INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Predictive Response Mail Campaign

TIAGO ANDRÉ QUEIRÓS OLIVEIRA
Outubro de 2016

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Predictive Response Mail Campaign

Tiago André Queirós Oliveira

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas de Informação e Conhecimento

Orientador: Fátima Rodrigues

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, outubro de 2016

A todos os que me apoiaram neste projeto

Resumo

O marketing direto está a tornar-se cada vez mais um componente crucial para a estratégia de marketing das empresas e é um processo que inclui várias abordagens para apresentar produtos ou serviços a clientes selecionados. Uma base de dados fiável de clientes-alvo é crítica para o sucesso do marketing direto. O objetivo principal da modelação de respostas é identificar clientes com maior probabilidade de responder a um anúncio direto.

Existem dois desafios comuns ao lidar com dados de marketing: dados não balanceados, onde o número de clientes que não respondem é significativamente superior ao daqueles que respondem; e conjuntos de treino com elevada dimensão dado a enorme variedade de informações que são recolhidas normalmente.

Esta tese descreve todo o processo de desenvolvimento de um modelo de previsão de respostas ao mesmo tempo que apresenta e estuda diversas técnicas e metodologias ao longo dos vários passos, desde o balanceamento dos dados e seleção de variáveis até ao desenvolvimento e teste dos modelos. Adicionalmente, é proposta uma técnica de seleção de variáveis que consiste no agrupamento de várias *random forests* para obter resultados mais robustos. Os resultados mostram que a técnica de seleção de variáveis proposta, combinada com *random under-sampling* para o balanceamento dos dados, e a recente técnica *Extreme Gradient Boosting*, conhecida como XGBoost, têm a melhor performance.

Palavras-chave: Data Mining, Direct Marketing, Response Modelling, Feature Selection, Data Balancing, Classification

Abstract

Direct marketing is becoming a crucial part of companies advertising strategy and includes various approaches to presenting products or services to select customers. A reliable targeted customer database is critical to the success of direct marketing. The main objective of response modelling is to identify customers most likely to respond to a direct advertisement.

There are two challenges commonly faced when dealing with marketing data: imbalanced data where the number of non-responding customers is significantly larger than that of responding customers; and large training datasets with high dimensionality due to the significant variety of features that are usually collected.

This thesis describes the whole process of developing an efficient response prediction model while presenting and studying several different techniques and methods throughout the many steps, from data balancing and feature selection to model development and evaluation. Additionally, an ensemble feature selection technique that combines multiple random forests to yield a more robust result is proposed. The results show that the proposed feature selection method, combined with random under-sampling for class balancing, and the newer prediction technique Extreme Gradient Boosting, known as XGBoost, provide the best performance.

Keywords: Data Mining, Direct Marketing, Response Modelling, Feature Selection, Data Balancing, Classification

Agradecimentos

A realização desta dissertação só foi possível graças ao apoio e contribuição de um grupo de pessoas e instituições a quem gostaria de agradecer.

À minha orientadora, Fátima Rodrigues, por toda a ajuda, aconselhamento, disponibilidade e principalmente a transmissão de conhecimento não só durante este projeto, mas também ao longo do meu percurso neste ramo do mestrado.

Ao Instituto Superior de Engenharia do Porto (ISEP) que me acolheu nos últimos seis anos e foi a minha segunda casa durante este ciclo da minha vida, e pela qualidade não só do ensino prestado, mas também dos seus profissionais que me acompanharam ao longo do caminho.

E um agradecimento especial a todos à minha volta que suportaram a minha falta de sono e me deram apoio de forma incansável durante todo o tempo em que andei concentrado na escrita e desenvolvimento deste documento e do projeto em si, principalmente aos meus pais e à minha namorada.

Obrigado.

Table of Contents

Table of Figures

Table of Tables

Acronyms and Symbols

Acronym List

BI	Business Intelligence
RFM	Recency, Frequency, Monetary Value
GB	Gigabyte
RAM	Random Access Memory

1 Introduction

1.1 Context

Direct marketing is becoming a crucial part of companies advertising strategy and consists on sending offers or personalized campaigns directly (e.g. through mail) to select customers thus establishing a closer contact and waiting for a positive response out of them. To select the targets, it's important to understand not only what the customers are currently buying but also what they are interested in purchasing in the future, or the probability of positively responding to a direct offer or campaign. Companies must make sure they are focusing on customers who are likely to respond in order to increase their profits in a smarter way and not upsetting customers with constant mail advertising with offers they are not likely to take advantage of (Bose & Chen 2009).

As such, direct marketing has become target of multiple studies matching Business Intelligence (BI) techniques (such as machine learning or data mining which will be expanded further into this dissertation) with campaign/offer response prediction. Using algorithms and techniques such as linear regression, Bayesian or artificial neural networks, or decision trees it is possible to generate models able to accurately predict whether or not a customer is going to respond to the offer or how likely it is to get a positive response out of them (Loshin 2013; Chen et al. 2012).

On this thesis, the dataset to process is a publicly available one filled with real-world data provided by a large insurance company to study multiple approaches to direct marketing success using prediction models generated with different techniques. The goal is to capitalize on the growing interest of this area of study and test which techniques provide the best results for this type of large and "messy" datasets.

1.2 Problem

Springleaf is an American financial services company based on Indiana with over 95 years of existence, 8,000 employees and nearly 2,000 branches across 43 states, whose goal is "to

deliver the best customer experience and empower (the customer) to take control of (their) finances" by offering them personal and auto loans. They consider sending direct offers through mail a fundamental part of their marketing strategy as those provide great value to customers who might need them. Springleaf published a dataset filled with real data (anonymized for customer security purposes) on Kaggle and made it publicly available for competition purposes (Springleaf 2016).

Kaggle is a website founded in 2010 that hosts data mining competitions sponsored by companies all around the world, some of them with large sums of prize money on the line. Data scientists sign up on the website and have access to the competitions which mostly consist on analysing provided datasets (both real-life or dummy) and submit the best possible generated model to predict the goal attribute(s). By participating the users get more website reputation until they achieve the rank of Master, this rank allows them to participate in exclusive competitions that usually involve bigger companies and large amounts of sensible data, thus requiring some trust bond between the companies and respected data scientists (Kaggle 2016).

The problem faced by Springleaf and other companies that follow the same business strategies is predicting whether a client will respond to a direct mail campaign or not. Knowing this information could help companies better direct their marketing efforts in order to capture business opportunities with prospective clients and not losing customers that could receive too many campaigns they are not interested in.

1.3 Goal

The goal of this project is to respond to Springleaf's challenge on Kaggle and use the data to build a model able to accurately predict how likely it is for each customer to positively respond to a direct mail offer, by processing the dataset and employing feature-selection techniques to approach it. Due to the scientific nature of this project the problem won't be solved just by answering the challenge but also by studying and analysing the different available techniques (e.g. naïve Bayes, decision trees, neural networks, …) and existing studies and solutions to find out which is the most appropriate method to apply on such a large dataset.

1.4 Value Analysis

Since the dataset is filled with real-life data provided by Springleaf, that company would theoretically be the biggest benefactor with the final product (the most accurate generated prediction model), but considering this project is being developed independently with no interaction with them and given the fact that a model for one dataset can't be transversal to others, no direct customers exist for the model.

Still the data science community can benefit from the extensive research input on this dissertation as it will approach several methodologies and technologies that can contribute to further research and future model development.

1.5 Proposed Methodology

The clear domain of this project is Business Intelligence (BI), which can be defined as the resources (technologies, applications, and so on) used to analyse business data, or in this case market data, to help make smarter decisions (Chen et al. 2012). As defined by the Data Warehousing Institute, BI includes "data warehousing, business analytic tools, and content/knowledge management" (Loshin 2013).

As such there are several possible methodologies in this area that can be studied and applied to generate the final predictive model. Since there is no universally "best" learning method, there is the need to research, evaluate and test at least some of them to compare and decide which is the most appropriate for this type of dataset. In addition to the algorithm to use for prediction, and due to the large nature of the data in question, searching for feature selection methodologies as well as balancing methods is also needed. The process of selection was based on existing research and published papers and articles in this field, to help narrow down some of the best techniques and/or technologies, which were then tested on this specific data in order to present them and their respective results on the dissertation.

In the end the chosen methodologies to work with where using an ensemble of random forests and the Relief algorithm for feature selection, SMOTE and random under-sampling for class balancing and three different classification methods: two machine learning algorithms, random forest and neural network, and a probabilistic model, Naïve Bayes. However, during further research, a more recent but extremely effective machine learning method based on gradient boosting trees named XGBoost was found, which provided surprising results on this dataset and as such was added as a fourth alternative to this study.

The final product is composed by a script able to generate an accurate (or the most accurate possible) prediction model and the model itself, along with all the research present on this document.

1.6 Achieved Results

A total of sixteen classification models were tested, resultant of the different combinations between the prediction algorithms, data balancing methods and feature selection. In the end, the combination that proved the most effective in correctly predicting responding customers was the combination of an ensemble of random forests for feature selection, with random under-sampling for class balancing, and an algorithm based on gradient boosting trees, XGBoost, as the prediction model.

1.7 Thesis Outline

The presented thesis report is organized as follows: In this first chapter, the motivations for this work, its goals and main contributions were stated. Chapter 2 presents the concepts of Data Mining and Direct Marketing, followed by value analysis and state of art regarding existing technologies and methodologies related to the subject. Chapter 3 describes existing approaches for evaluating this type of work. Chapter 4 provides an insight to the solution design and the data that will be processed. Chapter 5 first describes the characteristics of the marketing data and the initial cleaning operations applied to it. It is followed by a description of filter selection methods and two wrapper selection methods, the ensemble feature selection algorithm proposed, based on random forest, and the relief algorithm. Next, to overcome the problem with imbalanced data, two balancing methods that use different sampling strategies are presented. At the end of the chapter, several prediction models are built. On chapter 6, the results achieved with the different combinations of feature selection, class balancing and prediction model are discussed. Finally, in chapter 7, the main conclusions of this work are presented, along with directions for future work.

2 Context and State of Art

2.1 Business Concepts

As previously mentioned the main area of this project is Business Intelligence, but the core of the problem is the connection between direct marketing and using data mining techniques to enhance said marketing. To better understand the concept behind these two topics some general research on their concepts is provided below.

2.1.1 Direct Marketing

Direct Marketing can be seen as a type of advertising which consists in companies communicating with customers in a more direct way, whether by websites, online ads or by mail. The opposing type of advertising is called mass marketing and uses mass media (TV, radio and newspaper ads) to send messages to every customer regardless of whether he is a good target for the product in question. This expansive broadcast of information has an extremely low response rate by the clients, typically averaging at less than 5%. (Ling & Li 1998) Instead of opting for the more visible but much more expensive and impersonal way, companies started adopting direct marketing back in the mid-1980s where it started to gain visibility due to the increase of market competitivity. It involves a study of customer's characteristics and needs to select targets, and because it imposes direct contact with the customers, it means the companies can personalize communications with different names and/or messages and keep their campaigns mostly invisible and applied only to prospect consumers. Over the years it has established as a rentable alternative to mass marketing and allows companies to increase their profit (either by not spending on unnecessary marketing to possible non-responding customers, or by investing only on customers who are likely to respond) throughout campaigns. One of the key features that is directly responsible for the continuous growth of direct marketing is its measurability (Ling & Li 1998; Roberts & Berger 1999).

2.1.2 Data Mining

According to Tan (2005) data mining is "the process of automatically discovering useful information on large data repositories". It's a Business Intelligence technology that uses mathematical algorithms, machine learning methods and data-driven models to try to reach possibly useful information that could otherwise be inaccessible and would remain hidden on a database (Moro & Laureano 2011; Sing'oei & Wang 2013). It's increasingly used in diverse areas from scientific discovery to surveillance but also commonly used for marketing purposes, e.g. find out what group of customers are interested in buying a specific product, with the goal of increasing profits, reduce expenses, or both (Moro & Laureano 2011; Suman et al. 2012).

Data mining can also be referred to as knowledge discovery in databases, also known as KDD, the process of "discovering useful knowledge from data" (Sing'oei & Wang 2013). Commercially, the term data mining is used to describe the whole process but academically and for many authors such as Fayyad, Tan, Sing'oei, Wang, Suman and others, data mining is a step of KDD itself, with Fayyad going a step further to divide the process and explain the steps involved: data selection, retrieving the data; data pre-processing, cleaning irrelevant data and correcting flaws on the dataset, while transforming it for mining; data mining, choosing the mining techniques/algorithms and applying them; interpretation/evaluation, visualization and analysis of the results. This process is depicted on Figure 1.

Figure 1 - Process of knowledge and discovery in databases (KDD) (Fayyad et al. 1996)

2.2 Existing Restrictions

In terms of restrictions the most obvious one at first was the dataset size. Exact dimensions and its characteristics will be explained with more detail on outcome 4 but it's almost 1GB of training data and 1GB of test data that was provided by Springleaf with thousands of examples and over 1900 variables to process, making this one heavy dataset to work with. Initially the only solution that seemed feasible was to work with subsets of data, which would make the task of exploring, pre-processing and correcting the initial datasets considerably harder. However, due to the vast amount of time it initially took to execute just a small amount of processing on the dataset,

there was the need to upgrade the data mining environment. A laptop with Intel Core i7 processor with 6GB RAM was used in the start, and while it passed reasonably well through the initial data reading and cleaning, when the model generation part started the machine took too long to perform any operation. The working station received an upgrade to 16GB RAM and this alone made the rest of the work much more bearable. It still took some time to complete some operations but it allowed much better handling around the data size restriction.

Another possible restriction isn't a restriction per se but the existence of a large amount of data mining tools, techniques and algorithms available for this type of classification task that need to be analysed, studied and possibly tested certainly increased the workload substantially. The solution around this was to search a considerable amount of articles/papers, analyse them during the state of art elaboration and choosing the ones that proved most effective in order to test them and register the results.

2.3 Value Analysis

Following the Analysis of Business Value module, some questions/statements about value analysis in general and this project itself were proposed, and they are as follows.

2.3.1 The need of a well-defined value proposition on a business

A value proposition is seen as an overview not only of the products the company will make available for their customers but also how can those products or services provide value to them. Businesses need a well-defined value proposition so that customers can easily understand why they should choose that business instead of its competitors or why should they pay them anything in the first place. It also allows businesses to better visualize their target market and work to advertise and appeal to that specific audience. A proper value proposition should clearly show what the product/service is, for who it is and why it is unique since different customers have different perceptions of value for the same product and a concise VP can help direct their perception. A well-defined value proposition can also help a business through crucial decisions that could be made during development, by providing the staff with a market focus they know what attributes to target in development and can decide on any given problem if one characteristic provides a valuable trade-off with another, e.g. to make a laptop more powerful it needs to be heavier, will the target costumers be willing to accept that increase or would they still prefer a lighter product even if that meant worst performance (Chesbrough 2002).

2.3.2 Value for possible customers that might use this product

The product to be developed in this project is going to be built exclusively for Springleaf, but since there is no direct connection between the developer and them, there is no real customer.

Still, in a hypothetical scenario, the only possible customers would be Springleaf employees that would use the model and get predictions about whether a customer would respond positively or negatively to a mail campaign.

The value this application can provide to those employees is giving them an easy way to know if a certain individual is going to respond positively to a mail campaign, and by knowing what customers are more likely to respond they can avoid sending the campaign mail to the ones that probably won't. This has several advantages as it allows the company to save money on advertising that would go to the wrong customers and redirect it to prospect new clients and maybe get some more value of the existing ones, making each subsequent campaign more rentable with enhanced profits and reduced costs. Another benefit is avoiding loss of costumers by constantly sending campaigns for the "wrong" customers, as studies show that the constant direct mail marketing that serves no purpose to a customer can and probably will be perceived as irritating and intrusive to the recipients (Morimoto & Chang 2006).

2.3.3 Possible negotiation scenarios

Since this product is being developed as if it was ordered by Springleaf, and because it uses their exclusive data to build the prediction model, the result is not transversal to other datasets, and so the only possible customer would be the company itself. The scenarios that can be encountered would be distributive (win-lose) or integrative (win-win) negotiations.

On a distributive scenario both parties need to concede on some of their issues in favour of the opposing party. This scenario is appropriate when the product is limited, i.e. if there's something that needs to be divided among them and each time one party "wins" some element of the product, that element is "lost" by the opposing party. On the other hand, the integrative scenario provides a situation beneficial to both parties involved, as they can concede on smaller issues to achieve mutual better scenarios. This works when there is an interest on establishing a relationship around a "unlimited" product (such as this one), where it is possible to achieve something greater together then either party could reach on their own (Stöckli & Tanner 2014).

Since the product is not "limited" and both the company and the developer have interest on keeping the relationship going forward, as they need the model to be constantly updated with the current data from each campaign and the developer needs the fixed income that comes with it, an integrative negotiation would be the way to go, a situation where both could win. By asking for a small fee every month instead of a huge one-time payment it would be more profitable for both sides since the developer had a guaranteed revenue every month and the company wouldn't have to make a big investment at once and could keep getting their product up-to-date.

2.3.4 Business Model Canvas

Osterwalder and Pigneur (2010) define a business model canvas as "a shared language for describing, visualizing, assessing, and changing business models" and it is composed by nine blocks:

- Customer Segments: who are our potential customers?
- Value Propositions: how are we helping the customers?
- Channels: how do we deliver the product to customers?
- Customer Relationships: how do we establish and maintain relationships with the customers?
- Revenue Streams: where do we get revenue from?
- Key Resources: what assets do we require to accomplish our goals?
- Key Activities: what do we need to do to accomplish our goals?
- Key Partnerships: what partnerships do we need or are interested in?
- Cost Structure: what will we need to spend our funds on?

The business model canvas for the proposed project is presented on Figure 2 using a template from Strategyzer, a website founded by the creators of the business model canvas (Osterwalder & Smith 2016).

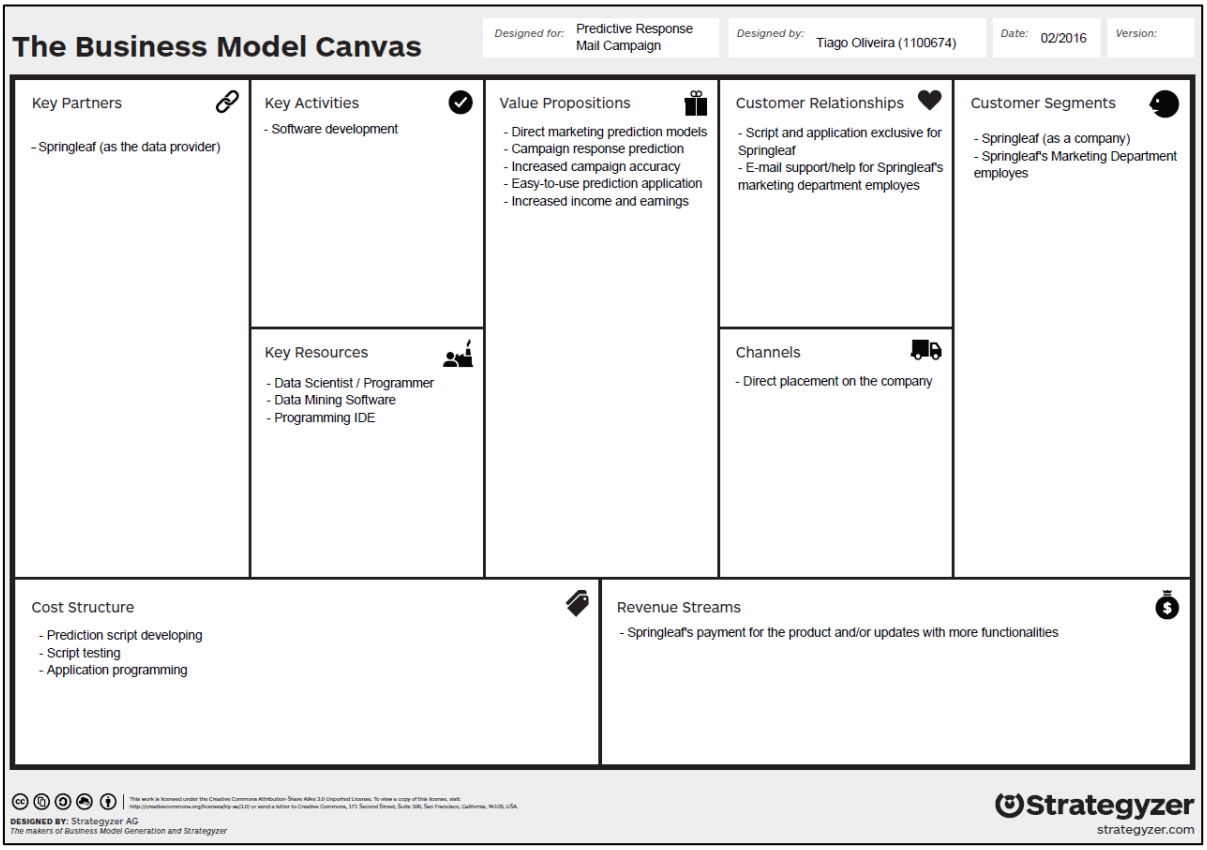

Figure 2 - Business Model Canvas for this product

2.4 State of Art

In order to grasp on what has already been studied on the field of data mining applied to direct marketing, a research was conducted for papers and articles that not only studied this field but that included experiments and trials on public or private datasets employing multiple data mining techniques and algorithms. To ease the research, reading and comparing process, a table was used to write and present the data in each of the articles/papers. The research is presented on Table 1 which contains the respective reference for each article, how many datasets and their sizes (when that information is available) were tested, the amount and type of goal attribute(s), what techniques and algorithms were applied, what evaluation processes were used and the results of each experiment with a comparison table.

<table>
<tr><th>Reference</th><th>Datasets/Goals</th><th>Techniques/Evaluation</th></tr>
<tr><td>“Improved response modelling based on clustering, under-sampling, and ensemble”
(Kang et al. 2012)</td><td>2 public datasets:
▪ 5 822 x 85 (CoIL 2000[1])
▪ 101 532 x 15 (DMEF4[2])

Goal attributes:
▪ 1 class (binary)</td><td>▪ Logistic Regression (LR)
▪ Multi-Layer Perceptron Neural Network (MLP)
▪ K-Nearest Neighbour (KNN)
▪ Support Vector Machine (SVM)

Evaluation:
▪ Accuracy (Confusion Matrix) (ACC)
▪ Balanced Correction Rate (BCR)

<table>
<tr><th></th><th></th><th>LR</th><th>MLP</th><th>K-NN</th><th>SVM</th></tr>
<tr><td rowspan="2">CoIL2000</td><td>ACC</td><td>0.699±0.14</td><td>0.673±2.29</td><td>0.717±0.21</td><td>0.706±0.24</td></tr>
<tr><td>BCR</td><td>0.673±0.21</td><td>0.662±0.78</td><td>0.681±0.29</td><td>0.671±0.21</td></tr>
<tr><td rowspan="2">DMEF4</td><td>ACC</td><td>0.824±0.22</td><td>0.861±0.09</td><td>0.845±0.04</td><td>0.815±0.25</td></tr>
<tr><td>BCR</td><td>0.776±0.15</td><td>0.815±0.76</td><td>0.837±0.20</td><td>0.781±0.10</td></tr>
</table>

Note: Data reconstructed using a new balancing method based on clustering, under-sampling and ensemble (CUE).</td></tr>
</table>

[1] http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000

[2] http://www.marketingedge.org/marketing-programs/data-set-library (former DMEF)

<table>
<tr><th>Reference</th><th>Datasets/Goals</th><th>Techniques/Evaluation</th></tr>
<tr>
<td>“Bayesian neural network learning for repeat purchase modelling in direct marketing”
(Baesens et al. 2002)</td>
<td>2 datasets:
▪ 100 000 x 12 (RFM[3] only)
▪ 100 000 x 22 (w/ non-RFM)

Goal attribute:
▪ 1 class (binary)</td>
<td>▪ Logistic Regression (LR)
▪ Bayesian Neural Networks (w/ ARD[4]) (BNN-ARD)

Evaluation:
▪ Percentage Correctly Classified (PCC)
▪ Area Under ROC[5] Curve (AUROC)

<table>
<tr><th colspan="2"></th><th>LR</th><th>BNN-ARD</th></tr>
<tr><td rowspan="2">RFM only</td><td>PCC</td><td>70.3±0.1</td><td>71.4±0.2</td></tr>
<tr><td>AUROC</td><td>77.5±0.1</td><td>78.7±0.2</td></tr>
<tr><td rowspan="2">non-RFM</td><td>PCC</td><td>71.4±0.2</td><td>72.5±0.3</td></tr>
<tr><td>AUROC</td><td>78.7±0.2</td><td>80.0±0.3</td></tr>
</table>
</td>
</tr>
</table>

[3] Recency, Frequency, Monetary Value
[4] Automatic Relevance Determination
[5] Receiver Operating Characteristic

<table>
<tr><th>Reference</th><th>Datasets/Goals</th><th>Techniques/Evaluation</th></tr>
<tr><td>“Machine Learning for Direct Marketing Response Models: Bayesian Networks with Evolutionary Programming” (Cui et al. 2006)</td><td>1 dataset:
▪ 106 284 x 361

Goal attribute:
▪ 1 class (probability)</td><td>▪ Bayesian Neural Networks (BNN)
▪ Artificial Neural Networks (ANN)
o w/ Bayesian learning and MCMC[6]
▪ Classification and Regression Tree (CART)
▪ Latent Class Regression (LCR)

Evaluation:
▪ 10-fold Cross-Validation Lift (Cumulative Lift)

<table><tr><th></th><th>BNN</th><th>ANN</th><th>CART</th><th>LCR</th></tr><tr><td>Cumulative Lift</td><td>408.0</td><td>396.5</td><td>365.7</td><td>401.1</td></tr></table></td></tr>
<tr><td>“A prediction model for the purchase probability of anonymous customers to support real time web marketing: A case study” (Suh et al. 2004)</td><td>1 dataset:
▪ 5313 x 21

Goal attribute:
▪ 1 class (binary)</td><td>▪ Decision Trees (DT)
▪ Neural Networks (NN)
▪ Logistic Regression (LR)

Evaluation:
▪ Accuracy (ACC)

<table><tr><th></th><th>DT</th><th>NN</th><th>LR</th></tr><tr><td>ACC</td><td>0.895</td><td>0.886</td><td>0.892</td></tr></table></td></tr>
</table>

[6] Markov Chain Monte Carlo

<table>
<tr><th>Reference</th><th>Datasets/Goals</th><th>Techniques/Evaluation</th></tr>
<tr><td>“Improving direct mail targeting through customer response modelling” (Coussement et al. 2015)</td><td>4 datasets:
99 200 x 11 (DS1)
96 551 x 142 (DS2)
106 284 x 250 (DS3)
101 532 x 87 (DS4)
Goal attribute:
1 class (binary)</td><td>Logistic Regression (LR)
Neural Networks (NN)
Naïve Bayes (NB)
Decision Trees
Chi-Square Automatic Interaction Detector (CHAID)
CART
C4.5
K-Near Neighbours (w/ K=10 and K=100) (KNN10 / KNN100)
Evaluation:
10-fold Cross-Validation AUROC
<table>
<tr><th>(AUROC)</th><th>LR</th><th>NN</th><th>NB</th><th>CHAID</th><th>CART</th><th>C4.5</th><th>KNN10</th><th>KNN100</th></tr>
<tr><td>DS1</td><td>0.66</td><td>0.68</td><td>0.62</td><td>0.68</td><td>0.67</td><td>0.67</td><td>0.63</td><td>0.67</td></tr>
<tr><td>DS2</td><td>0.64</td><td>0.60</td><td>0.58</td><td>0.64</td><td>0.64</td><td>0.54</td><td>0.52</td><td>0.57</td></tr>
<tr><td>DS3</td><td>0.82</td><td>0.82</td><td>0.73</td><td>0.86</td><td>0.84</td><td>0.82</td><td>0.68</td><td>0.75</td></tr>
<tr><td>DS4</td><td>0.79</td><td>0.82</td><td>0.67</td><td>0.82</td><td>0.80</td><td>0.78</td><td>0.70</td><td>0.76</td></tr>
</table>
</td></tr>
<tr><td>“A Hybrid Framework using RBF and SVM for Direct Marketing” (Govidarajan 2013)</td><td>1 dataset:
435 x 16
Goal attribute:
1 class (binary)</td><td>Radial Basis Function (RBF)
Support Vector Machines (SVM)
Radial Basis Function SVM (RBF-SVM)
Evaluation:
Accuracy (ACC)
<table>
<tr><th></th><th>RBF</th><th>SVM</th><th>RBF-SVM</th></tr>
<tr><td>ACC</td><td>94.48%</td><td>96.09%</td><td>99.31%</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Reference</th><th>Datasets/Goals</th><th>Techniques/Evaluation</th></tr>
<tr><td>“Semi-Supervised Response Modelling”
(Lee et al. 2010)</td><td>2 public datasets:
5 822 x 85 (CoIL 2000)
101 532 x 91 (DMEF4)
Goal attribute:
1 class (probability)</td><td>Logistic Regression (LR)
Support Vector Machines:
Linear SVM (LSVM)
Radial Basis Function SVM (RBFSVM)
Transductive SVM (TSVM)
Evaluation:
5-fold Cross-Validation AUROC
<table><tr><th>(AUROC)</th><th>LR</th><th>LSVM</th><th>RBFSVM</th><th>TSVM</th></tr><tr><td>CoIL2000</td><td>0.71</td><td>0.70</td><td>0.70</td><td>0.72</td></tr><tr><td>DMEF4</td><td>0.83</td><td>0.82</td><td>0.84</td><td>0.83</td></tr></table></td></tr>
<tr><td>“Using Data Mining for Bank Direct Marketing: An application of the CRISP-DM methodology”
(Moro & Laureano 2011)</td><td>1 dataset:
45 211 x 29
Goal attribute:
1 class (binary)</td><td>Naïve Bayes (NB)
Decision Trees (DT)
Support Vector Machines (SVM)
Evaluation:
AUROC
Area Under LIFT Curve (AULIFT)
<table><tr><th></th><th>NB</th><th>DT</th><th>SVM</th></tr><tr><td>AUROC</td><td>0.870</td><td>0.868</td><td>0.938</td></tr><tr><td>AULIFT</td><td>0.827</td><td>0.790</td><td>0.887</td></tr></table></td></tr>
</table>

<table>
<tr><th>Reference</th><th>Datasets/Goals</th><th>Techniques/Evaluation</th></tr>
<tr><td>“Direct marketing decision support through predictive customer response modelling”
(Olson & Chae 2012)</td><td>2 datasets:
101 532 x 3
1 009 009 x 3
Goal attribute:
1 class (binary)</td><td>RFM-based Models (RFM)
Logistic Regression (LR)
Decision Trees (DT)
Neural Networks (NN)
Evaluation:
Accuracy (ACC)
<table><tr><th>(ACC)</th><th>RFM</th><th>LR</th><th>DT</th><th>NN</th></tr><tr><td>DS1</td><td>0.907</td><td>0.907</td><td>0.984</td><td>0.911</td></tr><tr><td>DS2</td><td>0.6625</td><td>0.9385</td><td>0.9386</td><td>0.9386</td></tr></table></td></tr>
<tr><td>“Personalized Email Marketing with a Genetic Programming Circuit Model”
(Kwon & Moon 2001)</td><td>2 datasets:
86 classes each
Goal attribute:
1 class (binary)</td><td>Circuit Genetic Programming (CGP)
Collaborative Filtering (CF)
Artificial Neural Networks (ANN)
Evaluation:
Campaign Response Percentage
(average of both datasets)
<table><tr><th></th><th>CGP</th><th>CF</th><th>ANN</th></tr><tr><td>Response</td><td>4.78%</td><td>4.00%</td><td>4.44%</td></tr></table></td></tr>
</table>

<table>
<tr><th>Reference</th><th>Datasets/Goals</th><th>Techniques/Evaluation</th></tr>
<tr>
<td>“Customer-adapted coupon targeting using feature selection”
(Buckinx et al. 2004)</td>
<td>1 dataset:
▪ 3 500 x 98

Goal attribute:
▪ 1 class (binary)</td>
<td>▪ Decision Trees
o C4.5

Feature Selection:
• Relief-F (RelF)

Evaluation:
▪ Accuracy (ACC)

<table>
<tr><th></th><th>C4.5</th></tr>
<tr><td>ACC w/ RelF</td><td>62.92%</td></tr>
<tr><td>ACC wo/ RelF</td><td>60.89%</td></tr>
</table>
</td>
</tr>
<tr>
<td>“Data Mining Framework for Direct Marketing: A Case Study of Bank Marketing”
(Sing’oei & Wang 2013)</td>
<td>1 dataset:
▪ 45 212 x 17

Goal attribute:
▪ 1 class (binary)</td>
<td>▪ Decision Trees
o C5.0

Evaluation:
▪ 10-fold cross validation LIFT

<table>
<tr><th></th><th>C5.0</th></tr>
<tr><td>LIFT</td><td>0.8</td></tr>
</table>
</td>
</tr>
</table>

Table 1 - State of Art (existing solutions/approaches)

2.5 Relevant Existing Technology

After analysing the previous articles and according to the results presented on Table 1 it's possible to identify some techniques that can produce the accurate models for marketing prediction. For the sake of research instead of just picking one technique, and since there can't be an absolute best for all datasets in existence, three were chosen while trying to represent a wide range of approaches: two machine learning algorithms (Random Forest and Neural Network) and a probabilistic model (Naïve Bayes).

As a result of further literature research later on the project another technique came to light, Gradient Boosted Trees (GBT), and a system called XGBoost that employs this technique to create prediction models. GBTs function is to try to improve the performance of a model by creating an ensemble of weaker models, combining them for prediction (Elith et al. 2008). XGBoost on the other hand is a tool that promises to "achieve state-of-the-art results on many machine learning challenges" with its algorithm of tree boosting and the promise of using less resources than other existing systems (Chen & Guestrin 2016). Due to the positive turnout of the research and the proven results (Babajide Mustapha & Saeed 2016; Jain et al. 2015), a decision was made to add a fourth approach to this dissertation and join XGBoost to the other three previously mentioned.

Some general research on the concepts of each technology/algorithm is provided in this section.

2.5.1 Random Forest

Decision trees are a popular classification technique and one of the most accessible in the industry, used in multiple areas inside marketing such as customer segmentation, sales forecasting or predicting survey responses (Coussement et al. 2015). The tree is composed by nodes, branches and leaves where each node represents a certain test on an attribute, each branch represents a possible test result, and each leaf represents a classification. It's usually constructed from training data so that test data can then be tested from the root throughout the tree until it reaches a leaf and is classified (Sing'oei & Wang 2013).

The type of tree changes according to the type of data to predict: classification trees, in case the outcome to be predicted is a class (which is the case for this project, where the objective is to predict if a customer responds or not), and regression trees, where the outcome is a number (e.g. the probability of a customer responding to the campaign). There are various decision tree algorithms that can be implemented (such as CART, C4.5 and CHAID) that provide different results for the same problem since they differ on the splitting criteria for cutting the tree, whether they work with regression or classification trees, if they are capable of handling incomplete data and if they are capable of eliminating or reducing over-fitting (Tan et al. 2005).

Figure 4 is a representation of data being classified according to a decision tree. The root node in this case is the variable “Body Temperature”, which can lead directly to a leaf (“Non-mammals”) or another test “Gives Birth”, which finally classifies the data (Tan et al. 2005).

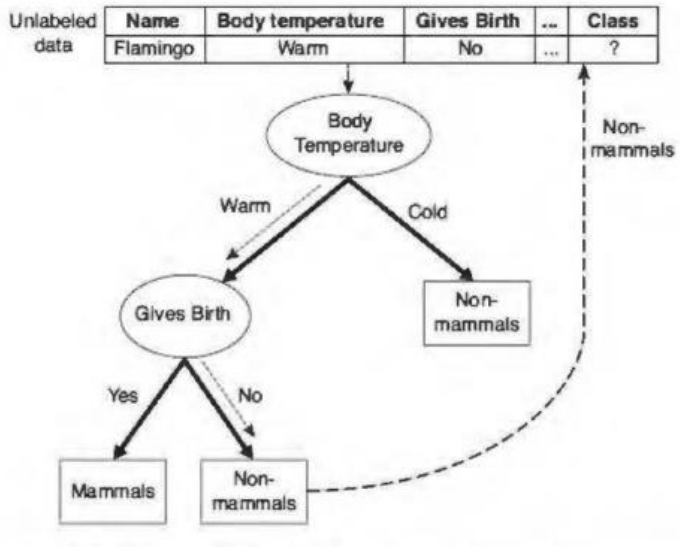

Figure 3 - Example of data being classified with a binary class (Tan et al. 2005)

For this thesis, the selected method was random forest (Breiman 2001), which is an ensemble method, i.e. a method that uses multiple trees in order to improve the accuracy of the prediction. The method then outputs the result that appears more often throughout all the generated trees. Another great advantage of this method is that it can be used to study variable importance, which in this case where the dataset is so extensive and it is certain to need some filtering, is a feature of great importance. Decision trees have a significant problem with overfitting, something that occurs when the algorithm tries to reduce the training set error but increases the test set error, and the use of the random forest algorithm considerably diminishes that problem (Kuhn & Johnson 2013).

2.5.2 Neural Networks

Neural networks are one of the classic data mining tools, commonly found throughout multiple data mining products, with proven efficiency in several case studies (as seen on Table 1), and can be described as a “processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use” (Olson & Chae 2012; Coussement et al. 2015). They produce high performance and mimic both the structure and functioning of the brain by simulating its neurons and connections. A neural network consists on at least three layers, the first one is the input layer, corresponding to the independent variables, one node per variable. The last layer is the output layer, corresponding to the dependent variable, the classifications being predicted, one node for each possible category. In between one can have as many “hidden” layers as needed, although literature shows that one is complex enough for most problems. Every input neuron is connected to the

hidden neurons and every hidden neuron is connected to the output neurons (Coussement et al. 2015; Olson & Chae 2012; Heilman et al. 2003). Figure 3 contains a depiction of a neural network with one hidden layer.

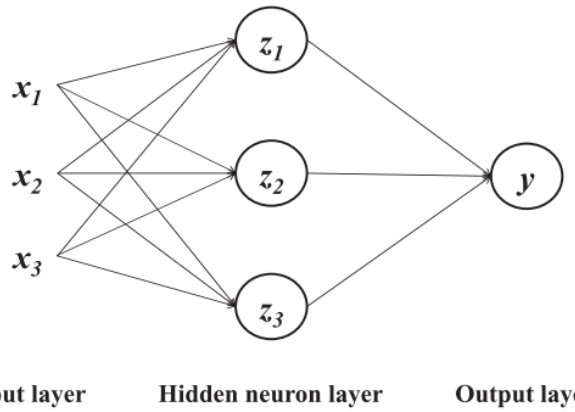

Figure 4 - Neural Network with one hidden neuron layer (Coussement et al. 2015)

By applying Bayes conditional probability theorem to train a neural network one can assign a conditional probability to each of the nodes in the network and by taking into account the connections between them, make predictions about each of the outcomes on the output layer and how likely they are to be selected (Baesens et al. 2002; National 2005; Cui et al. 2006). Those types of networks are known as Bayesian Neural Networks and the theorem is explained on the next section.

2.5.3 Naïve Bayes

Naïve Bayes is an extremely simple classifier based on Bayes theorem (represented on **(1)**) to calculate probability of a certain class while assuming that all features are independent and not correlated with each other in any way.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (\mathbf{1})$$

In which A and B are events, P(A) and P(B) are the respective probabilities of those events happening without any knowledge of each other, P(B|A) is the probability of B happening while knowing for sure that A happened. The equation gives us the probability of A happening knowing that B happened, or in this case, the likelihood of being classified as a specific class knowing that it has a certain value for B. The process starts by constructing a frequency table for each attribute and the target feature and use the Bayesian equation to calculate the probability for each class. The algorithm predicts the class to be the one with the highest probability in the end (Koch 1990).

It's a classifier that has been studied for many years now and although the assumption of variable independence (to which it attributes the "naïve" on its name) is generally poor, the classifier is surprisingly able to compete with more sophisticated approaches (Langley et al.

1992; Rish 2001) and with proven efficiency in many practical applications (Management et al. 2000).

2.5.4 XGBoost

XGBoost stands for "Extreme Gradient Boosting" and is a tool that uses boosted trees to solve classification problems while promising "state-of-art results" and much better performance while tackling machine learning challenges (Chen & Guestrin 2016). It's an open source technology based on a CART[7] tree ensemble model (a combination of multiple CART trees, shown of Figure 5) and has been appearing more and more throughout various data mining competitions and challenges, with almost every solution to appear at the top spots makes use of this machine learning method (Chen & He 2015). Chen and Guestrin (2016), the developers of this method attribute its success to both its scalability and speed, running "more than ten times faster" than other existing methods on a single machine, and this is due to various technical and algorithmic optimizations.

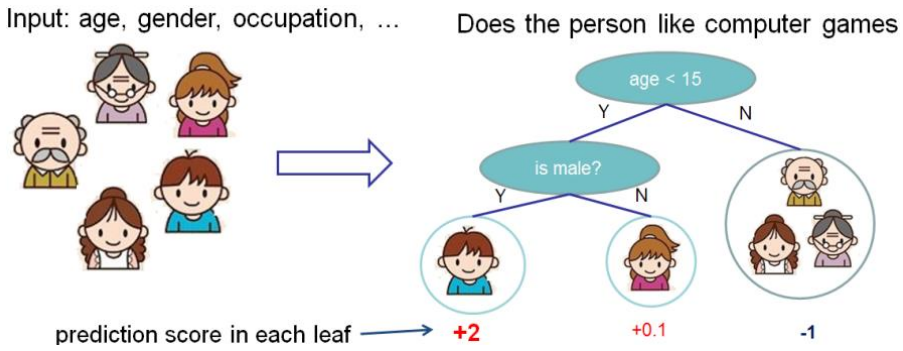

Figure 5 - Example of a CART tree, the basis of XGBoost (Chen & Guestrin 2016)

The results provided by this tool on this dataset were extremely surprising and were proven very effective on increasing the accuracy of predictions. Although it was discovered later on the development process for this project and despite being a relatively recent method, the amount of documentation and the parameter tuning it enables, makes it possible to considerably improve the final scores. On this specific case, instead of just determining for each customer whether he will respond positively or not to the marketing campaign, XGBoost presents the likelihood of a positive response per customer in percentage.

2.5.5 Support Vector Machines

Support vector machine (SVM) is a more recent classification technique that has become popular due to its efficiency and performance on practical applications such as text classification or pattern recognition (Govidarajan 2013). They aim to minimize training set error and can be applied to problems containing binary target variables. It works by linearly separating all the

[7] Classification and Regression Tree

examples of a training set where each one belongs to one the two classes. The SVM then searches for the optimal solution, the one that separates both groups by the largest margin (Lee et al. 2010; Steinwart & Christmann 2008).

The points on the boundaries of that margin are called the support vectors while the middle of the margin is the ideal hyperplane to separate both classes. An example of the hyperplane separation with support vector machines is presented on figure 5. It's important to notice that points placed opposite of what they should don't have as much weight as they normally would on the separation process (Williams 2008).

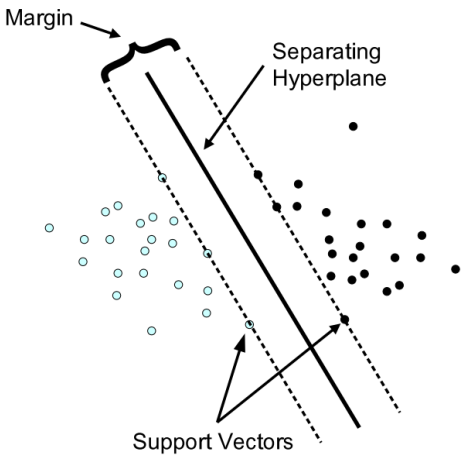

Figure 6 - Classification using a support vector machine (Steinwart & Christmann 2008)

Although SVMs appeared to be very successful among the papers studied during the state of art, they were ultimately not considered viable for this particular project due to their incompatible nature with unbalanced datasets and computational weight due to their quadratic optimization algorithm (Lee et al. 2010; Williams 2008). The dataset approached on this thesis is extremely unbalanced and the amount of data would make an SVM model impossible to run on the working environment.

3 Evaluate Existing Solutions

Due to the nature of the project being BI, data mining and machine learning it is necessary to use specific evaluation techniques for model accuracy comparison instead of the usual time to execute or memory usage. On outcome 2 several evaluation processes were identified throughout the state of art study, and some of them were selected for use on this project.

3.1 Evaluation Measures Identified

The key objective of a learning algorithm is to build models with good generalization capability, i.e. models that accurately predict the class labels of previously unknown records. In order to make a fair evaluation, once a model has been constructed, it must be applied to a test set to predict the class labels of previously unseen records. It is useful to use a test set, because such a measure provides an unbiased estimation of its generalization error. The accuracy or error rate computed from the test set can also be used to compare the relative performance of different classifiers on the same domain. Throughout this chapter, the selected processes to be used on this project for evaluating the performance of the classifiers are presented.

3.1.1 Confusion Matrix (Accuracy, Sensitivity and Specificity)

A confusion matrix is a record of how many examples were correctly or incorrectly predicted by a classifier model. In this specific context of direct marketing it would represent how many people who were predicted to respond to the campaign actually did or did not, and vice-versa, as depicted in table 2 (Tan et al. 2005).

		Predicted	
		Responders	**Non-responders**
Actual	**Responders**	True respondents (TP)	False non-respondents (FN)
	Non-responders	False respondents (FP)	True non-respondents (TN)

Table 2 - Example of a direct marketing confusion table (Kang et al. 2012)

In this particular context, the TP and TN represent customers that would respond to the campaign and were predicted as such, and customers who wouldn't respond to the campaign and were correctly predicted as non-responders, respectively. The FP represents customers predicted as responders but who would not respond to the campaign, and FN represents customers predicted as non-responders but who would actually respond. It's important to note that the costs associated with FP, which is the cost of mailing the campaign offer, are much lower than those associated with FN, which represent business opportunities that are lost because of a wrong prediction, which makes FN the most critical value on the confusion matrix for this particular project on a marketing perspective.

A confusion matrix allows one to summarize the data to easily compare classification models with some performance metrics that can easily be obtained from the table, e.g. the accuracy of the model **(2)** (Tan et al. 2005).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (\mathbf{2})$$

Other measures such as precision and recall can also be obtained by mixing confusion matrix values.

Although accuracy is the typical solution for prediction models' evaluation, this method is not the most effective on imbalanced data and/or when the cost of errors are very different for each class (Provost et al. 1998). For a two class prediction model, when one class is interpreted as the event of interest, as is the case in study, the statistics sensitivity and specificity are more relevant (Kuhn & Johnson 2013). Sensitivity **(3)** is the rate of correctly predicted samples with the event of interest for all the samples having the event, often considered the true positive rate (TPR). Opposite to that is specificity **(4)**, the rate of correctly predicted non-event samples for all the samples without the event. The false positive rate (FPR) is defined as 1-specificity.

$$Sensitivity = \frac{TP}{TP + FN} \quad (\mathbf{3})$$

$$Specificity = \frac{TN}{TN + FP} \quad (\mathbf{4})$$

For the project discussed on this thesis, it is desirable to have a model with high sensitivity given the fact that the event of interest are the responders, which is the minority class, and not the non-responders. Usually there is a trade-off between these two measures, where increasing the sensitivity of a model is likely to cause a decrease on the specificity, but these trade-offs

can be easily dealt with when the classes to predict have different costs and as such increasing the error rate on the less important class can be worth it if it means a significant increase on the more important one (Kuhn & Johnson 2013). This trade-off can be evaluated with the technique described next, which can also be used for model evaluation.

3.1.2 Area Under Receiver Operating Characteristic (ROC) Curve

A ROC curve is a standard technique for summarizing a classifier performance over a range of trade-offs and similar to accuracy it can also be obtained by taking results from the confusion matrix since it works with sensitivity and specificity previously mentioned. It consists in a graphic depicting the true positive rate (TPR) on the y-axis versus the false positive rate (FPR) on the x-axis, and the area under the curve (AUC) itself is an accepted performance metric for ROC curves (Bradley 1997). A perfect model would have 100% sensitivity and specificity, and the area under it would be 1, while a completely useless prediction model would produce a ROC curve that follows the diagonal of the graphic and would have an AUC of 0.5. Comparing models using this method is intuitive as the superimposition of multiple curves on the same graphic provides an easy to view result. During comparisons, the lines of two models can cross, and that means that none of them is the absolute best for that specific case, but instead one of them is better on a portion of the data, and the other one on another portion (the best is the one represented by the above line). One advantage of using this method for model evaluations is that because it is based on sensitivity and specificity, the curve is insensitive to imbalanced data (Kuhn & Johnson 2013), and thus a perfect method for the project on this thesis.

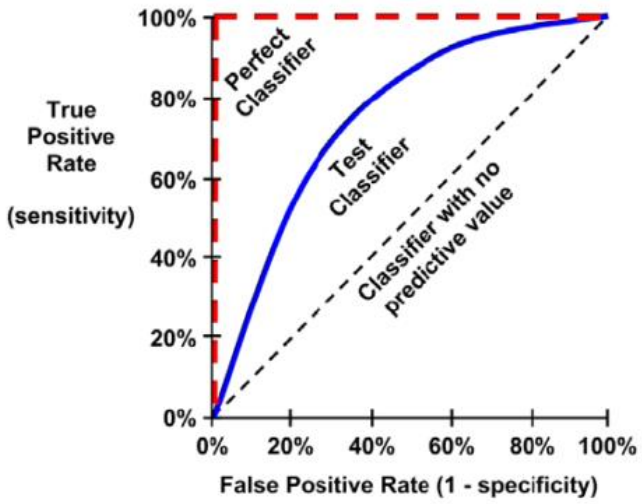

Figure 7 - Example of a ROC Curve and it's behaviour (Rodrigues 2015)

3.1.3 K-Fold Cross Validation

Cross-validation is an approach where each example is used the same number of times for training purposes and only once for testing. With the k-fold the data is partitioned into k equal-sized parts with one of them being the test while the others are for training and they are tested one by one consecutively against the partition selected as test. In the end, another partition is selected as being the test and the process repeats until every partition has been the test partition. Finally, the total error is the sum of the errors of all the runs divided by k (Tan et al. 2005). Figure 7 depicts an example of 5-fold cross validation, but usually the value of k varies between 5 or 10 as there is no formal rule for this selection and these values have been proven competent on the clear majority of cases. Note that the higher the k, the more computational power is required to perform the process (Kuhn & Johnson 2013; Rodrigues 2015).

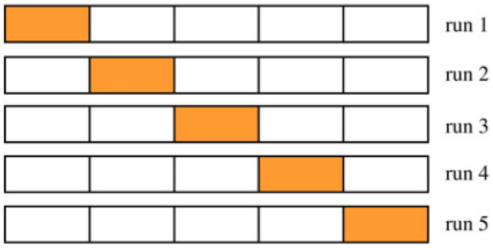

Figure 8- Example of 5-fold cross validation (Lee et al. 2010)

4 Solution Design

4.1 Conceptual Solution Design

The solution is composed by a number of R scripts (explained further) that follow the approach set by Tan (2005), depicted in figure 9, and Khun (2013) to solve the proposed classification problem. It started by analysing both datasets (training and test) provided by Springleaf, followed by a cleaning process which involved removing duplicate features, invalid data, empty fields and irrelevant variables that could harm the prediction. To get the most accurate models possible and due to this dataset's high dimensionality, it is important to select the best subset of features, which is why the next step was feature selection, using simple filters such as checking the interquartile range (IQR) and the Relief algorithm (Kira & Rendell 1992; Kononenko 1994), and a wrapper filter in the form of an ensemble feature selection composed by several processes within. The following step is balancing the data and developing the models. For data balancing the chosen methods were SMOTE (Chawla et al. 2002) and random under-sampling, followed by the development of the models described on chapter 2 (Random Forest, Neural Network, Naïve Bayes and XGBoost), generates a total of 8 different models, two models with different methods of data balancing for each modelling technique. Finally, the generated prediction models are evaluated against the test data using the methods specified on chapter 3 and proceed to compare them and evaluate what is the best solution for this specific problem.

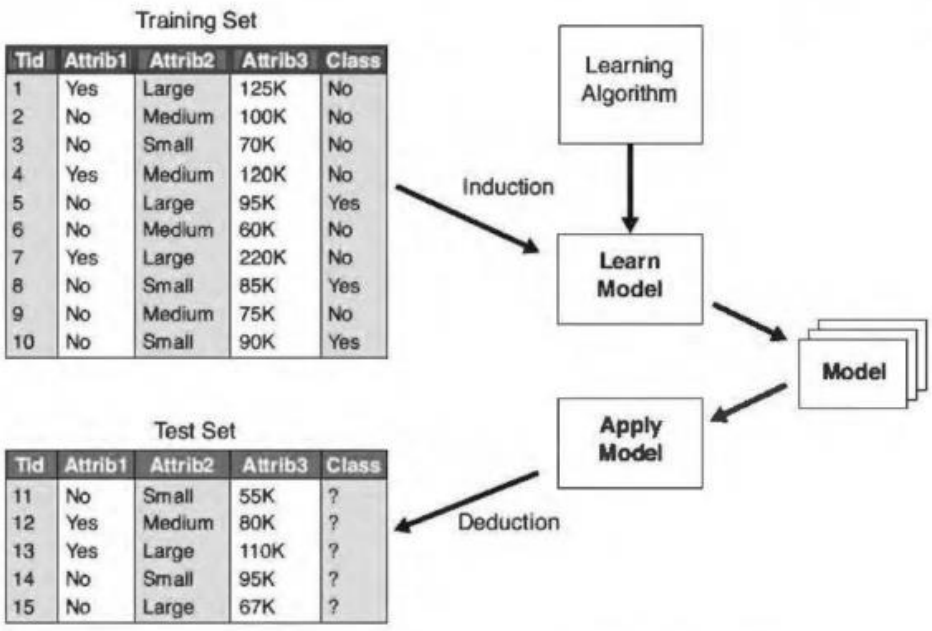

Figure 9 - General approach to solving a classification problem according to Tan (2005)

4.2 Provided Data

All the data provided for the project were 2 high dimensional and highly unbalanced datasets, a training one and a testing one, containing anonymized Springleaf customer information where each row corresponds to one customer and each column to one specific (and unknown) information. The datasets are almost 1GB each and have 1934 features and more than 100,000 rows each, with both numerical and categorical values. The main problem is that, in order to protect privacy, all the features have been anonymized and their values and types are provided "as-is", because as Springleaf wrote "handling a huge number of messy features is part of the challenge" (Kaggle 2016; Springleaf 2016).

The following code snippet depicts a sample of an execution of the command "head" on R to the original training set to give an idea of how the features are anonymized with their names being "VAR_XXXX" each with a corresponding number and no apparent meaning for most of them.

```
> head(train)
  ID VAR_0001 VAR_0002 VAR_0003 VAR_0004 VAR_0005 VAR_0006 VAR_0007
  2        H       224         0     4300        C        0        0
  4        H         7        53     4448        B        1        0
  5        H       116         3     3464        C        0        0
  7        H       240       300     3200        C        0        0
  8        R        72       261     2000        N        0        0
```

Code 1 – Command *head*() executed on the original training set

5 Building the Solution

Response modelling is a complex process consisting in several steps such as data collection, data cleaning, feature selection, class balancing, classification and evaluation. In order to reach the results of this work, all these steps had to be performed, and the process will be described throughout this chapter and the next.

5.1 Data Collection and Description

The data collection process, as previously described, consisted on downloading the data from Springleaf's challenge on Kaggle. Both the training set and the test set were 1GB in size, with thousands of entries and almost 2000 features to describe each one. The values were provided "as-is" and the feature names anonymized to protect privacy, which left no clue about what each number or text meant when analysing the data.

The software chosen to work on the data was RStudio, an open-source IDE designed to "empower users to be productive with R" and simplify working with that statistical programming language (RStudio 2016). The choice of this specific programming language and IDE was mainly due to familiarity with it, the ease of access to resources throughout the web and the fact that there are hundreds of packages with valuable content in the form of data mining techniques, algorithms and operations that can enhance this work.

The first step is to get an initial look at the data and start to analyse it for future processing. The problem with the dataset size was obvious from the start and it took some extra computer power only to read and do some basic processing on the dataset, but that was already expected given what was known about the data. But another problem surfaced, related to data imbalance. The dataset contains 145,231 registered clients, described by a total of 1934 features, 1880 of which are numerical while the other 54 are categorical (text or dates), and of all those only

23.3% responded to the campaign. The high variety of features and clients and the goal attribute so imbalanced, most modelling techniques will struggle to get any meaningful result.

5.2 Data Cleaning

To solve the first problem, data cleaning and filtering was necessary in order to remove some irrelevant features. Features that have almost all distinct values (such as ID features) provide no useful information to the prediction, so all features where there were more than 80% of distinct values were cut. On the same note, all features in which there were no distinct values (constant features) are useless and were therefore discarded as well.

The next step was taking care of missing values. The data had a lot of incorrectly filled information and some corrections needed to be made, such as replacing all "-1", "[]", "", and other invalid info for "NA" to help identify the missing values. At the end the amount of "NA" was a lot, and as such the next step was to remove every feature in which over 50% of it were missing information. Following that, all repeated variables on the dataset were removed, and this whole process caused the number of features to drop to 1698, 1676 numeric and 22 character attributes.

The number of features was still too high and in order to create a viable prediction model it is important to select the best subset of features, the most relevant ones which can provide the most information.

5.3 Feature Selection

It's important to state that the process described on this section was only applied to the numeric features since all of the methods employed only work on that type of data and it wasn't a problem due to the reduced number of categorical features on the dataset.

The importance of feature selection has already been discussed in many works and is a critical process in response modelling (Buckinx et al. 2004; Sing'oei & Wang 2013; Tan et al. 2005). It involves searching among all features for a subset of them that is more relevant for the characterization of the goal attribute. In this thesis only two approaches for this will be considered, filters and wrappers (John et al. 1994). Due to the dataset size, using only wrapper approaches would be impossible so it needed an initial selection by using filter methods.

Filter methods (represented on figure 10) evaluate variable importance independently from the induction algorithm and don't account for relationships between features. They are usually more computationally efficient and require less resources, but make it possible to select important but redundant results that can harm the model's accuracy. On the other hand, wrapper methods (represented in figure 11) work by adding and/or removing predictors to try and find an optimal combination that, when entered into the model, produce the best results.

These methods are computationally heavier and there's an increased risk of over-fitting on complex models (Kuhn & Johnson 2013).

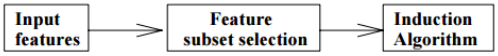

Figure 10 - A representation of the filter model (John et al. 1994)

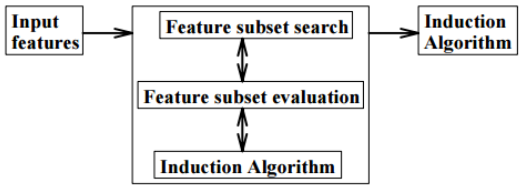

Figure 11 – A representation of the wrapper model (John et al. 1994)

After applying simpler filter methods dealing with data variance and feature correlation the process continues by employing a more advanced filter method (Relief) and a wrapper method (an "ensemble" selection model) and compare their feature selection.

5.3.1 Filter Methods

The first filter method applied to the training dataset was a simple function to remove features with near-zero variance, i.e. features that have very few unique values that occur with very low frequencies. These predictors have a single value for the majority of the samples, with the frequency of other unique values being severely disproportionate. This resulted on 415 numerical variables being removed.

The following step consisted on checking the interquartile range (IQR) of the remaining features. For a numeric attribute, the range is the difference between its minimum and maximum value, and considering the attribute is sorted in an increasing order, it can be divided into equal-sized parts to obtain the quantiles, i.e. the points on which that division occurs. One of the most widely used forms of quantiles is the quartile (represented on figure 12), the division in 4 parts with 3 data points (Han & Kamber 2011). The first quantile cuts off the lowest 25% of the data while the third quantile cuts off the lowest 75%, with the second quantile representing the median, the centre of the data distribution.

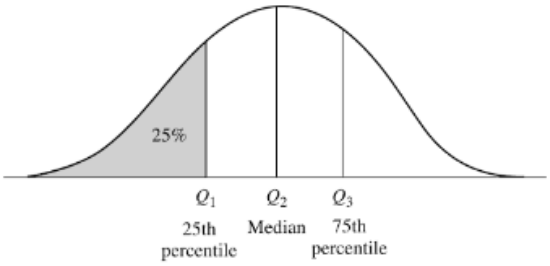

Figure 12 – Plot of a sample distribution with 3 quantiles plotted (Han & Kamber 2011)

The IQR is the distance between the first and third quartiles, $IQR = Q_3 - Q_1$, and thus lower IQRs can be interpreted as features with low variance. R was used to process the IQR of the remaining numeric features on the dataset and noticed that a large portion of them have near zero IQR. Features with low variability won't be useful in discriminating the responders and non-responders so 262 variables were safely removed.

In general, there are good reasons to avoid highly correlated predictors since if two predictors are highly correlated it implies that they are measuring the same underlying information, and they usually provide more complexity to the model than they provide information. Due to the high dimensionality of the dataset a correlation matrix (depicted on figure 13) of 50 randomly selected features was plotted, just to get an idea of the general state of the data.

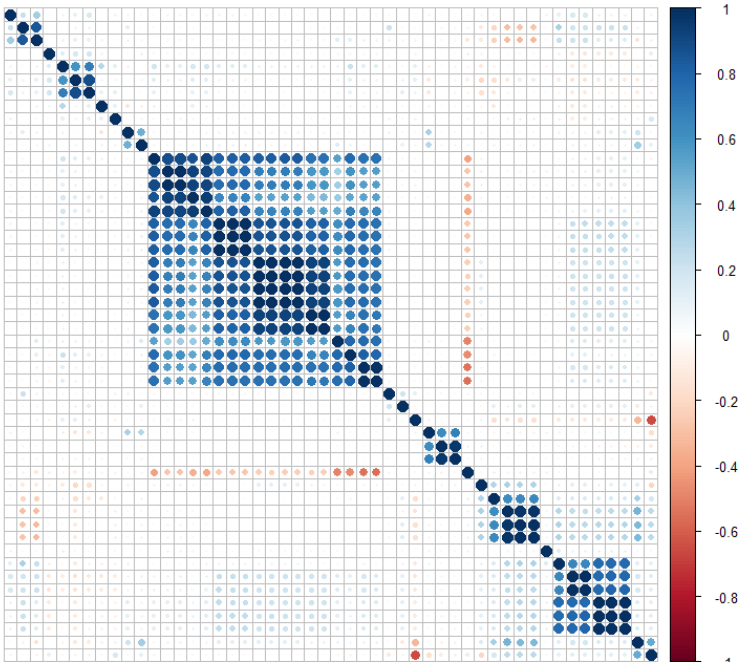

Figure 13 – Correlation matrix between 50 randomly selected features

This type of matrix is built from the training data and each pair of features has a square coloured according to the correlation magnitude. The matrix is symmetric, which means that what appears over the diagonal is repeated under it. The colour code is fairly simple to understand, with dark blue colours meaning strong positive correlations and dark red meaning strong negative correlations. The lighter the tone, the weaker the correlation between those two variables, with the white meaning there is no empirical relationship between them (Kuhn & Johnson 2013). The predictor variables are grouped on the matrix according to a clustering technique so that stronger correlations can be adjacent to each other, which creates those blocks along the diagonal, "clusters" of collinearity (Everitt et al. 2011).

In order to eliminate highly correlated features, a function of package "caret" (Kuhn 2016) was used to find predictors whose correlation was above 0.85 and remove them from the dataset, which led to a total of 538 variables being eliminated.

After applying all these filters, the dataset was left with 474 features out of the original 1934, which is a rather significant reduction but still far from something "manageable" by most classification models.

5.3.1.1 Relief Algorithm

The RELIEF algorithm evaluates the relevance of each feature by comparing how their values distinguish between neighbouring instances of the same and different classes. The algorithm requires a set of randomly selected instances and trough them it searches for the k nearest neighbours from the same class and k from each of the other possible classes. Next, it updates the feature quality information by increasing the predictive value of a feature if it feels that feature separates instances with different classes well or decreasing it on the contrary. The whole process is repeated several times until the result is reached (Kuhn & Johnson 2013).

One of the reasons for selecting this method, besides the fact that it was researched during the state of art, was that it's complexity scales well to large feature datasets compared to other methods (Saeys et al. 2009).

For this project, the RELIEF algorithm ran with random samples of size 22, 5 neighbours and selected the top 30 features.

5.3.2 Wrapper Methods

By this point the effects of individual independent variables were already studied so it's time to discuss their interactions, and the next step is selecting the features that together can capture user response in an effective manner. The product of this was an ensemble feature selection method that will be compared with the filter methods across different classification algorithms.

Similar to what happens with ensemble learning, where multiple classifiers are combined to attain a more effective classifier, the same can be done with multiple feature selectors to attain better features (Saeys et al. 2009). The proposal for this thesis is an ensemble feature selection method based on the combination of multiple Random Forest feature selection models.

Ensembles consist on building a set of predictive models that classify new cases using some form of averaging of the predictions from these models, and they are known for often performing better than the individual models that form the ensemble. The key to this process is the difference between the models, whether it relies on different model parameter settings or considering different predictors for each model (Saeys et al. 2009).

For this case the chosen option was using different samples to obtain each model. This approach works better if the data from which each model is selected is highly redundant, and it was assumed that the necessary degree of redundancy on the datasets would be achieved using k-fold method.

The first step is determining the adequate number of features (N) for the dataset to ensure a better prediction. In order to do that, the random forest algorithm is applied to all data using function “rfcv” from package “randomForest” (Liaw & Wiener 2002). The algorithm measures the importance of the features by randomly exchanging one in the “out-of-bag” samples and calculating the percent increase in misclassification when compared to the rate with all variables intact. After obtaining the importance score of the features, the most important ones are chosen using a backward elimination method. The process of selection consists on calculating the cross-validated prediction error of models with successively reduced number of predictors (ranked by variable importance) following a nested cross-validation procedure (Breiman 2001).

Next the data is split into k-folds that are used to build k different feature selectors. The value of k used was 3, due to the computational load of the process. The N most important features of each of the 3 folds are selected and for each variable the number of times it has been selected by the k feature selectors is registered. The N features with the greater count are chosen, which leads to a result with the consistently more important features across all folds.

As previously mentioned the objective now is to determine a good number of predictors that can represent the best characterization of responders/non-responders. R was used to plot a graph depicting the classification error rate according to the number of selected features. As seen on figure 14, the error generally decreases as the number of predictors increases. Although the minimum error is obtained with 169 predictors, the difference between that error and the one using 30 predictors is not significant, and since the objective is to reduce the number of features to a minimum, only the first 30 predictors with the highest importance would be selected to build the models.

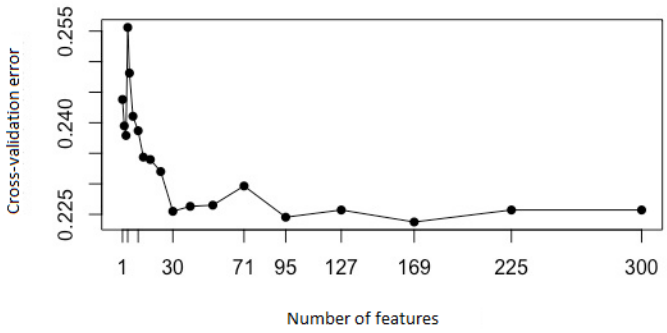

Figure 14 – Classification error rate with different number of features

After an ensemble of 10 random forest iterations, the 30 most relevant features were obtained, and they were all numeric.

The 30 features selected by the filter method RELIEF and by the wrapper method of ensemble feature selection were different, without a single feature in common.

5.4 Class Balancing

In the context of this project, where the dataset has a very reasonable size, it makes sense to select the stratified hold out method to create a training and a test partition. This method consists of randomly splitting the dataset in two disjoint partitions (usually in a 70%-30% ratio) while maintaining the initial proportion of the goal attribute. The 70% will be used to obtain the models while the remaining 30% will be used to test them. But although the test set must maintain the initial distribution of responders/non-responders, the training set will need to be balanced since most learning algorithms cannot perform well with imbalanced data as they usually tend to omit the smaller class, a problem that aggravates in this project considering that the smaller class is the most relevant class (Han & Kamber 2011; Kohavi 1995; Tan et al. 2005).

There are some class balancing processes already proposed and they fall into two main categories: algorithm modification and data balancing, as depicted on figure 14. Methods based on algorithm modification work by inserting an additional specialized mechanism into the original algorithm, by using evaluation metrics more sensitive to the minority class, either by shifting the decision towards that class or attributing different costs to each class. Data balancing methods on the other hand work by building a new dataset in which all classes are well balanced, by under-sampling the majority class and/or over-sampling the minority class (Kang et al. 2012).

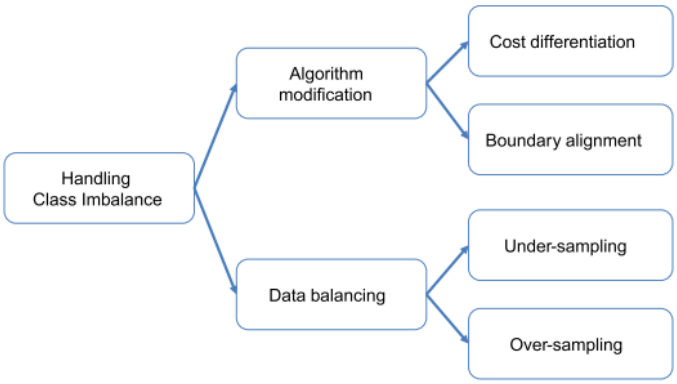

Figure 15 – General overview of the approaches to class imbalance (Kang et al. 2012)

Data balancing methods work independently from classification algorithms, making them universal and able to be combined with any classifier, opposite to algorithm modification methods which work well only with the classifiers for which they were designed. Due to this fact, two processes from the data balancing category were selected: random under-sampling (which is an under-sampling technique, as the name states) and the SMOTE method (an over-sampling technique).

5.4.1 Random Under-Sampling and SMOTE

The process of random under-sampling involves under-sampling, i.e. removing samples from, the majority class samples at random until their quantity matches the minority class numbers. This process is effective on reducing training time but its major flaw is that the class distribution is often distorted due to large numbers of majority class samples being removed, which cannot guarantee a stable response rate due to the randomness of the procedure potentially removing important samples of the majority class, or in this case, the non-responders class (Kang et al. 2012). In the event of that happening the prediction would be harmed because the model wouldn't know how to correctly identify a non-responder and what are the main differences between them and responders.

SMOTE (Synthetic Minority Over-sampling TEchnique), described by Chawla et al. (2002), is a data balancing method based on over-sampling (although it performs some under-sampling as well for increased balance) that synthetizes new cases of the minority class in order to reach a balanced outcome. It allows the user to choose the number of neighbours to consider when creating the new samples. This synthetization process starts by picking a random data point from the minority class and determine the specified number of its near neighbours, thus creating a new data point composed by a random combination of the predictors of both the initially selected data point and determined neighbours. As previously mentioned, this algorithm also does some under-sampling of the majority class in order to help balance the training set (Kuhn & Johnson 2013). This technique is able to preserve the original data distribution but is computationally heavier and requires more time to process due to the increase on the total number of training samples. Additionally, in this context, over-sampling

may provide some wrong information about the customers since it generates virtual responders from a reduced pool of actual responders, which can result in unrealistic customers that could distort the characteristics of respondent customers (Kang et al. 2012).

After the process of feature selection and class balance, a total of 4 datasets are ready to be evaluated by the prediction models. Table 3 presents an overview of the 4 datasets according to the combination of methods used for selection and balance.

Balancing Method	**Feature selection**	
	RELIEF	*Ensemble RF*
Random Under-Sampling	DS1	DS2
SMOTE	DS3	DS4

Table 3 – The datasets resultant of feature selection and class balancing

5.5 Model development

With the four created datasets with different selected features, the next step is to start the development of the prediction models so we can compare them and the feature selection techniques. Once again R was used for the model developing by using a different set of packages for each type of model to build. As mentioned on section 2.5 the chosen prediction methods were four and the goal was to represent a wide range of approaches: two machine learning algorithms, Random Forest and Neural Network, a probabilistic model, Naïve Bayes, and a method based on gradient boosting trees, XGBoost. In order to tune the models the process described in section 3.1.3 was used, k-fold cross-validation, with a k value of 10, used to provide reasonable estimates of uncertainty. Since the research about each of the model types was already described on this thesis, this section will focus only on the commands and tuned parameters used on R to create each model.

The random forest was generated using a function from package "randomForest" (Liaw & Wiener 2002), a package that implements Breiman's original algorithm (Breiman 2001) for classification on R. It is depicted on code 2, where *ntree* represents the number of trees to grow, with the chosen value of 2000.

```
randomForest(target ~ ., data = trainSet, ntree = 2000)
```

Code 2 – The code to generate a random forest model

The neural network model was generated using a function from package “caret” (Kuhn 2016), short for Classification and Regression Training, which consists on a set of functions for training and plotting classification and regression models. This specific function “train” can work with multiple modelling techniques and has the huge upside of automatically choosing the best fit in terms of tuning parameters. The function uses a grid of parameters and trains the model repeatedly with slightly different values on each try. In the end, it calculates the result and chooses the optimal combination of parameters to generate the model. In this case, the neural network was tuned over the number of units in the hidden layer (ranging from 1 to 5) as well as the amount of weight decay (with 4 different values: 0, 0.1, 1, 2). This is depicted on code 3 below. The data was also centred and scaled prior to fitting, so that attributes whose values are large in magnitude do not dominate the calculations.

```
nnetGrid <- expand.grid(.size = 1:5, .decay = c(0, .1, 1, 2))
```

Code 3 – Code for parameter tuning the neural network model

The “train” function is depicted on code 4, where *method* represents the chosen model (which in this case is neural network), *metric* defines what evaluation process will be used to select the optimal model, *preProc* is pre-processing that can be applied before fitting (the centring and scaling already mentioned), *tuneGrid* consists of a data frame with possible tuning values, *trace* is disabled due to tracing optimization not being necessary, *maxit* is the number of maximum iterations to execute and was defined 2000, *MaxNWts* defines the maximum allowable number of weights, which is calculated from both the hidden layer size and the training set size, and *trControl* receives a list of values that in this case configure the function to compute sensitivity, specificity and the area under the ROC curve.

```
train( x             = trainSet[,1:ncols-1],
       y             = trainTarget,
       method        = "nnet",
       metric        = "ROC",
       preProc       = c("center", "scale"),
       tuneGrid      = nnetGrid,
       trace         = FALSE,
       maxit         = 2000,
       MaxNWts       = numWts,
       trControl     = ctrl)
```

Code 4 – Code to generate a neural network model

The naïve Bayes model was generated using a function from package “e1071”, a group of miscellaneous functions of the department of statistics, probability theory group from the Technological University of Vienna (TU Wien), which includes an implementation of the standard naïve Bayes classifier. The function itself is depicted on code 5, where *laplace* represents the value to use for Laplace correction, which is needed to correct probabilities that would be zero if some predictor does not have any samples on the training set for a specific class (Kuhn & Johnson 2013). This value is usually between 1 or 2, with the latter being selected for this case.

```
naiveBayes(target ~., data=trainSet, laplace = 2)
```

Code 5 – Code to generate a naïve Bayes model

Finally, to generate the XGBoost model, a function of package "xgboost" was used. The package is the R interface of a gradient boosting framework and can automatically do parallel computation on a single machine to increase speed. It provides a simple interface for training a XGBoost model, the function depicted on code 5, where *data* and *label* are the training set and the class to be predicted respectively, *nrounds* is the max number of iterations and was defined as 15000, *objective* is the desired objective function which in this case is logistic regression for classification, *max_depth* is the maximum depth of the tree, *eval_metric* is the selected evaluation metric, which for this case just like with neural networks was defined area under the ROC curve, *subsample* and *colsample_bytree* deal with subsample ratio of the training instance and columns respectively to prevent overfitting (although it implies an increase of *nround* to achieve the desired effect), *verbose* allows the algorithm to print information about the training in real time, *eta* is used to control the learning rate and also used to prevent overfitting, the smaller it is the largest *nrounds* has to be, which makes it slower to compute but results in a more robust model against overfitting.

```
xgboost( data         = data.matrix(trainSet),
         label        = trainSet$target,
         nrounds      = 15000,
         objective    = "binary:logistic",
         max_depth    = 15,
         eval_metric  = "auc",
         subsample    = 0.7,
         colsample_bytree= 0.5,
         verbose      = 1,
         eta          = 0.0025)
```

Code 6 – Code to train a xgboost model

The parameters chosen for XGBoost were result of both trial-and-error and research (Jain et al. 2015; Zhang 2015).

In all, sixteen models were built, with two different feature sets and two different data sampling approaches. The next step is to run each model on the independent test dataset (the 30%) and measure the overall predictive performance of each model.

6 Solution Evaluation

The next step is to test the models and evaluate them. As previously mentioned the methods used to evaluate a data mining project such as this one are specific tools and methodologies instead of standard tests. The evaluation processes to be employed are those described in chapter 3, namely the specificity and sensitivity measures as well as the ROC curve and measuring the area under the curve as well. They were selected due to their proven efficiency in model comparison (as proven earlier across the state of art analysis).

The prediction results of all classification models are presented on table 4 in terms of area under the ROC curve (AUC). Each model attempts to correctly identify existing responders in the test dataset of 72615 customer profiles.

Balancing Method	**Classifier**	**Feature Selection**	
		Relief	**Ensemble RF**
Random Under-Sampling	*Random Forest*	0.675	0.721
	Neural Network	0.693	0.708
	Naïve Bayes	0.642	0.64
	XGBoost	0.71	**0.736**
SMOTE	*Random Forest*	0.651	0.7
	Neural Network	0.67	0.655
	Naïve Bayes	0.672	0.716
	XGBoost	0.692	**0.73**

Table 4 – Performance comparison of prediction models in terms of AUC measure

From table 4 we can reach some conclusions about the efficiency of the tested prediction models as well as the balancing methods and feature selection processes. The first thing to notice is that for almost all classifiers, the ensemble of random forests was the most effective feature selection method. It was a lengthy process and more complicated to assemble than its counterpart on this test but in the end the result was worth the effort. In respect to balancing

methods it's evident that random under-sampling surpassed SMOTE and is apparently the best choice for this kind of project. This weaker performance of SMOTE might have to do with the fact that even after the pre-processing made on the data, there are still a lot of missing values, which can make it hard for the algorithm to find truly nearest neighbours for a minority sample. And finally, the most effective classifier when dealing with heavy datasets such as this one is XGBoost, surpassing the others in all four conditions, independently of feature selection or balancing method. The outcome was expected as this technology is recent and has been gaining more reputation for uses in datasets like this with high amounts of data and unbalanced classes. This shows that gradient boosted trees and this implementation have high potential and can achieve even better results with further tuning and modifications.

For a visual representation, the ROC analysis of the best models of each classifier (represented in bold on table 4) are depicted on figure 16.

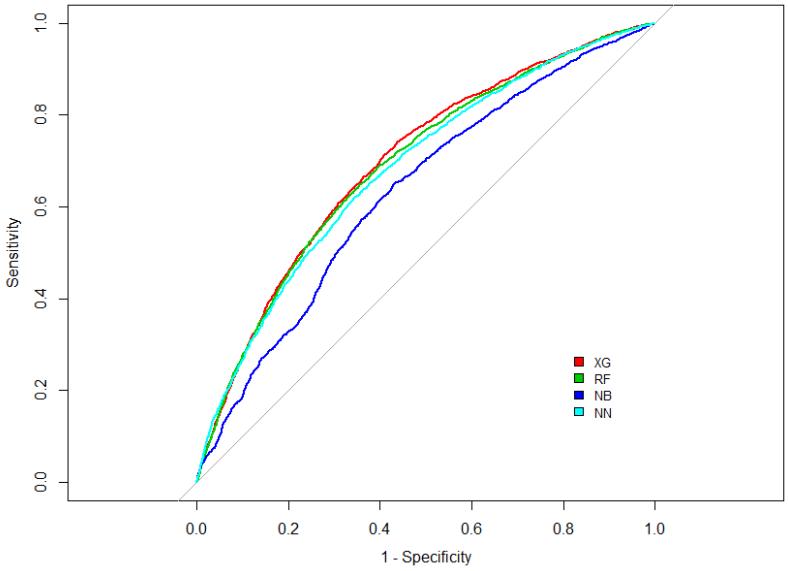

Figure 16 – ROC curves of classification models: XGBoost (XG), Random Forests (RF), Naïve Bayes (NB) and Neural Network (NN)

As shown in figure 16, the worst algorithm to model this data is NB. And although there may not be such a significant difference between the other four, XGBoost comes out on top, even if by small margin, and the results on table 4 confirm it. As such, concerning this dataset it is possible to say that the combination of the proposed ensemble of random forests for feature selection, with random under-sampling for class balancing and using XGBoost as the prediction model, achieves the best performance.

7 Conclusions

Throughout this document, four classifiers were selected and studied to try and solve a classification problem related to direct marketing which involved a rather extensive dataset in terms of both samples and features. Two processes of feature selection were discussed as well as two data balancing methods, to try and tackle the dataset as efficiently as possible.

After a total of sixteen possible combinations of methods were analysed, the combination of an ensemble of random forests for feature selection, with random under-sampling for class balancing and XGBoost as the prediction model, proved to be the most effective solution to this specific problem.

Nevertheless, this was observed from the specific experiments described throughout the document, and it's important to note that the performance of sampling approaches is largely data dependent.

The results achieved with this project lead to new paths for further research. With the proposed feature selection method and the positive results it achieved, the next step should be trying to tune it and improve it now that it proved itself as a viable alternative to known feature selection algorithms. Also, XGBoost allows a multitude of parameters to be tuned when training the classifier and although the proposed values gave it an advantage over the other alternatives, this thesis only scratched the surface of what it has to offer in terms of parameter flexibility, which could allow for some interesting developments later on. Finally, this research would be worth extending to up-lift modelling (Radcliffe & Surry 2011), which contrary to traditional response modelling techniques, is able to skip customers who would already buy the product regardless of whether they are targeted by a marketing campaign or not, targeting only those who will only buy it if they are directly in contact. Class imbalance is a problem for this type of modelling so the proposal made on this document could help extend its research even further.

References

Babajide Mustapha, I. & Saeed, F., 2016. Bioactive Molecule Prediction Using Extreme Gradient Boosting. *Molecules*, 21(8), p.983. Available at: http://www.mdpi.com/1420-3049/21/8/983.

Baesens, B. et al., 2002. Bayesian neural network learning for repeat purchase modelling in direct marketing. *European Journal of Operational Research*, 138(1), pp.191–211.

Bose, I. & Chen, X., 2009. Quantitative models for direct marketing: A review from systems perspective. *European Journal of Operational Research*, 195(1), pp.1–16. Available at: http://dx.doi.org/10.1016/j.ejor.2008.04.006.

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7), pp.1145–1159.

Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), pp.5–32.

Buckinx, W. et al., 2004. Customer-adapted coupon targeting using feature selection. *Expert Systems with Applications*, 26(4), pp.509–518.

Chawla, N. V. et al., 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, pp.321–357.

Chen, H., Chiang, R.H.L. & Storey, V.C., 2012. Business Intelligence and Analytics: From Big Data To Big Impact. *Mis Quarterly*, 36(4), pp.1165–1188.

Chen, T. & Guestrin, C., 2016. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. New York, New York, USA: ACM Press, pp. 785–794. Available at: http://arxiv.org/abs/1603.02754.

Chen, T. & He, T., 2015. Higgs Boson Discovery with Boosted Trees. *JMLR: Workshop and Conference Proceedings*, 42(May 2014), pp.69–80.

Chesbrough, H., 2002. The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology spin-off companies. *Industrial and Corporate Change*, 11(3), pp.529–555. Available at: http://icc.oupjournals.org/cgi/doi/10.1093/icc/11.3.529.

Coussement, K., Harrigan, P. & Benoit, D.F., 2015. Improving direct mail targeting through customer response modeling. *Expert Systems with Applications*, 42(22), pp.8403–8412. Available at: http://linkinghub.elsevier.com/retrieve/pii/S095741741500456X.

Cui, G., Wong, M.L. & Lui, H.-K., 2006. Machine Learning for Direct Marketing Response Models: Bayesian Networks with Evolutionary Programming. *Management Science*, 52(4), pp.597–612.

Elith, J., Leathwick, J.R. & Hastie, T., 2008. A working guide to boosted regression trees.

Journal of Animal Ecology, 77(4), pp.802–813. Available at: http://doi.wiley.com/10.1111/j.1365-2656.2008.01390.x.

Everitt, B.S. et al., 2011. *Cluster Analysis*, Available at: http://www.springerlink.com/index/10.1007/BF00154794.

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P., 1996. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), p.37.

Govidarajan, M., 2013. A Hybrid Framework using RBF and SVM for Direct Marketing. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 4(4), pp.121–126.

Han, J. & Kamber, M., 2011. *Data Mining: Concepts and Techniques*, Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: http://link.springer.com/10.1007/978-3-642-19721-5.

Heilman, C.M., Kaefer, F. & Ramenofsky, S.D., 2003. Determining the appropriate amount of data for classifying consumers for direct marketing purposes. *Journal of Interactive Marketing*, 17(3), pp.5–28. Available at: http://dx.doi.org/10.1002/dir.10057.

Jain, A., Menon, M.N. & Chandra, S., 2015. Sales Forecasting for Retail Chains. , pp.1–6.

John, G., Kohavi, R. & Pfleger, K., 1994. Irrelevant Features and the Subset Selection Problem. *… Learning: Proceedings of the …*, pp.121–129.

Kaggle, 2016. About Kaggle. Available at: https://www.kaggle.com/about [Accessed October 8, 2016].

Kang, P., Cho, S. & MacLachlan, D.L., 2012. Improved response modeling based on clustering, under-sampling, and ensemble. *Expert Systems with Applications*, 39(8), pp.6738–6753. Available at: http://dx.doi.org/10.1016/j.eswa.2011.12.028.

Kira, K. & Rendell, L., 1992. A practical approach to feature selection. *Proceedings of the Ninth International Conference on Machine Learning*, pp.249–256.

Koch, K.-R., 1990. Bayesian Inference with Geodetic Applications. In Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 4–8. Available at: http://dx.doi.org/10.1007/BFb0048702.

Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence*, 14(12), pp.1137–1143.

Kononenko, I., 1994. Estimating attributes: Analysis and extensions of RELIEF. *Machine Learning: ECML-94*, 784, pp.171–182. Available at: http://www.springerlink.com/index/10.1007/3-540-57868-4.

Kuhn, M., 2016. The Caret Package. Available at: http://topepo.github.io/caret/index.html [Accessed October 21, 2016].

Kuhn, M. & Johnson, K., 2013. *Applied Predictive Modeling*, Available at: http://link.springer.com/10.1007/978-1-4614-6849-3.

Kwon, Y.-K. & Moon, B.-R., 2001. Personalized Email Marketing with a Genetic Programming Circuit Model. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2011)*, pp.1352–1358.

Langley, P., Iba, W. & Thompson, K., 1992. An Analysis of Bayesian Classifiers. *AAAI-92 Proceedings*, (October 2016), pp.223–228.

Lee, H. et al., 2010. Semi-Supervised Response Modeling. *Journal of Interactive Marketing*, 24(1), pp.42–54. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1094996809000917.

Liaw, A. & Wiener, M., 2002. Classification and Regression by randomForest. *R news*, 2(December), pp.18–22.

Ling, C.X. & Li, C., 1998. Data Mining for Direct Marketing: Problems and Solutions. *Fourth International Conference on Knowledge Discovery*, 98, pp.1–7.

Loshin, D., 2013. *Business Intelligence: The Savvy Manager's Guide*, Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Business+Intelligence:+The+Savvy+Manager's+Guide#4.

Management, P. et al., 2000. Recognizing End-User Transactions in Recognizing End-User Transactions in Performance Management. , (July).

Morimoto, M. & Chang, S., 2006. Consumers' Attitudes Toward Unsolicited Commercial E-mail and Postal Direct Mail Marketing Methods. *Journal of Interactive Advertising*, 7(1), pp.1–11. Available at: http://jiad.org/download?p=83.

Moro, S. & Laureano, R.M.S., 2011. Using Data Mining for Bank Direct Marketing: An application of the CRISP-DM methodology. *European Simulation and Modelling Conference*, (Figure 1), pp.117–121.

National, F., 2005. Bayesian neural networks. , pp.1–4.

Olson, D.L. & Chae, B., 2012. Direct marketing decision support through predictive customer response modeling. *Decision Support Systems*, 54(1), pp.443–451. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167923612001881.

Osterwalder, A. & Pigneur, Y., 2010. *Business Model Generation*, Available at: http://www.amazon.com/Business-Model-Generation-Visionaries-Challengers/dp/0470876417.

Osterwalder, A. & Smith, A., 2016. Strategyzer. Available at: https://strategyzer.com [Accessed February 14, 2016].

Provost, F., Fawcett, T. & Kohavi, R., 1998. The Case Against Accuracy Estimation for Comparing Induction Algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning*, pp.445–453.

Radcliffe, N. & Surry, P., 2011. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic …*, (section 6), pp.1–33. Available at:

http://www.stochasticsolutions.com/pdf/sig-based-up-trees.pdf.

Rish, I., 2001. An Empirical Study of the naive Bayes Classifier. , (January 2001).

Roberts, M.L. & Berger, P.D., 1999. *Direct Marketing Management*,

Rodrigues, F., 2015. Avaliação de Modelos. *Descoberta de Conhecimento*.

RStudio, 2016. About - RStudio. Available at: https://www.rstudio.com/about/ [Accessed October 8, 2016].

Saeys, Y., Abeel, T. & Van de Peer, Y., 2009. *Robust Feature Selection Using Ensemble Feature Selection Techniques*, Available at: http://linkinghub.elsevier.com/retrieve/pii/S1532046409000033.

Sing'oei, L. & Wang, J., 2013. Data Mining Framework for Direct Marketing: A Case Study of Bank Marketing. *IJCSI International Journal of Computer Science Issues*, 10(2), pp.198–203. Available at: http://ijcsi.org/papers/IJCSI-10-2-2-198-203.pdf.

Springleaf, 2016. Springleaf - About Us. Available at: https://www.springleaf.com/about-us [Accessed August 29, 2016].

Steinwart, I. & Christmann, A., 2008. *Support Vector Machines* Springer Science & Business Media, ed.,

Stöckli, P.L. & Tanner, C., 2014. Are integrative or distributive outcomes more satisfactory? The effects of interest-based versus value-based issues on negotiator satisfaction. *European Journal of Social Psychology*, 44(3), pp.202–208. Available at: http://doi.wiley.com/10.1002/ejsp.2003.

Suh, E. et al., 2004. A prediction model for the purchase probability of anonymous customers to support real time web marketing: A case study. *Expert Systems with Applications*, 27(2), pp.245–255.

Suman, M., Anuradha, T. & Veena, K.M., 2012. Direct Marketing With the Application of Data Mining. , 2(1), pp.41–43.

Tan, P.-N., Steinbach, M. & Kumar, V., 2005. *Introduction to Data Mining*,

Williams, C., 2008. Support Vector Machines. , 1(October), pp.1–8.

Zhang, O., 2015. Open Source Tools and DS Competitions. *Open Data Science Conference - Boston 2015*.