	16	0,25%
7	10	0,15%
8	5	0,08%
9	1	0,02%
10	1	0,02%
12	2	0,03%
Total	**6473**	**100,00%**

Table 33 *– Absolute and relative frequencies of deals won for variable Number of Contacts.*

Number of Contacts	Absolute Frequency (deals won)	Relative Frequency (% deals won)
0	24	15,29%
1	83	52,87%
2	34	21,66%
3	12	7,64%
4	3	1,91%
5	1	0,64%
Total	**157**	**100,00%**

13. Aesthetic Rank

Aesthetic Rank is a discrete numerical variable that ranges between 1 and 10 points and determines how attractive the aesthetic of a brand is.

Table 34 shows that there are 3859 missing values for this property. Excluding those, the average aesthetic rank is 6.91 points and the mode is 8, which is quite a high value considering the property's range (Table 34).

Table 35 shows that for deals that have been won, the average aesthetic rank is 6.35 and the mode is 6 – both values are lower than when we consider the entire leads database (Table 34). On the other hand, the standard deviation of the deals won is slightly higher (Table 35), which shows higher variability in the data. In terms of missing values, there are 106 blanks for deals won.

Table 34 *– Descriptive statistic measures and missing values for variable Aesthetic Rank.*

	Mean	Mode	Standard Deviation	Missing Values
Aesthetic Rank	6,91	8,00	1,53	3859

Table 35 *– Descriptive statistic measures and missing values of deals won for variable Aesthetic Rank.*

	Mean	Mode	Standard Deviation	Missing Values
Aesthetic Rank - Deals Won	6,35	6,00	1,62	106

Table 36 *– Absolute and relative frequencies for variable Aesthetic Rank.*

Aesthetic Rank	Absolute Frequency	Relative Frequency
Blank	3859	59,62%
1	1	0,02%
2	46	0,71%
3	83	1,28%
4	179	2,77%
5	339	5,24%
6	416	6,43%
7	663	10,24%
8	593	9,16%
9	262	4,05%
10	32	0,49%
Total	**6473**	**100,00%**

Table 37 *– Absolute and relative frequencies of deals won for variable Aesthetic Rank.*

Aesthetic Rank	Absolute Frequency (deals won)	Relative Frequency (% deals won)
Blank	106	67,52%
3	1	0,64%
4	6	3,82%
5	10	6,37%
6	11	7,01%
7	10	6,37%
8	6	3,82%
9	7	4,46%
Total	**157**	**100,00%**

As seen in the preliminary analysis, the data contains many missing values. This may be due to the fact that many information fields are hard to obtain, especially at early stages of the sales process. Many deals never make it to the qualifying stage – they go through profiling and/or campaigns and are postponed because they do not meet HUUB's criteria. Out of the 6473 deals in HubSpot, only 500 have been fully qualified.

Overview Report

Lastly, to visualize trends and extract insights from the data, a report was built using Microsoft Power BI, with HUUB's PRO account (Figure 16).

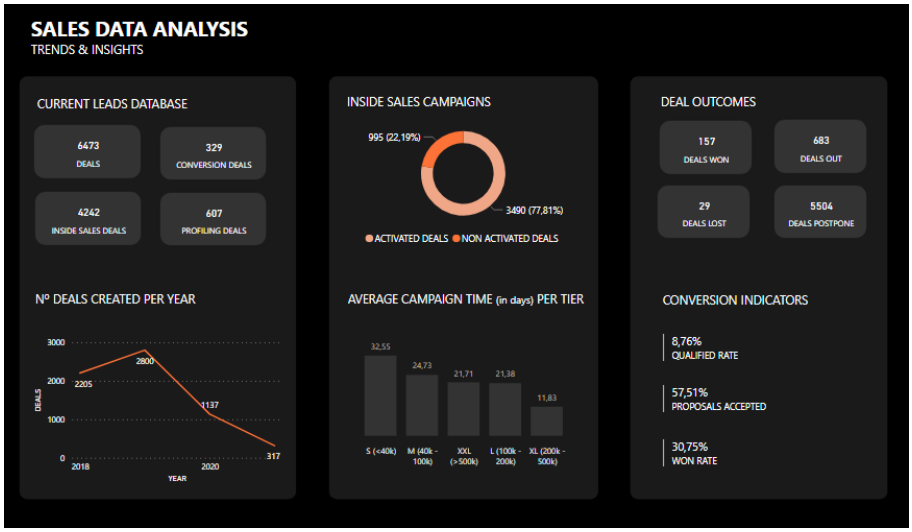


Figure 16 *– Sales data report in Microsoft Power BI.*

The upper left corner of the report contains information of the current leads database. Currently, there are 6473 deals in HubSpot, of which 1137 were added in 2020. Of the 6473 deals, 4242 are currently on the Inside Sales pipeline, and, of these, 607 deals are in the profiling stage. These 607 deals are the ones in which the lead scoring model would act, by attributing a conversion outcome to each, which would help the Sales team prioritize the deals to enter the campaigns. Then, on the lower left side, there is a plot of the number of deals created per year; this chart performs a distinct count of the deals' ID, segmented by the property Create Date[6]. The chart shows that 2019 was the year with the highest number of created deals (2800 brands).

The middle section of the report focuses on the Inside Sales campaigns. We can see the distribution of the Inside Sales deals that have been activated[7]. The chart shows that 3490 deals that have passed through the Inside Sales pipeline have been contacted through a campaign at least once; on the other hand, 995 deals have never been activated, which probably refer to deals that are either still being profiled or did not meet HUUB's criteria during profiling. On the bottom part of this section, a chart shows the average campaign time, in days,

[6] The year of 2017 was omitted since HUUB was still experimenting with HubSpot (14 deals were created during that year).
[7] A deal is activated when it enters a campaign.

per brand tier. The highest tiers, L, XL and XXL, exhibit slightly shorter average campaign times than the remaining tiers.

The upper right side of the report shows the number of deals per outcome: 157 deals have been won, 683 have been moved to the stage out, 29 have been lost and 5504 have been postponed.

Lastly, there are three relevant conversion indicators. The qualified rate corresponds to: number of deals qualified/number of deals activated; the data shows that approximately 9% of the deals that have been in a campaign have been qualified. The current value of this indicator is low, and the goal is for it to improve with the lead scoring model. Then, the data shows that, of the proposals that were presented to the brands, approximately 58% were accepted. Lastly, the won rate, computed as: number of deals won/number of deals qualified, is approximately 31%; hence, less than a third of the deals that have been qualified has converted into a customer.

5.2.5 Data Sampling and Attribute Construction

HubSpot stored 6473 deals at the time of this project. However, not all were used for training and testing the classifier. The final dataset should only contain deals that have had the conversion outcomes we want to predict – Won or Postpone.

To perform the data sampling, an Excel file was connected to the dataflow to extract the deals that have been through the stages Won or Postpone of the Conversion pipeline, along with the relevant properties (chosen during feature selection). The instances in the final dataset amount to 462 deals.

Since this is a classification problem, a new feature with the final classification of each lead was added to the dataset. This feature, Conversion Outcome, is a categorical variable with two possible values: *Won* – if the lead has converted into a new customer for HUUB at some point in the past (if it passed through the stage Won), or *Postpone*, for leads that have been postponed

in the conversion process[8] (if it passed through the stage Postpone of the Conversion pipeline).

The final sample, contained in an Excel file, was imported into a data frame in RStudio, called DF_DEALS_DATA. The summary of DF_DEALS_DATA showed that R was wrongly interpreting the categorical variables as "characters", so these were converted into "factors". The code and output for these actions is presented in Appendix IV.

Then, a new summary was requested (Figure 17), which shows that the final dataset contains 305 deals with conversion outcome Postpone and 157 with conversion outcome Won. The NA's represent the number of missing values for each variable, which were not removed since the chosen algorithm can handle missing values. Furthermore, the summary allows us to see the frequency of each category for categorical variables, and the descriptive statistic measures for the numerical variables.

```
> summary(DF_DEALS_DATA)
 CONVERSION_OUTCOME          BRAND_SEGMENT   ECOMMERCE_PLATFORM    ECO               COUNTRY    INSTAGRAM_FOLLOWERS
 Postpone:305       Low Cost (<50€)  :  8   Shopify    :184      FALSE: 15   Portugal      :135   Macro   : 16
 Won     :157       Luxury (>500€)   :  8   WooCommerce: 89      TRUE : 86   United Kingdom: 82   Mega    :  8
                    Premium (150-500):320   Other      : 81      NA's :361   France        : 48   Micro   : 76
                    Value (50-150)   : 94   None       : 24                  Spain         : 46   Mid-Tier: 75
                    NA's             : 32   Prestashop : 17                  Germany       : 26   Nano    : 79
                                            (Other)    : 26                  (Other)       :124   NA's    :208
                                            NA's       : 41                  NA's          :  1
 INSTAGRAM_ENGAGEMENT INSTAGRAM_INFLUENCERS          PRODUCTS_CATEGORY                      SALES_CHANNELS     BRAND_TYPE
 Average  :103        Macro    : 21          Apparel           :242     Wholesale;Ecommerce          :225    Kids     :139
 High     : 71        Mega     : 16          Footwear          : 68     Ecommerce                    :120    Women    : 87
 Low      : 48        Micro    : 80          Apparel;Accessories: 32     Wholesale                    : 81    Men      : 62
 Very High: 69        Mid-Tier:133           Accessories       : 28     Ecommerce;Own Stores         :  7    Men;Women: 60
 NA's     :171        Nano     : 30          Other             : 18     Wholesale;Ecommerce;Own Stores:  4    Adult    : 34
                      None     :  9          (Other)           : 67     (Other)                      :  9    (Other)  : 66
                      NA's     :173          NA's              :  7     NA's                         : 16    NA's     : 14
    PRODUCTION  NUMBER_CONTACTS  AESTHETIC_RANK
 Portugal:291   Min.   : 0.000   Min.   : 1.000
 China   : 28   1st Qu.: 1.000   1st Qu.: 4.000
 Spain   : 22   Median : 1.000   Median : 6.000
 Italy   :  9   Mean   : 1.552   Mean   : 5.982
 India   :  8   3rd Qu.: 2.000   3rd Qu.: 7.000
 (Other) : 82   Max.   :22.000   Max.   :10.000
 NA's    : 22                    NA's   :186
```

Figure 17 *– Summary of the final dataset.*

[8] If a deal has been through both the stages of Postpone and Won of the Conversion pipeline, the conversion outcome assigned to the lead is the most recent.

5.3 Modeling

In the fourth stage of the CRISP-DM method, the modeling technique is applied to the data to build the model and assess its performance.

5.3.1 Supervised and Unsupervised Learning

In machine learning, the common tasks can be categorized as supervised or unsupervised learning. The main difference is that in supervised learning the training set includes both input and output data. Hence, the model is constructed using labeled data, and then it is used to predict the value of the target variable of unseen data.

On the other hand, unsupervised learning does not have labeled outputs, which includes clustering and association analyses (Soni, 2018).

The main goal of this lead scoring model is prediction. In particular, we want to predict if a lead with a set of known characteristics will convert into a customer or not. For that, we will use past leads for which the conversion outcome is known. Hence, this problem fits into supervised learning.

Supervised learning can be further divided into classification or regression tasks. If the target variable is categorical, it is a classification problem, otherwise, if it is numerical, it is a regression problem.

In both regression and classification, the goal is to uncover relationships in the data that allow us to produce the correct output, by learning a function that best approximates the relationship between the input and output in the data (Soni, 2018). Common algorithms in supervised learning include logistic regression, Naïve-Bayes, support vector machines, neural networks, decision trees and random forests (Soni, 2018).

Since this lead scoring model will predict a categorical output, which states if a brand will convert into a customer or not, it is a classification problem.

5.3.2 Decision Trees

Even though several classification methods could be attempted for this problem, the chosen supervised machine learning model was a decision tree. Decision trees are one of the most used algorithms both in organizational and academic contexts (Gama et al., 2017).

Decision trees have been briefly presented in Subchapter 3.3.3 – they use a recursive partitioning algorithm for partitioning the observations of a dataset into different subsets. Several algorithms can be used for building decision trees, such as CART, ID3 and C4.5 (Bouchon-Meunier et al., 2006). In this case, the tree is a classification tree, as it separates the dataset into classes of the response variable (Kotu & Deshpande, 2015). From the resulting tree, classification rules can be extracted for decision-making.

The choice of decision trees was based on the advantages of this method. Firstly, decision trees provide an intuitive visual representation, making its results clear for any decision maker, unlike most algorithms (Kotu & Deshpande, 2015). Since the user of the model's results will be the Sales team, it seems adequate to choose a method that allows the team to easily understand the functioning behind it. This is a characteristic of decision trees that contributes to their popularity (Gama et al., 2017).

Other advantages of decision trees are that they perform feature selection, are able to handle numerical and categorical data, and do not require much data preparation (Gupta, 2017). Furthermore, they are flexible, as they do not assume any distribution for the data (Gama et al., 2017).

Nonetheless, decision trees also have some disadvantages. Firstly, they can incur in overfitting issues, such that the tree is unable to generalize to unseen data. Secondly, they are unstable, such that small variations in the training data may result in very different trees (Gama et al., 2017). However, there are ways to minimize these risks, such as pruning the tree to avert overfitting and

lowering the variance through methods such as bagging and boosting (Gupta, 2017).

The process of building a decision tree starts at the root node, which is the initial node that contains all of the observations of the training dataset. Then, a split condition is recursively applied at each non-leaf node to partition the data into subsets with the same output class label (Tangirala, 2020).

Ideally, each subset would only contain instances with the same output label. However, in most cases, this is impossible. Hence, at each node, the goal is to split the data using the attribute that minimizes the mixing of class labels, such that nearly pure (i.e., homogenous) sets are achieved (Tangirala, 2020). Hence, the selection of the attributes is important for the instances to be correctly classified (Raileanu & Stoffel, 2004).

There are various splitting criteria that can be used to assess the goodness of a split, being the Gini Index and the Information Gain two of the most commonly proposed in the literature (Raileanu & Stoffel, 2004). The Gini Index is used by the CART algorithm, and the Information Gain is used in ID3 and C4.5 algorithms (Tyagi, 2020).

A recent study (Tangirala, 2020) has concluded that, for classification trees, the choice of which splitting method to use does not impact the performance of the model – the results achieved, in terms of accuracy measures, is very similar. Moreover, Raileanu and Stoffel (2004) found that the Gini Index and the Information Gain criteria disagree on only 2% of the splits. Hence, for this model, the Gini Index will be used.

The Gini Index defines the purity of a class after splitting the data on a certain attribute, being the best split the one that increases the purity of the sets resulting from the split (Tangirala, 2020). The attribute that provides the split with the lowest Gini Index is chosen to split the node (Tyagi, 2020).

Mathematically, the Gini Index of a training dataset L with j class labels can be expressed as: $GINI(L) = 1 - \sum_{i=1}^{j} p_i^2$, where p_i is the relative frequency of class i in L (Tangirala, 2020). When L is split on an attribute A, resulting in subsets L_1 and L_2 with sizes N_1 and N_2 respectively, the Gini Index of that split is computed as $GINI_A(L) = \frac{N_1}{N} GINI(L_1) + \frac{N_2}{N} GINI(L_2)$; the reduction of impurity is the difference between GINI(L) and $GINI_A(L)$ (Tangirala, 2020).

The Gini Index ranges between 0 and 1. A Gini Index of 0 occurs when all elements belong to the same class (Tyagi, 2020) – this is the perfect scenario where the set is pure. On the other hand, a value of 1 means there is a random distribution of elements across various classes (Tyagi, 2020). A Gini Index of 0.5 happens for an equal distribution of elements over the classes (Tyagi, 2020).

5.3.3 Model Construction

In supervised learning, the data has to be split into two sets: one for training the model and another to assess its performance on unseen data (Kotu & Deshpande, 2015).

Regarding the choice of method to split the data, in classification problems performing a stratified random sampling is more adequate than a simple random sampling, as it ensures that both datasets have an equal distribution of output values (Kotu & Deshpande, 2015).

Hence, a command was used on RStudio to split the data (Appendix V). The training set usually contains between 70% and 90% of the original instances (Kotu & Deshpande, 2015). Since this sample is relatively small, 70% was defined as the proportion of instances for the training set. Therefore, of the 462 rows of the original dataset, approximately 30% will be in the test set and 70% on the train set. Each dataset was stored in a separate data frame: TRAIN_SAMPLE (322 observations) and TEST_SAMPLE (140 observations).

The TRAIN_SAMPLE contains 213 deals with conversion outcome Postpone (approximately 66% of the instances) and 109 deals with outcome Won (approximately 34% of the instances) (Figure 18). Hence, this set is imbalanced – there is a skew in the class distribution (Brownlee, 2020), being Postpone the majority class. This bias can lead the algorithm to ignore the minority class (Brownlee, 2020), Won, which is the class we are most interested in correctly predicting.

```
> table(TRAIN_SAMPLE$CONVERSION_OUTCOME)

Postpone       Won
     213       109
> prop.table(table(TRAIN_SAMPLE$CONVERSION_OUTCOME))

 Postpone       Won
0.6614907 0.3385093
```

Figure 18 *– Original training sample.*

To reduce this bias and improve model performance, three random sampling methods were tested on the training sample – over-sampling, under-sampling and a combination of both. For that, the *ovun.sample* function of the *ROSE* package in R was used (Appendix VI). Then, the classifiers were built with an optimal complexity parameter to prune the tree for each training sample. The results obtained are presented in Table 38.

Table 38 *– Performance metrics for different sampling methods.*

Method	Accuracy	Balanced Accuracy	Sensitivity	Specificity
Over-sampling (n=426 instances)	61.43%	63.18%	68.75%	57.61%
Under-sampling (n=218 instances)	52.14%	56.61%	70.83%	42.39%
Over-sampling + Under-sampling (n=322 instances)	56.43%	59.87%	70.83%	48.91%
Original unbalanced sample (n=322 instances)	62.14%	57.74%	43.75%	71.74%

The original sample, without any balancing method, achieved the highest accuracy – 62.14% (Table 38). However, for imbalanced datasets, the balanced accuracy is a more adequate measure, for which the original sample showed one of the worse results (Table 38). Since we are not only interested in having a reasonable accuracy, but also in achieving a good sensitivity (since our main goal is to correctly predict the "positive" class – Won), the sample that

combined both over and under-sampling was chosen as the final training sample for this project.

Hence, the training sample used to build the model was TRAIN_SAMPLE1 (Appendix VI), with 322 instances. This sample size was chosen in order to keep the same dimension of the original training set. This sampling combined over-sampling, which consists of randomly duplicating examples from the minority class (Won), and under-sampling, which randomly deletes examples from the majority class (Postpone) (Brownlee, 2020). Figure 19 shows that TRAIN_SAMPLE1 is a lot more balanced than the original TRAIN_SAMPLE (Figure 18), with each output label representing approximately 50% of the instances.

```
> table(TRAIN_SAMPLE1$CONVERSION_OUTCOME)

Postpone      Won
     160      162
> prop.table(table(TRAIN_SAMPLE1$CONVERSION_OUTCOME))

 Postpone       Won
0.4968944 0.5031056
```

Figure 19 *– Training sample after over and under-sampling.*

Then, the decision tree was built with the *rpart* function in R (Figure 20), which uses the CART algorithm. *Rpart* stands for Recursive Partitioning and Regression Trees (Therneau et al., 2019).

```
# Construction of the model - decison tree
library(rpart)
decision_tree <- rpart(CONVERSION_OUTCOME~.,
                       data = TRAIN_SAMPLE1,
                       method = "class",
                       parms = list(split = 'gini'))
```

Figure 20 *– R code for building the decision tree model.*

There are several components to the *rpart* function (Figure 20):

- The formula – determines that CONVERSION_OUTCOME is the response variable and the remaining columns of the dataset are the predictors.
- The data – the data frame TRAIN_SAMPLE1 contains the data to train the model.

- The method – the response variable, CONVERSION_OUTCOME, is categorical (more specifically, binary), so the method is classification.
- The parameters – the Gini Index was the chosen splitting criterion[9].

For the remaining arguments of the function, the default values were assumed. For instance, there are the controls, which are tuning parameters that allow to pre-prune the tree. These are known as early stopping criteria, since the tree will stop growing when they are met (Das, 2017).

The *minsplit* and *minbucket* are common pre-pruning criteria. The *minsplit* corresponds to the minimum number of instances in a node for a split to be attempted and the *minbucket* corresponds to the minimum number of records in a terminal node (Ashish, 2017). By default, R assumes *minsplit* = 20, such that the tree requires at least 20 observations in a node to split further (Sachdeva, 2020). For *minbucket*, the default value corresponds to *minsplit*/3, which is approximately 7 – there must be at least 7 instances in a leaf node (Sachdeva, 2020). Different variations of these parameters were attempted but the model performance remained constant, so the default values were assumed.

Then, to avoid overfitting in the model and to reduce its complexity, the *printcp* function in R was used to determine the optimal complexity parameter to prune the tree. The output of this function is a table of optimal prunings based on a complexity parameter (Figure 21).

[9] The CART algorithm uses the Gini Index as the splitting criterion. Hence, by default, R assumes this parameter. However, Information Gain could also be chosen in R, by writing 'information' instead of 'gini' in the *rpart* function arguments.

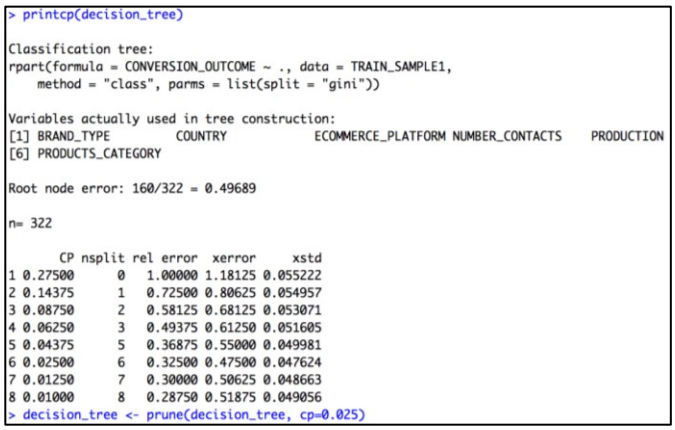


Figure 21 *– Optimal complexity parameter analysis.*

The complexity parameter (cp) controls how the splits are carried out, hence also controlling the size of the tree. The smaller the value of cp, the more branches the tree has – a value of zero would build the tree to its maximum depth. It saves computing time by pruning off unworthy splits – if the cost of adding another variable to the tree is above the value of cp, the tree will stop there (Ashish, 2017).

Figure 21 shows that the number of splits that minimizes the cross validation error (*xerror*) is 6 splits, which corresponds to the optimal cp of 0.025 (Ashish, 2017). Therefore, based on the last command in Figure 21, the tree was pruned with a complexity parameter of 0.025, using the *prune* function in R. With this pruning, we arrived at the final tree, which will now be presented.

5.3.4 Generated Tree Interpretation

Decision trees provide a visual representation that allows us to understand the rules defined by the algorithm. To analyze the constructed tree, the function *fancyRpartPlot* from the *rattle* package in R was used. The command used and the generated tree are presented in Figure 22.

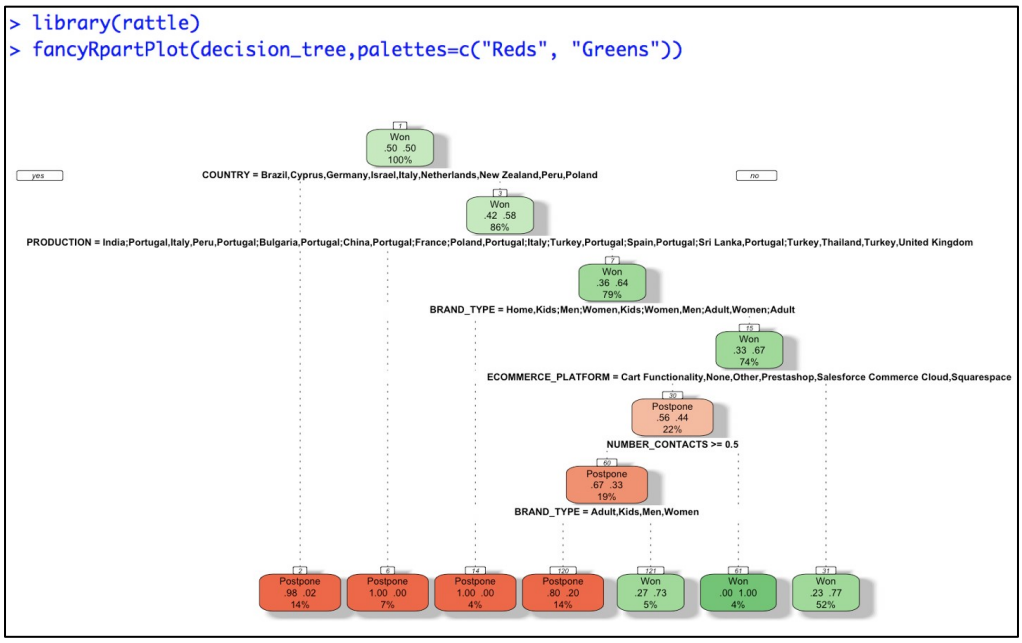


Figure 22 *– R code and output for the decision tree plot.*

The first insight we can extract from Figure 22 is that the algorithm only used five variables to build the tree – COUNTRY, PRODUCTION, BRAND_TYPE, ECOMMERCE_PLATFORM and NUMBER_CONTACTS. This is due to the fact that decision trees have built-in feature selection and the algorithm concluded that the remaining variables of the dataset were irrelevant or redundant for the leads' classification (Gama et al., 2017). One variable, BRAND_TYPE, was used twice to perform splits in the tree.

We can now interpret the tree by looking at Figure 22. The root node (1) contains all of the training set's observations – approximately 50% of the instances have an output label Won and around 50% Postpone. The attribute that performs the first split is COUNTRY. If the origin of the brand is Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland, then we go to a terminal node – this leaf (2) contains 14% of the instances in TRAIN_SET1; 98% of the leaf's instances belong to class Postpone and 2% to class Won. If the answer to the root node is "no", then we move down to the root's child node on the right (3). This node (3) asks about the production of the

brand – if it is located is India and Portugal, Italy, Peru, Portugal and Bulgaria[10], or another place/combination of places shown in that branch, then we move to another leaf node (6) with the class Postpone; otherwise, if the answer is "no", we move to the right node (7), which asks about the brand type. The process continues throughout the various branches following the same logic.

The tree is relevant to understand the rules followed by the algorithm to attribute a classification to each deal. Furthermore, if a member of the Sales team has a deal with certain known characteristics and would like to quickly assess its predicted conversion outcome, he can observe the tree to know which output label would be assigned. Table 39 presents these rules in more detail.

Table 39 *– Rules behind the decision tree algorithm.*

Subset number	Conversion outcome assigned	Content of subset
2	Postpone	Deals from Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland.
6	Postpone	Deals not from Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland, with production made in India and Portugal; Italy; Peru; Portugal and Bulgaria; Portugal and China; Portugal, France and Poland; Portugal, Italy and Turkey; Portugal and Spain; Portugal and Sri Lanka; Portugal and Turkey; Thailand; Turkey or United Kingdom.
14	Postpone	Deals not from Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland, with production not made in India and Portugal; Italy; Peru; Portugal and Bulgaria; Portugal and China; Portugal, France and Poland; Portugal, Italy and Turkey; Portugal and Spain; Portugal and Sri Lanka; Portugal and Turkey; Thailand; Turkey or United Kingdom, with brand type(s) Home; Kids, Men and Women; Kids and Women; Men and Adult or Women and Adult.
120	Postpone	Deals not from Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland, with production not made in India and Portugal; Italy; Peru; Portugal and Bulgaria; Portugal and China; Portugal, France and Poland; Portugal, Italy and Turkey; Portugal and Spain; Portugal and Sri Lanka; Portugal and Turkey; Thailand; Turkey or United Kingdom, with brand type(s) not Home; Kids, Men and Women; Kids and Women; Men and Adult or Women and Adult, with e-commerce platform Cart Functionality, None, Other, Prestashop, Salesforce Commerce Cloud or Squarespace, that have at least 1 contact and brand type is adult; kids; men or women.
121	Won	Deals not from Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland, with production not made in India and Portugal; Italy; Peru; Portugal and Bulgaria; Portugal and China; Portugal, France and Poland; Portugal, Italy and Turkey; Portugal and Spain; Portugal and Sri Lanka; Portugal and Turkey; Thailand; Turkey or United Kingdom, with brand type(s) not Home; Kids, Men and Women; Kids and Women; Men and Adult; Women and Adult; Adult; Kids; Men or Women, with e-commerce platform Cart Functionality, None, Other, Prestashop, Salesforce Commerce Cloud or Squarespace, that have at least 1 contact.
61	Won	Deals not from Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland, with production not made in India and Portugal; Italy; Peru; Portugal and Bulgaria; Portugal and China; Portugal, France and Poland; Portugal, Italy and Turkey; Portugal and Spain; Portugal and Sri Lanka; Portugal and Turkey; Thailand; Turkey or United Kingdom, with brand type(s) not Home; Kids, Men and Women; Kids and Women; Men and Adult or Women and Adult, with e-commerce platform Cart Functionality, None, Other, Prestashop, Salesforce Commerce Cloud or Squarespace, that have 0 contacts.
31	Won	Deals not from Brazil, Cyprus, Germany, Israel, Italy, Netherlands, New Zealand, Peru or Poland, with production not made in India and Portugal; Italy; Peru; Portugal and Bulgaria; Portugal and China; Portugal, France and Poland; Portugal, Italy and Turkey; Portugal and Spain; Portugal and Sri Lanka; Portugal and Turkey; Thailand; Turkey or United Kingdom, with brand type(s) not Home; Kids, Men and Women; Kids and Women; Men and Adult or Women and Adult, with e-commerce platform not Cart Functionality, None, Other, Prestashop, Salesforce Commerce Cloud or Squarespace.

5.3.5 Model Assessment

Lastly, the classifier was used on the TEST_SAMPLE to assess its performance on unseen data. For that, the CONVERSION_OUTCOME of the

[10] When the branch shows ";" it means it is a combination of classes (for instance, India;Portugal means that the production is located in both India and Portugal.

TEST_SAMPLE and the CONVERSION_OUTCOME predicted by the model (*prediction* in Figure 23) were compared.

The function *confusionMatrix* in R was used for this analysis, where the conversion outcome Won was considered as the "positive" class (Figure 23).

```
> library(caret)
> library(e1071)
> prediction <- predict(decision_tree, newdata=data.frame(TEST_SAMPLE[,-c(1)]), type="class")
> confusionMatrix(prediction,TEST_SAMPLE$CONVERSION_OUTCOME,positive='Won')
     Confusion Matrix and Statistics

               Reference
     Prediction Postpone Won
       Postpone       45  14
       Won            47  34

                    Accuracy : 0.5643
                      95% CI : (0.478, 0.6478)
        No Information Rate : 0.6571
        P-Value [Acc > NIR] : 0.9911

                       Kappa : 0.1696

      Mcnemar's Test P-Value : 4.182e-05

                  Sensitivity : 0.7083
                  Specificity : 0.4891
               Pos Pred Value : 0.4198
               Neg Pred Value : 0.7627
                   Prevalence : 0.3429
               Detection Rate : 0.2429
         Detection Prevalence : 0.5786
            Balanced Accuracy : 0.5987

             'Positive' Class : Won
```

Figure 23 *– Confusion matrix and performance metrics for the decision tree model.*

Figure 23 presents the common performance metrics used for assessing classification models. On the top section, there is the confusion matrix, which is the foundation for the other metrics (Hale, 2020a). It is visible that 45 deals with actual conversion outcome Postpone were correctly predicted, as well as 34 deals with actual outcome Won; on the other hand, for 47 deals the predicted outcome was Won, however the output label on the TEST_SAMPLE was Postpone, and 14 actual Won deals were predicted as Postpone.

The accuracy corresponds to (True Positives + True Negatives)/all observations (Hale, 2020a). The accuracy obtained by the model was (34+45)/(45+47+14+34)*100=56.43% – of the 140 instances on the TEST_SET, the model was able to correctly predict the conversion outcome of approximately 56% deals. Moreover, we can state with 95% confidence that the accuracy of the model will range between 47.80% and 64.78%. Usually, to assess the goodness

of the accuracy, it is compared to other model (Hale, 2020a). In this case, if someone guesses the conversion outcome of a deal, there is a 50% probability of correctly guessing. Since the model's accuracy surpassed the 50% threshold, we can conclude that, even though it is not substantial, the model adds some value to this process.

Furthermore, sensitivity and specificity are also relevant measures to analyze. Sensitivity is calculated as True Positives/(True Positives + False Negatives), and it corresponds to the percentage of actual positive cases that the model correctly predicted (Hale, 2020b). The model achieved a sensitivity of 34/(34+14)*100=70.83% – out of the 48 actual positive cases, the model correctly predicted approximately 71%. This metric is especially important when we really want to correctly predict the positive class (Hale, 2020b). This is the case for this project since we are particularly interested in correctly predicting deals Won.

Specificity determines how good the model was at catching the negative cases – it is calculated as True Negatives/(True Negatives + False Positives) (Hale, 2020b). The model achieved a specificity of 45/(45+47)*100=48.91%, meaning that, of the 92 deals with actual outcome Postpone, the model was able to correctly predict approximately 49%. Specificity is especially relevant when it is crucial to correctly predict the negative cases (Hale, 2020b).

Another important metric is the precision (*Pos Pred Value* in Figure 23). Precision is computed as True Positives/(True Positives + False Positives) (Hale, 2020b). The precision of the model was 34/(34+47)*100=41.98% – hence, approximately 42% of the model's positive predictions were correct; the remaining 58% were false positives, which are quite substantial.

Lastly, the balanced accuracy achieved was 59.87% (Figure 23). The balanced accuracy is computed as (sensitivity + specificity)/2 = (70.83%+48.91%)/2 = 59.87%. It corresponds to the average of sensitivity and

specificity, and is a good measure when both metrics are equally important (Hale, 2020c).

Regarding the accuracy, even though it achieved a value of 56.43% for the TEST_SAMPLE, the 95% confidence interval suggests that the accuracy can go up to as high as 64.78%, which is slightly more reasonable. Moreover, for the most relevant metric here, sensitivity, the model achieved a value of approximately 71%, which means it correctly classified more than two thirds of the Won deals. However, it should be noted that the precision obtained was approximately 42%, which denotes there was a large percentage of false positives.

We can further distinct precision and sensitivity by stating that precision tells us how certain we can be of the predicted positives, whilst sensitivity tells how certain we can be that the actual positives are not missed (Sharma, 2019). In this case, we would rather have some extra false positives (deals predicted as Won that are actually Postpone) instead of missing some positive cases (deals with actual outcome Won). Sensitivity is more important in cases where the occurrence of false negatives is not tolerable (Sharma, 2019), which is this case – its preferable to have more false positives then false negatives. Hence, since the model achieved a sensitivity of approximately 71%, we can be confident that it will predict many Won deals correctly and will help the Sales team improve its performance results.

Before concluding the model assessment, an important remark should be noted – the quality of the results and the performance of the model were impacted by the low quality of the data. Firstly, the data contained many missing values, which impacted the training of the model. Secondly, it was visible that some variables, such as BRAND_TYPE, were not always correctly filled in by the Sales team – this also impacted the rules of the decision tree. Lastly, some variables, such as PRODUCTION, allow for many combinations of

classes (countries/continents, in this case), which increased the complexity of the model and the rules used to classify the deals.

To conclude this phase, a comparison between the results achieved by our model and the ones presented in the literature review will be presented. The models proposed in subchapter 3.2 will not be included since the authors used different performance metrics.

Table 40 presents the performance metrics for the Bayesian network, presented in subchapter 3.1, the logistic regression, decision tree, random forest and neural network, presented in subchapter 3.3, and, lastly, our decision tree, in grey.

Table 40 *– Comparison of the models' performance.*

Model	Accuracy	Sensitivity	Specificity	Precision
Bayesian Network (Benhaddou & Leray, 2017)	75%	88%	---	80%
Logistic Regression (Nygård & Mezei, 2020)	59%	77%	58%	70%
Decision Tree (Nygård & Mezei, 2020)	69%	66%	69%	72%
Random Forest (Nygård & Mezei, 2020)	69%	69%	69%	76%
Neural Network (Nygård & Mezei, 2020)	86%	36%	90%	75%
Decision Tree	56%	71%	49%	42%

Table 40 reinforces our previous conclusions – the accuracy of our model was not impressive; in fact, of the models presented, it achieved the worse accuracy. However, in terms of sensitivity, the model compares quite well to the other trials, obtaining the third highest sensitivity presented in Table 40. Furthermore, if we compare our model to the one with the highest accuracy, the neural network, we can see that our sensitivity is almost twice the value of the one for the network. Lastly, in terms of specificity and precision, our model does not show reasonable results – in fact, it has the lowest specificity and precision of the models presented.

This comparison allows to us to understand that, even though our model was not an over-performer, it might be justified by the data quality issues previously mentioned, as other authors have proven that machine learning algorithms have worked quite well for lead scoring.

Lastly, it is important to state that interpretable algorithms, such as decision trees, tend to have worse accuracy than more complex predictive algorithms, such as neural networks. In this project, the interpretability was very important, because one of the goals was that the Sales team could understand the functioning behind the model and its results. This led to other techniques, which could have obtained more promising results, to not be tested in this dissertation.

5.4 Evaluation

In the evaluation phase, the project is reviewed to assess if the desired goals have been achieved and if there are factors that have been overlooked, as well as to determine future actions that may be required. Pete et al. (2000) divide this stage in three tasks: evaluating the results, reviewing the process and determining the next steps.

5.4.1 Evaluation of the Results

It is important to state that, due to time constraints, the lead scoring model has not yet been tested at HUUB. It is expected that this practical experiment would take several months from the application of the model until the gathering of the final results (model's actual accuracy, improvement in conversion rate, etc.). Hence, the evaluation of the results at this stage will focus on what has been achieved so far.

Firstly, it should be considered the degree to which the model was able to meet the business objectives (Pete et al., 2000). As previously seen, the model achieved an accuracy of approximately 56%. Even though this result is not extremely satisfying, there are two important factors to note: 1) the value for sensitivity, which was approximately 71%, was good – this was in fact the main goal of the model, since we do not want to miss deals with conversion outcome

Won; 2) without this model, the prioritization of which brands to contact first would be random, with a probability of correctly predicting the outcome of 50%, so this model would actually add value to this process, as it achieved an accuracy above 50%. The model's classifications would allow the Sales team to objectively and efficiently select the deals that should be prioritized when it comes to the campaigns' activation. Hence, it can be determined that the model achieved the main goal. Nonetheless, the low value of the precision cannot be overlooked – a precision of 42% means that there was a large percentage of false positives in this trial; in a practical context, this may mean that many deals will enter a campaign, because the model predicted a successful conversion, however they will be postponed in the end.

Secondly, it is suspected that these not-so-great results may be due to the low quality of the data. As previously noted, there is a large percentage of missing values, many possible combinations of classes for some variables and incomplete data for some properties (for instance, for BRAND_TYPE, some deals have only been assigned a gender/age category instead of both).

Moreover, Pete et al. (2000) suggest that this phase should also assess other findings unveiled in the project, since they are also an output of the data mining project. Some important findings of this project besides the model include:

- Many properties in HubSpot were actually not used by the Sales team. Some cases include duplicated properties and properties that had been created by employees that no longer work at HUUB. These properties were identified to be deleted.
- The HubSpot deals data contained a lot of missing values – this was an important finding since it highly impacts the quality of several analyses, so it was crucial to enlighten the Sales team on the importance of inputting this data whenever it is available.
- The HubSpot deals data contained a lot of errors – the preliminary analysis showed several manual errors in the data that needed to be corrected. Once

again, it was discussed with the Sales team the importance of avoiding these errors such that the quality of the analyses and data mining results can be the best possible.

- Many properties in HubSpot were not being stored in the appropriate data format, which contributed to several errors in the data. The adequate formats for these properties were identified and updated in HubSpot.
- The preliminary analysis was not only important to get familiarized with the data for building the model, but also allowed to extract insights, detect relevant trends and assess the current conversion indicators – these findings were also an important output of this project.

Based on the results achieved by the model and the findings uncovered during this project, it can be stated that this project brought valuable outputs to the company's sales process and analyses.

5.4.2 Process Review

The process review consists of reviewing the data mining engagement to assess if there are important factors that have been overlooked and to assure the quality of the tasks that were performed (Pete et al., 2000).

Firstly, regarding any factors that may have been overlooked, by the end of this project no such cases had been found. Hence, we can be confident no important factor has been disregarded in this work.

Regarding the quality assurance, there are some important remarks:

- Concerning the attributes used to build the model, there are features, such as ERP, that could have been relevant predictors for the lead scoring model. However, they were excluded in feature selection as the percentage of missing values was extremely high. Perhaps including these features could improve the performance of the model.

- In terms of which attributes to use, we believe this selection was correct since we targeted the properties that will in fact be available in the future after the deals are profiled.
- The training sample used to build the model was small (322 instances). Generally, the larger the training sample, the better the classifier, so the model's results could have been better with a larger data sample.
- Lastly, regarding the model, we believe it was correctly built. However, methods such as bagging and boosting, that can be used to lower the variance of the model, were not attempted.

To conclude this section, it can be stated that, after reviewing the process, the tasks of this project appear to have been well-designed and performed, without any important factors being overlooked. However, there are certainly aspects that could be improved in the future, which will now be presented.

5.4.3 Next Steps

This stage starts by assessing the remaining resources available, since they impact the decision of where to go next (Pete et al., 2000). In this case, the main constraint in terms of resources is time. Hence, this section will present what the next steps of this project would be, if there were no time constrains (i.e., if the project had a longer timespan).

The suggested future steps are:

- Create new simplified properties in HubSpot to replace the attributes that allow for multiple combinations of classes. For instance, instead of PRODUCTION, a new property that defines only the main production location of the brand, instead of all, could be used. Hence, there would no longer be a huge list of possible combinations (e.g. France instead of France;Portugal;Italy). This would decrease the complexity of the model and could improve its performance.

- Create new deal properties in HubSpot that may be more valuable to predict the leads' conversion outcome. Figure 22 showed that only a few attributes were relevant to classify the leads, which means that a lot of the properties stored in HubSpot are irrelevant for our model.
- Collect data for a longer period of time – with a larger training sample, the model could be re-trained, which may improve its performance. Moreover, after a period of time of collecting data, properties such as ERP would likely have a smaller percentage of missing values and could be used as predictors for the model.
- The conversion indicators (qualified rate, proposals acceptance rate and qualified rate) should be assessed in the long term, after the model has been used for a period of time, to evaluate if it has contributed to improvements in those rates. Moreover, the actual accuracy of the model should be analyzed – i.e., have the deals that the model predicted would be Won actually become customers?
- Lastly, in the future, other algorithms could be tested to determine if they achieve better results than a decision tree for predicting the deals' conversion outcome. For instance, a random forest could be attempted, since some authors suggest that random forests tend to achieve better classification performance than decision trees, especially for large datasets (Ali et al., 2012). Nonetheless, for this task, it is important to determine if the assumptions of other models attempted apply to this data; for instance, the Naïve-Bayes algorithm assumes that all numeric variables have a normal distribution, which does not apply to our data.

5.5 Deployment

Finally, there is the deployment, the last stage of the CRISP-DM method. This subchapter is dedicated to determining how the monitoring and

maintenance of the model and data mining results would be conducted after the deployment of the project.

Two teams at HUUB are involved in this project: Data Insights & Analytics and Sales teams. For the Sales team, some suggestions were developed to ensure the model achieves good results:

- Firstly, for the model to work properly, the data has to be adequately inserted in HubSpot by the Sales team. This means that the team members must be careful when manually inserting the data, to avoid errors that may impact the leads' classification.
- Secondly, the Sales team must make an effort to avoid contributing to missing values, by inserting data whenever that information is available. This will contribute to the accuracy of the model's predictions.

On the other hand, the Data Insights & Analytics team is in charge of regularly analyzing the deals' data to assure its quality and assess the data mining results. Hence, this team would be responsible for:

- Regularly monitoring the deals data to detect errors, outliers and missing values, that may need to be corrected. This should be done on a weekly basis, since every week new campaigns are activated.
- Assessing the accuracy of the model after some months of it being used to classify deals. There needs to be a reasonable number of brands that have gone through the process of conversion before the model's performance is assessed. For this evaluation, the metrics presented in subchapter 5.3.5 could be computed, especially the accuracy, sensitivity and precision.

Lastly, there needs to be some criteria to determine when the model and its results should stop being used for decision making. In this case, a threshold for the overall accuracy of the model could be set. For the first trial, this value could be set to 50%, such that if the model obtained an accuracy lower than that, it would not be used anymore. This threshold seems adequate since an accuracy above 50% means that the model adds some value to the process.

Sixth Chapter

6. Conclusion

This dissertation focuses on studying how machine learning can be used to build a lead scoring model for a B2B company. For that, past data of converted leads was extracted from the CRM platform of a logistics company, HUUB, and a decision tree was built, using the past data as training data for the CART algorithm, to predict each lead's conversion outcome (Won or Postpone).

The following subchapters are dedicated to presenting the main conclusions and contributions of this dissertation, as well as identifying some suggestions for future work regarding this theme.

6.1 Final Considerations

The assessment of the lead scoring models presented in the literature led to the conclusion that machine learning and historical data can be combined to develop automated lead scoring models that work quite well in practice. However, in comparison to the ones presented in the literature review, our model did not achieve impressive results for most performance metrics – the accuracy, specificity and precision of our model were the worse out of the ones presented. However, the sensitivity achieved by our model was very good and compared quite well to the literature's results, which suggests it would be a good predictor for the positive cases (deals Won) and, hence, a good performer, considering our main goal of prioritizing these deals.

This dissertation proposed to build an automated lead scoring model for a logistics start-up, HUUB. HUUB's Sales team contacts 60 brands each week, which are randomly selected by the team members. Hence, HUUB was in a need of a solution to efficiently and adequately perform contact prioritization

before the campaigns are activated. Moreover, the company is expanding, and, hence, relying on automation and technology to scale up the business. In this context, this project proposed to build a model to help HUUB's Sales team prioritize the brands that have a high probability of becoming customers.

Formally, the research question answered by this dissertation is: *How can a B2B company automate the process of lead scoring?.* At this time, and after assessing the results presented in Chapter 5, we can state that a company can automate this process by using data from past converted leads, which are usually stored in CRM systems, to train supervised learning algorithms; the resultant lead scoring model provides relevant outputs for contact prioritization. Both the results achieved in this project and the ones extracted from the literature support this hypothesis, so we can be confident that this is a viable approach.

The results achieved can be summarized into two main conclusions:

- Firstly, it can be stated that, even though the performance of our model was not astonishing, it still adds value to the prioritization of contacts. This is not only justified by a high value for the sensitivity, but also by an accuracy over 50%. Hence, this seems to be a feasible approach to automate the process of lead scoring, since it still adds value to the company's sales process.
- Secondly, the fact that our results were not great compared to the literature may be justified by the following issues: i) interpretable algorithms tend to have worse accuracy than more complex algorithms, ii) the data used to train the model contains many missing values, iii) some of the variables used in the model allow for several class combinations, iv) some properties revealed incomplete data for several instances, v) some attributes, that could have been relevant predictors, had to be excluded in feature selection because of a high percentage of missing values and, lastly, vi) the training sample used to build the model was quite small. It is plausible that these factors may have impacted the model's results, leading it to have a worse performance than the literature's models.

Taking the aforementioned conclusions into consideration, it can be stated that this dissertation supports the conclusion of the literature review section, which is that machine learning can be used to develop automated lead scoring models. However, it should be noted that there are several factors, especially related to the data used, that may impact the performance of the model.

Lastly, we can briefly summarize the goals that were achieved during this project, which were the following:

- Identify relevant trends and insights from past converted leads.
- Identify which attributes are relevant to predict lead conversion at an early stage of the sales process.
- Develop a lead scoring model that allows the Sales team to adequately perform contact prioritization.

6.2 Main Contributions

This work provides three major contributions. Firstly, for the academic context, as it presents a study that diminishes a relevant research gap – the literature does not have yet many studies concerning the use of machine learning models for automating the lead scoring process in companies.

Secondly, for companies – an organization that desires to develop an automated lead scoring model can follow the steps presented in the case study section of this dissertation.

Lastly, for HUUB – this work provided great findings for the company, that were uncovered throughout the project, such as: i) many HubSpot deal properties were not used by the Sales team, so they could be deleted, ii) the deals data contained many missing values and errors, so the Sales team was enlightened on the importance of correctly inputting this data, iii) many HubSpot properties were not stored in the appropriate format, so they were corrected, and, lastly, iv) the preliminary analysis provided insights over

important sales indicators and trends, which were extremely relevant for HUUB.

6.3 Future Work

This section summarizes the suggested future steps for this project at HUUB, which include the following tasks:

- Create properties in HubSpot that simplify the attributes that allow for multiple combinations of classes, to decrease the complexity of the model and perhaps improve its performance. Moreover, assess if there are relevant predictors for lead conversion that are not in HubSpot and create them.
- Collect data for a longer period of time in order to re-train the model with a larger training sample.
- Assess the improvements in the conversion indicators after the model has been used for several months.
- Lastly, test other algorithms to determine if they are better performers than the decision tree in predicting the leads' conversion outcome.

Lastly, there is also a suggestion for future research on this thematic. That recommendation would be to compare the results of manual versus automated lead scoring in companies, using data from a real company, to assess if in fact machine learning offers a superior alternative to the predominant manual approach. Most studies only compare the two options theoretically, so it may be relevant to compare them with actual results from a practical experiment.

Bibliographic References

Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random Forests and Decision Trees. *International Journal of Computer Science Issues*, *9*(5), 272–278.

Ashish, R. (2017). *Tuning Parameters Of Decision Tree Models*. Urban Pro. https://www.urbanpro.com/data-science/tuning-parameters-of-decision-tree-models

Azevedo, A., & Santos, M. F. (2008). KDD , SEMMA and CRISP-DM: A Parallel Overview. *IADIS European Conference Data Mining*, 182–185. http://recipp.ipp.pt/handle/10400.22/136%0Ahttp://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf

Benhaddou, Y., & Leray, P. (2017). Customer Relationship Management and Small Data - Application of Bayesian Network Elicitation techniques for building a Lead scoring model. *Proceedings of the 14th International Conference on Computer Systems and Applications*, 251–255. https://doi.org/10.1109/AICCSA.2017.51

Bošnjak, Z., Grljević, O., & Bošnjak, S. (2009). CRISP-DM as a framework for discovering knowledge in small and medium sized enterprises' data. *Proceedings - 5th International Symposium on Applied Computational Intelligence and Informatics*, *114*, 509–514. https://doi.org/10.1109/SACI.2009.5136302

Bouchon-Meunier, B., Coletti, G., & Yager, R. (2006). *Modern Information Processing From Theory to Applications*. Elsevier Science. https://doi.org/https://doi.org/10.1016/B978-0-444-52075-3.X5000-6

Bradford, W. R., Johnston, W. J., & Bellenger, D. (2016). The Impact of Sales Effort on Lead Conversion Cycle Time in a Business-to-Business Opportunity Pipeline. *Proceedings of the International Engaged Management Scholarship Conference*, 1–21. https://doi.org/10.2139/ssrn.2866954

Brownlee, J. (2020). *Random Oversampling and Undersampling for Imbalanced Classification*. Machine Learning Mastery. https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

Cox, L. (2019). *Lead Scoring 101: How to Use Data to Calculate a Basic Lead Score*. HubSpot. https://blog.hubspot.com/marketing/lead-scoring-instructions?utm_campaign=Pillar Pages %7C CS Team %7C 2018&utm_source=youtube&utm_medium=social

Dakouan, C., Benabdelouahed, R., & Anabir, H. (2019). Inbound Marketing vs. Outbound Marketing: Independent or Complementary Strategies. *Expert Journal of Marketing*, *7*(1), 1–6.

Das, S. (2017). *Decision Trees and Pruning in R*. DZone. https://dzone.com/articles/decision-trees-and-pruning-in-r

DiGangi, E. A., & Moore, M. K. (2013). *Research Methods in Human Skeletal Biology*. Elsevier. https://doi.org/https://doi.org/10.1016/C2010-0-65850-0

Doyle, C. (2016). *A Dictionary of Marketing*. Oxford University Press. https://doi.org/10.1093/acref/9780198736424.001.0001

Duda, R., Hart, P., & Stork, D. (2001). *Pattern Classification* (2nd Editio). Wiley-Interscience.

Duggal, N. (2020). *Artificial Intelligence Career Guide: A Comprehensive Playbook to Becoming an AI Expert*. https://www.simplilearn.com/artificial-intelligence-career-guide-pdf

Duncan, B., & Elkan, C. (2015). Probabilistic Modeling of a Sales Funnel to Prioritize Leads. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1751–1758. https://doi.org/10.1145/2783258.2788578

Fenton, N. E., & Neil, M. (2019). An Extension to the Noisy-OR Function to Resolve the "Explaining Away" Deficiency for Practical Bayesian Network Problems. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12),

2441–2445. https://doi.org/10.1109/TKDE.2019.2891680

Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Gama, J., Carvalho, A., Faceli, K., Lorena, A., & Oliveira, M. (2017). *Extração de Conhecimento de Dados* (3ª Edição). Edições Sílabo.

Gao, H., Yang, V., Jin, Y., & Rai, A. (2019). *Predictive Analytics for Leads Generation and Engagement Recommendations*. https://patentimages.storage.googleapis.com/b8/c5/45/451e8e59eb60fb/US20190378149A1.pdf

Giacomele, S. (2015). *Lead Nurturing*. Pmweb. https://blog.pmweb.com.br/o-que-e-lead-nurturing/

Gupta, P. (2017). *Decision Trees in Machine Learning*. Towards Data Science. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

Hale, J. (2020a). *Classification Metrics Everyone Should Know*. Towards Data Science. https://towardsdatascience.com/classification-metrics-everyone-should-know-b67fd0044c0c

Hale, J. (2020b). *The 3 Most Important Basic Classification Metrics*. Towards Data Science. https://towardsdatascience.com/the-3-most-important-basic-classification-metrics-3368dd425f74

Hale, J. (2020c). *The 3 Most Important Composite Classification Metrics*. Towards Data Science. https://towardsdatascience.com/the-3-most-important-composite-classification-metrics-b1f2d886dc7b

Hardesty, L. (2017). *Explained: Neural networks*. Massachusetts Institute of Tecnhology News. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

HubSpot. (2021). *Property Field Types in HubSpot*. https://knowledge.hubspot.com/contacts/property-field-types-in-hubspot

Kotu, V., & Deshpande, B. (2015). *Predictive Analytics and Data Mining* (1st Editio). Morgan Kaufmann Publishers.

Kotu, V., & Deshpande, B. (2019). *Data Science - Concepts and Practice*. Morgan Kaufmann Publishers. https://doi.org/https://doi.org/10.1016/C2017-0-02113-4

Kurama, Vi. (2020). *Gradient Boosting In Classification: Not a Black Box Anymore*. Paperspace Blog. https://blog.paperspace.com/gradient-boosting-for-classification/

Lindahl, E. (2017). *A qualitative examination of lead scoring in B2B marketing automation, with a recommendation for its practice.* CSC School of Computer Science and Communication.

Marion, G. (2016). *Lead Scoring is Broken. Here's What to Do Instead.* Autopilot. https://medium.com/marketing-on-autopilot/lead-scoring-is-broken-here-s-what-to-do-instead-194a0696b8a3

McCraw, C. (2020). *What is Predictive Lead Scoring: Features, Benefits, Top Tools*. Get Voip. https://getvoip.com/blog/2020/09/01/predictive-lead-scoring/#predictiveLeadScoring

McGlaughlin, F., Doyle, J., & Bortone, B. (2012). *Lead Generation Benchmark Report*.

Niemi, A. (2017). *Digital Lead Generation and Nurturing: A Holistic Approach*. Aalto University.

Nygård, R., & Mezei, J. (2020). Automating Lead Scoring with Machine Learning: An Experimental Study. *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 1439–1448. https://doi.org/10.24251/hicss.2020.177

Patterson, L. (2007). Marketing and sales alignment for improved effectiveness. *Journal of Digital Asset Management*, *3*(4), 185–189. https://doi.org/10.1057/palgrave.dam.3650089

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Elsevier Inc.

Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An Introduction to Logistic Regression Analysis and Reporting. *The Journal of Educational Research*, *96*(1), 3–14. https://doi.org/10.1080/00220670209598786

Pete, C., Julian, C., Randy, K., Thomas, K., Thomas, R., Colin, S., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS.

Peterson, R. M., Rodriguez, M., & Krishnan, V. (2011). CRM and Sales Pipeline Management: Empirical Results for Managing Opportunities. *The Marketing Management Journal*, *21*(1), 60–70.

Prog, L. (2011). *A History of Bayes' Theorem*. Less Wrong. https://www.lesswrong.com/posts/RTt59BtFLqQbsSiqd/a-history-of-bayes-theorem

Raileanu, L., & Stoffel, K. (2004). Theoretical Comparison between the Gini Index and Information Gain Criteria. *Annals of Mathematics and Artificial Intelligence*, *41*(1), 77–93. https://doi.org/10.1023/B:AMAI.0000018580.96245.c6

Ranganathan, S., Gribskov, M., Nakai, K., & Schönbach, C. (2019). *Encyclopedia of Bioinformatics and Computational Biology*. Elsevier.

Reinders, C., Yang, M. Y., Ackermann, H., & Rosenhahn, B. (2019). *Multimodal Scene Understanding*. Elsevier Inc. https://doi.org/https://doi.org/10.1016/B978-0-12-817358-9.00010-X

Rosenbröijer, C.-J. (2014). Customer Relationship Management and Business Analytics: a Lead Nurturing Approach. *DYNAA 2014*, *5*(1), 29–34. https://doi.org/10.5772/30551

Sachdeva, J. (2020). *Minsplit and Minbucket*. Talking with Data. https://medium.com/talking-with-data/minsplit-and-minbucket-a49ff56026c8

Salesforce. (2014). *Marketing Automation for Sales Playbook*. https://www.pardot.com/whitepapers/marketing-automation-sales-playbook/

Shafique, U., & Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD , CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), 217–222.

Sharma, P. (2019). *Decoding the Confusion Matrix*. Towards Data Science. https://towardsdatascience.com/decoding-the-confusion-matrix-bb4801decbb

Silva, R. (2015). *Bayesian networks and the search for causality*. Slideshare. https://www.slideshare.net/BayesNetsMeetupLondon/bayesian-networks-and-the-search-for-causality

Soni, D. (2018). *Supervised vs. Unsupervised Learning*. Towards Data Science. https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Taktak, A. F. G., & Fisher, A. C. (2007). *Outcome Prediction in Cancer*. Elsevier. https://doi.org/https://doi.org/10.1016/B978-0-444-52855-1.X5000-4

Tangirala, S. (2020). Evaluating the impact of Gini Index and Information Gain on Classification using Decision Tree Classifier Algorithm. *International Journal of Advanced Computer Science and Applications*, *11*(2), 612–619. https://doi.org/10.14569/ijacsa.2020.0110277

Therneau, T., Atkinson, B., & Ripley, B. (2019). Recursive Partitioning and Regression Trees. In *CRAN R Project*. https://cran.r-project.org/package=rpart

Todor, R. (2016). Marketing Automation. *Bulletin of the Transilvania University of Braşov*, *9*(58), 87–94. https://doi.org/10.4324/9781315764597-12

Tyagi, N. (2020). *Understanding the Gini Index and Information Gain in Decision Trees*. Medium. https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8

Venkatesan, S., Thakur, T., Venkatarathinam, V., Goud, B., & Ramasamy, P. (2018). *System and Method for Identification and Prediction of Positive Business Leads through Lead Scoring*.

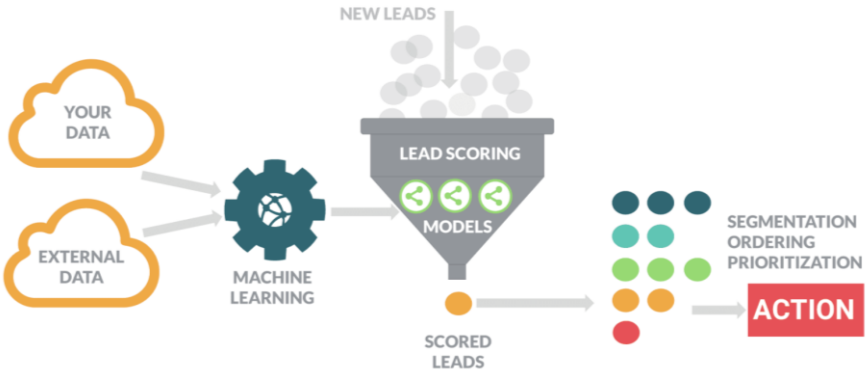
Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 29–39. http://www.ijisr.issr-journals.org/

Attachments

Attachment I – Manual lead scoring matrix[11].

Activity	Points
Form/Landing Page Submission	+ 5
Submitted "Contact Me" Form	+25
Received an Email	0
Email Open	+1
Email Clickthrough	+3
Registered for Webinar	+3
Attended Webinar	+10
Downloaded a Document	+5
Visited a Landing Page	+2
Unsubscribed from Newsletter	-2
Watched a Demo	+8
Contact is a CXO	+5
Visited Trade Show Booth	+3
Visited Pricing Page	+10

Attachment II – Lead scoring automation[12].



[11] Source: Marion (2016)
[12] Source: McCraw (2020)

Appendixes

Appendix I – Methodology overview.

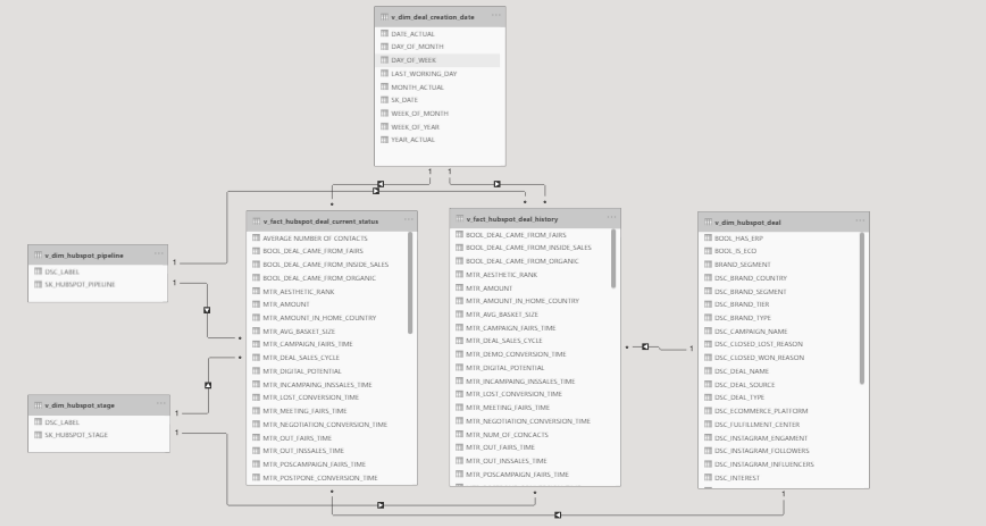
CRISP-DM phase	Tasks performed	Tools used
Business Understanding	- Meet with HUUB's Sales team to gather information on the company's Inside Sales process. - Study HUUB's Inside Sales framework to identify which needs the lead scoring model should satisfy.	
Data Understanding & Data Preparation	- Study HubSpot (CRM platform) to get familiarized with its functioning and the data it stores. - Analyze HubSpot deal properties: properties' description, data format and percentage of missing values. - Study HubSpot's field types to identify properties that are not stored in the adequate format and update those properties' format. - Identify relevant deal properties that are available immediately after the profiling stage to use as predictors in the model.	HubSpot
	- Develop the data necessities: design fact and dimension tables for the required data. - Integrate the HubSpot's required data with HUUB's data warehouse.	Google BigQuery
	- Create a dataflow in Microsoft Power BI for the HubSpot's data. - Perform a preliminary data analysis of the relevant deal properties using the dataflow as the source of data. - Develop a report in Power BI with relevant insights of the deals data. - Identify errors in the data.	Power BI
	- Perform data cleaning of aforementioned errors directly in HubSpot to ensure data quality for next steps and future analyses.	HubSpot
	- Perform data sampling of deals with past conversion outcome Won or Postpone, to obtain the final dataset. - Add new attribute to the dataset – Conversion outcome (Won or Postpone).	Excel Power BI
Modeling	- Choose a machine learning algorithm for the classification problem: the choice is a decision tree due to its interpretability. - Import and analyze final dataset in RStudio. - Split the dataset into a training and test sample with a 70%/30% distribution, using a stratified random sampling. - Test three random sampling methods on the training sample to improve the class imbalance: over-sampling, under-sampling and a combination of both. The final sample, the best performer, combines both methods. - Build a decision tree for classifying the deals' conversion outcome by applying the CART algorithm to the training sample. - Prune the tree with an optimal complexity parameter that minimizes the cross-validation error, to avoid over-fitting. - Present the generated tree and interpret the rules defined by the algorithm. - Assess the model's performance using the confusion matrix-based metrics – accuracy, sensitivity, specificity and precision.	RStudio
Evaluation	- Evaluate the project's results: model performance and other valuable findings of the project. - Review the process to assess the quality of the tasks performed. - Identify the steps that could be performed in the future to improve the results of the project.	
Deployment	- Define the maintenance and monitoring strategy of the data mining results: definition of tasks assigned to Sales and Data Insights & Analytics teams.	

Appendix II – HubSpot deal properties.

HubSpot Field Name	HubSpot Field Type	Open / Closed / Automatic Field	Description	% of Missing Values
Last Modified Date	Date picker	Automatically generated by HubSpot	Most recent timestamp of any property update for the deal.	0,0%
Create Date	Date picker	Automatically generated by HubSpot	Date of when the deal was created.	0,0%
Deal ID	Number field	Automatically generated by HubSpot	Unique ID for each deal.	0,0%
Number of Contacts	Number field	Automatically generated by HubSpot	Number of contacts associated with the deal.	0,0%
Deal Name	Single-line text	Open field	Name of the deal.	0,0%
Deal Stage	Radio select	Closed field with 25 categories: Qualified, Demo, Pre-proposal, Proposal, Negotiation, Won, Clients, Postpone, Lost, Profiling, In Campaign, Qualifying, Out, Pre-Campaign, Meeting, Pos Campaign, Welcome to HUUB, STEP 1, STEP 2, Ready to Go, Done, Early Birds, Active, Previous Years, Churned	The stage the deals is in - allows to categorize and track the progress of the deals.	0,0%
Pipeline	Dropdown select	Closed field with 8 categories: Conversion, Inside Sales, Fairs, Organic, Onboarding, PT Inside Sales, My Brands, Nurturing	The pipeline the deal is in - determines which stages are options for the deal.	0,0%
Original Source Type	Dropdown select	Closed field with 9 categories: Organic search, Paid search, Email marketing, Social media, Referrals, Other campaigns, Direct traffic, Offline sources, Paid social	Original source for the contact with the earliest activity for this deal.	0,9%
Product's Category	Multiple checkboxes	Closed field with 7 categories: Apparel, Footwear, Swimwear, Homewear, Underwear, Accessories, Other	Details the type of product for the deal.	2,1%
Country	Single-line text	Open field	Country of origin of the deal.	2,4%
Original Source Data 1	Single-line text	Open field	Additional information about the original source for the associated contact, or associated company if there is no contact.	4,1%
Brand Type	Multiple checkboxes	Closed field with 6 categories: Kids, Men, Women, Home, Other, Adult	Product categories sold by the brand.	6,0%
Sales Channels	Multiple checkboxes	Closed field with 4 categories: Wholesale, Ecommerce, Own Stores, Marketplace	Sales channels used by the brand.	6,1%
Original Source Data 2	Single-line text	Open field	Additional information about the original source for the associated contact, or associated company if there is no contact.	6,2%
Deal Owner	HubSpot user	Closed field: HubSpot users	The HubSpot user that the deal is assigned to.	8,9%
Last Activity Date	Date picker	Automatically generated by HubSpot	The last time a note, call, email, meeting, or task was logged for a deal.	12,2%
Last Contacted	Date picker	Automatically generated by HubSpot	The last time a call, sales email, or meeting was logged for this deal.	13,6%
Deal Source	Dropdown select	Closed field with 5 categories: Cold Contact, Fair, Organic, Referral, Pipespike	Source of the deal.	20,4%
Followers	Single-line text	Open field	Number of Instagram followers of the deal.	21,8%
E-commerce Platform	Dropdown select	Closed field with 65 categories: Drupal Commerce, Shopify, WooCommerce, Pretashop, Magento, IBM Webshpere Commerce, Intershop, Squarespace, Wix, Magnolia CMS, OneStop Internet, DynamicWeb, MyStore NO, Net-a-Porter, Other, None, In Development, Not Identified, Salesforce Commerce Cloud, Cart Functionality, Ecwid, LogiCommerce, Global-e, OXID Eshop, Demandware, Visualsoft, IAI SHOP, Shoplo, Edrone, Litium, OpenCart, Shoper, nopCommerce, Big Cartel, blubol, CorvusPay, textalk, xt Commerce, Sylius, JTL-Shop, Shopblocks, NET2B, SupaDupa, Snipcart, Stylehub, GetShopped, Vendd, Hybris, Shopsoftware, T-Soft, Gambio, JetShop, osCommerce, Versa Commerce, SmartWeb, VirtueMart, Shopware, PlentyMarkets, CS Cart, Shoprocket, BigCommerce, eShop, Jumpseller, Tiendy, Redicom	E-commerce platform used by the brand.	29,8%
Brand Tier	Radio select	Closed field with 5 categories: S (<40k), M (40k-100k), L (100k-200k), XL (200k-500k), XXL (>500k)	Brand segmentation by ARR. Sales classification that adjusts a tier to get a brand's weight by its expected revenue.	36,2%
Brand Segment	Multiple checkboxes	Closed field with 4 categories: Luxury (>500€), Premium (150-500), Value (50-150), Low Cost (<50€)	Segmentats the brands by the average price of their products.	37,9%
Production	Single-line text	Open field	Country(ies) where the brand's production is located.	39,9%
Campaign Name	Single-line text	Open field	Name given to the campaign related to the deal.	40,3%

HubSpot Field Name	HubSpot Field Type	Open / Closed / Automatic Field	Description	% of Missing Values
Postpone Reason	Multiple checkboxes	Closed field with 15 categories: Brand is considering the investment costs, Operational pending developments, Product pending developments, Brand is relutant to change, Brand is handling logistics inhouse, Brand does not fit HUUB, Shipping Costs are too high, Logistic Costs are too high, Brand did not reply, Operations does not fill brand needs, Product does not fill brand needs, Brand does not want to outsource to foreign parties, Timing is out, Changes in Brand's business, Bounced Email	Reason(s) why the brand did not accept HUUB's offer.	40,7%
Close Date	Date picker	Automatically generated by HubSpot	The date the deal was closed.	48,6%
Profiling Source	Multiple checkboxes	Closed field with 8 categories: Database, Paid Database, Instagram, Other Social Media, Blog, Web, Offline, N.D.	Source used to profile of the deal.	54,4%
Tradeshow	Multiple checkboxes	Closed field with 23 categories: Playtime Paris, CIFF Copenhagen, Pitti Bimbo, Kind + Jugend, FIMI Madrid, Supreme Kids Munich, SEEK Berlin, Who's Next Paris, Pure London, Revolver Copenhagen, LondonEdge, Supreme Women&Men Munich, Selvedge Run, Neonyt, Jacket Required, Man / Woman, Modtissimo, N.D., Season AW, Season SS, Pitti Uomo, Premium Berlin, SIL Paris	Tradeshow(s) the brand is attending.	57,0%
Instagram Influencers	Multiple checkboxes	Closed field with 4 categories: Mega, Macro, Mid-Tier, Micro, Nano, None	Ranking of the most popular influencers mentioning the brand.	57,2%
Instagram Engagement	Multiple checkboxes	Closed field with 4 categories: Very High, High, Average, Low	Instagram engagement of the brand.	58,8%
Instagram Followers	Multiple checkboxes	Closed field with 5 categories: Nano, Micro, Mid-Tier, Macro, Mega	Converts the number of followers in a category, from nano to mega brands.	58,8%
Digital Potential	Number field	Open field	Number from 0 to 10 calculated by the combination of followers, engagement and influencers on instagram, as well as brand's aesthetic.	59,3%
Aesthetic Rank	Number field	Open field	Sensitive analysis based on product/branding relevance.	59,6%
Amount	Number field	Open field	Annual recurring revenue of the deal - how much it is expected the brand will provide HUUB in a year.	66,6%
Amount in company currency	Calculation	Automatically generated by HubSpot	Amount of the deal, using the exchange rate, in HUUB's currency.	66,6%
Deal Description	Multi-line text	Open field	Brief description of the deal.	70,3%
Referral	Dropdown select	Closed field with 4 categories: Client, Partner, Team, N.D.	Who referred HUUB to the deal	72,2%
Main Markets	Multiple checkboxes	Closed field with 20 categories: Portugal, Spain, Germany, Europe, USA & Canada, Asia, Middle East, Africa & ROW, Switzerland, Sweden, Belgium, Turkey, Italy, Greece, Hungary, Austria, Denmark, Netherlands, Finland, Central & South America	Markets that are essential for company's operation and market positioning - where the brand exports to.	75,8%
ECO	Single checkbox	Closed field with 2 categories: Yes, No	Brand's mission to be environmentally sustainable.	80,8%
N° Items / Season	Single-line text	Open field	Expected number of items entering HUUB's warehouse per season.	82,5%
Interest	Radio select	Closed field with 3 categories: Not Interested, Interested, Very Interested	Level of Interest.	86,9%
Start Date	Date picker	Open field	Date of when the brand will start its operation with HUUB.	87,2%
Price p/ Item	Number field	Open field	Price per item being charged to the brand for all of the inbound and outbound activities accoring to the sales channels used by the brand.	91,8%
Barcodes	Single checkbox	Closed field with 2 categories: Yes, No	Determines if the brand's products have barcodes with 13-digits EANs, which is crucial to work with HUUB.	93,7%
Fulfillment Center	Multiple checkboxes	Closed field with 3 categories: HUUB Portugal, Agility Portugal, Maersk Netherlands	Fulfillment centers used by the brand.	97,7%
Won Reason	Multiple checkboxes	Closed field with 11 categories: Better logistic costs, Better shipping costs, Costs predictability & Simplicity, Proximity to production, End-to-end supply chain solution, Operation fills brands needs, Product fills brand needs, Visibility over the operation, Focus on core activities, Positive Referral, Service Levels fit brand needs	Details the value propostions that most atracted the customer to accept the deal	97,7%
Sales Owner	HubSpot user	Closed field: HubSpot users	The sales person who won this deal.	98,0%
Loss Reason	Multiple checkboxes	Closed field with 8 categories: High Logistic costs, High Shipping costs, Insufficient Service Levels, Product limitation, Legal context limitation, Negative referral, Changes in Deal's business during negotiation, Distance to production	Reason why the deal was lost.	99,1%
Next Activity Date	Date picker	Automatically generated by HubSpot	The date of the next upcoming activity for a deal. This includes logging a future call, sales email, or meeting using the Log feature, as well as creating a future task or scheduling a future meeting.	99,4%
Basket Size	Number field	Open field	Determines the average basket size of a brand eCommerce order.	99,7%
ERP	Radio select	Closed field with 3 categories: Yes, No, Unknown	Defines if the brand has an ERP system.	99,9%

Appendix III – Data warehouse star schema.



Appendix IV – Data import and understanding in R.

```
# Import deals dataset into a data frame and check summary of the data
library(readxl)
DF_DEALS_DATA <- data.frame(read_excel("DATASET_R_DT.xlsx",col_names=TRUE))

> summary(DF_DEALS_DATA)
 CONVERSION_OUTCOME BRAND_SEGMENT      ECOMMERCE_PLATFORM     ECO              COUNTRY
 Length:462         Length:462         Length:462         Length:462         Length:462
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character



 INSTAGRAM_FOLLOWERS INSTAGRAM_ENGAGEMENT INSTAGRAM_INFLUENCERS PRODUCTS_CATEGORY  SALES_CHANNELS
 Length:462          Length:462           Length:462            Length:462         Length:462
 Class :character    Class :character     Class :character      Class :character   Class :character
 Mode  :character    Mode  :character     Mode  :character      Mode  :character   Mode  :character



  BRAND_TYPE         PRODUCTION        NUMBER_CONTACTS  AESTHETIC_RANK
 Length:462         Length:462         Min.   : 0.000   Min.   : 1.000
 Class :character   Class :character   1st Qu.: 1.000   1st Qu.: 4.000
 Mode  :character   Mode  :character   Median : 1.000   Median : 6.000
                                       Mean   : 1.552   Mean   : 5.982
                                       3rd Qu.: 2.000   3rd Qu.: 7.000
                                       Max.   :22.000   Max.   :10.000
                                                        NA's   :186

# Convert characters to factors
DF_DEALS_DATA$CONVERSION_OUTCOME <- as.factor(DF_DEALS_DATA$CONVERSION_OUTCOME)
DF_DEALS_DATA$BRAND_SEGMENT <- as.factor(DF_DEALS_DATA$BRAND_SEGMENT)
DF_DEALS_DATA$ECOMMERCE_PLATFORM <- as.factor(DF_DEALS_DATA$ECOMMERCE_PLATFORM)
DF_DEALS_DATA$ECO <- as.factor(DF_DEALS_DATA$ECO)
DF_DEALS_DATA$COUNTRY <- as.factor(DF_DEALS_DATA$COUNTRY)
DF_DEALS_DATA$INSTAGRAM_FOLLOWERS <- as.factor(DF_DEALS_DATA$INSTAGRAM_FOLLOWERS)
DF_DEALS_DATA$INSTAGRAM_ENGAGEMENT <- as.factor(DF_DEALS_DATA$INSTAGRAM_ENGAGEMENT)
DF_DEALS_DATA$INSTAGRAM_INFLUENCERS <- as.factor(DF_DEALS_DATA$INSTAGRAM_INFLUENCERS)
DF_DEALS_DATA$PRODUCTS_CATEGORY <- as.factor(DF_DEALS_DATA$PRODUCTS_CATEGORY)
DF_DEALS_DATA$SALES_CHANNELS <- as.factor(DF_DEALS_DATA$SALES_CHANNELS)
DF_DEALS_DATA$BRAND_TYPE <- as.factor(DF_DEALS_DATA$BRAND_TYPE)
DF_DEALS_DATA$PRODUCTION <- as.factor(DF_DEALS_DATA$PRODUCTION)
```

Appendix V – Stratified random sampling in R.

```
# Creating test and train sample with stratified random sampling
set.seed(7)
p = 0.30

TRAIN_SAMPLE <- data.frame()
TEST_SAMPLE <- data.frame()

for(i in levels(DF_DEALS_DATA$CONVERSION_OUTCOME)) {
  dsub <- DF_DEALS_DATA[DF_DEALS_DATA$CONVERSION_OUTCOME == i,]
  B=ceiling(nrow(dsub)*p)
  C=sample(1:nrow(dsub),B)
  dsub_1 <- dsub[C,]
  dsub_2 <- dsub[-C,]
  TEST_SAMPLE <- rbind(TEST_SAMPLE, dsub_1)
  TRAIN_SAMPLE <- rbind(TRAIN_SAMPLE, dsub_2)}
```

Appendix VI – Balancing the training sample with three different methods in R.

```
# Balancing the training sample
library(ROSE)
TRAIN_SAMPLE1<-ovun.sample(CONVERSION_OUTCOME~.,data=TRAIN_SAMPLE,method="both",na.action=na.pass,N=322,seed=6)$data
TRAIN_SAMPLE2<-ovun.sample(CONVERSION_OUTCOME~.,data=TRAIN_SAMPLE,method="over",na.action=na.pass,N=426,seed=6)$data
TRAIN_SAMPLE3<-ovun.sample(CONVERSION_OUTCOME~.,data=TRAIN_SAMPLE,method="under",na.action=na.pass,N=218,seed=6)$data
```