Tunisian Republic
Ministry of Higher Education and Scientific Research
University of Carthage - Higher School of Statistics and Information Analysis

Graduation Project Report submitted to obtain the degree of
National Diploma of Engineering in Statistics and Data Analysis

Machine Learning Scoring Methodology: Application and Exploitation

Submitted by

Fatma Bennour

Completed at

Defended on 15/07/2020 in front of the committee composed of

Mokhtar Kouki, Professor, ESSAI (President)

Ghazi Belmufti, Teacher Assitant, ESSAI (Reviewer)

Aicha El Golli, Teacher Assitant, ESSAI (University supervisor)

Amine Ben Bey, Data Scientist, D-AIM (Company supervisor)

Academic year 2020

To my dear parents for each of their sacrifices,
To my father's soul,
To my mother, my reason for being,
To all my family and friends,
And finally, to him.

Acknowledgment

I would like to express my special appreciation and thanks to M. Hichem ABDELHAK, Head of D-AIM Tunisie & Telecom Business Unit, and M. Amine Ben Bey, Data Scientist and my technical supervisor, for the continued support and the confidence they have given me during this internship.

Special thanks are also due to my university's supervisor Prof. Aicha El Golli, for having directed this work, supported me throughout the project as well as for her precious advice.

My deepest thanks and gratitude are also due Prof. Mokhtar Kouki for agreeing to chair the jury and Prof. M.Ghazi Belmufti for kindly accepting to evaluate my modest work and giving me the honor to be my jury members. I hope the defense will be an enjoyable moment.

Special gratitude goes to all my teachers at the Higher School of Statistics and Information Analysis for guiding me in my learning journey.

Finally, I would also like to express my gratitude to the entire D-AIM team for their warm welcome and for making this internship a pleasant experience.

Abstract

Predictive scoring and individualized targeting are increasingly adopted by companies of all fields and sizes. However, achieving the perfect targeting strategy is beyond the reach of many among them. In fact, the challenge is to find and adopt the most appropriate scoring methodology for the business issue at hand.

In this context, this work was carried out within "D-AIMTunisie" consulting company specialized in relationship marketing, and is part of an end-of-study project with a view to obtaining an engineering degree in statistics and information analysis.
The objective of which is to apply a predictive scoring methodology in order to propose a novel approach for combining predictive scores and evaluate its performance comparing with the basic currently used one.
As well as establishing a mathematical and scientific ground of comparing the probability generated by the predictive scoring algorithms in order to select for each individual, the opportunity that best fit his consumption.

Keywords: Machine Learning, Ensemble Learning, Predictive scoring, Client Targeting, Probability Theory, Standardization.

Contents

List of Figures

General Introduction

Predictive analytics, which has been around for decades, is now a mature technology. More and more companies from different fields, are using it to stimulate innovation in all areas, provide the opportunity to put new ideas into play as well as accelerate the transformation of targeting strategies. In fact, predictive analysis states for the use of the proper Artificial Intelligence algorithm and technologies in order to analyze the data at hand and extract the accurate predictions and hypotheses.

The reason predictive analysis is more and more adopted by companies is that this approach helps them, not only, to predict the future preferences, choices and offers of the customer, but also, it gives them the ability to influence the future portfolio of offers of the customer by adopting and refining the targeting strategy to meet every client's need.
However, this mission remains difficult for telecommunication companies, considering the tremendous increase in the size of real life and time data to manipulate and analyze. But luckily, along with the enormous amount of data, we are supported by more advanced algorithms, high end computing power and storage that can deal with that huge data size.

In this context, several AI and ML algorithms have become the heart of customer journey management strategies. Thanks to its analytical power, this advance in artificial intelligence allows companies to build an almost complete, objective and individual vision on each of its customers, which has given them the advantage of adopting an individualized targeting strategy i.e. the communicating with each consumer in a unique way depending on his current preferences. In other words, the customer is then placed at the center of the challenges.

In fact, this project aims to evaluate two business practices of the company; the first one is established in association with a telecommunications company facing a particular Business problematic and desiring to seek an innovative approach of combining two predictive scores. While the other is dedicated to enhance the exploitation of the Machine Learning outputs in order to refine the individualized targeting strategy.

In this project, our objective is, first of all, to democratize the different Machine Learning analytical techniques used by predictive scoring practitioners and to justify their importance. In addition, we will review the arbitration performed by technical and commercial experts as well as the different stages of predictive score creation in order to respond to a concrete business problem.
To another extent, we will introduce, in the most broad and general way, the statistical and probability concepts useful to further comprehend the establishment of a proper targeting strategy based on the predictive scoring algorithms.

This end-of-study project report allows readers in a first chapter to know the host organization, to summarize some basic concepts and to interact with the problematic of the project.
In the second chapter, we will study predictive methods of supervised classification and develop the concepts that criticize the performance and robustness of a model. As for the next chapter, we will explain the different probabilities fundamental concepts and their use in predictive scoring.
Regarding the last chapter, we will display in detail the process followed to deploy predictive scores by competing two commercial approaches for customer targeting and validating them using technical and marketing performance evaluation metrics.

This work infers with a general conclusion and at the same time puts forward possible perspectives with regard to this project.

Chapter 1

Conceptual Framework

In this chapter, we will first introduce the host company; D-AIM. Then, we will throw a line upon some basic concepts used in our work. Finally, we will present the problem and the main objectives of this project.

1.1 The Host Company

1.1.1 D-AIM, by Inbox

Founded in 2001, Inbox, or D-Aim currently, has become accustomed to combining commercial expertise with the power of AI and Data Science, allowing her to build high level Marketing Strategies in several areas, such as Retail, Telecommunication, Media and Banking. Thanks to the competency of its collaborators and business experts, D-AIM is an international actor present in 5 countries, namely, France, Tunisia, Canada, Morocco and Belgium, intervening with projects in more than 20 countries [1].

Figure 1.1: Company's Logo [1]

1.1.2 Fields of expertise

D-AIM is a software creator and editor. Its solutions are dedicated to covering and respond to all issues related to customer marketing [2].

1.5emRegardless the needs and the maturity level of its partners, D-AIM combines the expertise of 3 rich and diversified profiles; the web developers, the data and studies managers and the marketing experts. All for the sake of manufacturing highly effective customer knowledge indicators, designed to assist on 3 different axes, being:

- The deterministic marketing: Thanks to a set of behavioral monitoring and profiling indicators and eligibility rules allowing to understand and analyze the client's actual and historical attitude.
- The predictive marketing: Thanks to the scoring methodology developed by D-AIM, in order to anticipate the client's future behavior and tendencies. Using advanced machine learning algorithms.
- The individualized marketing: the result of the association of deterministic and predictive marketing indicators, in order to identify, for each profile, the best targeting strategy.

Figure 1.2: D-AIM ecosystem [1]

All these technologies combined have led to several success stories and overcome numerous challenges in all the activity fields of D-AIM, in particular the field of Telecommunications, to which this project will relate [2].

1.2 Project Framework

1.2.1 Study of The Existing

The main aim of any company is to build better client knowledge since it's the ultimate way to guarantee a growing market share. Yet this is a challenging mission for the telecommunication companies, considering the tremendous increase in the size of data, the customer mobility as well as the diversification of offers and services, namely Voice, Data, Rooming, SMS, etc. Therefore, building highly effective and innovative customer knowledge seems to be paramount yet out of reach.

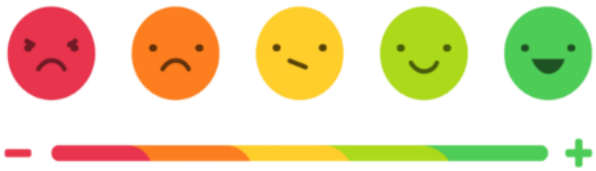
In this regard, numerous telecommunication companies chose to rely on D-AIM's expertise and confine in its AI based marketing strategies, Not only because it responds to different marketing issues and challenges, but also, it is able to generate innovative added value in terms of data exploitation and analysis.

As mentioned, D-AIM helps its partners build a hyper-personalized Customer and Market knowledge, via its algorithms and ultra-powerful software. Its strategies are based on deterministic indicators such as historical activities, buying and browsing behavior, client segmentation, reactions to solicitations and offers etc.

As well as predictive ones which are Artificial Intelligence and Machine Learning based scores designed to predict the potential behavior regarding several problems related to the telecommunication field [3].
So what's predictive scoring? Why use scoring in marketing strategies?

Predictive Scoring

Figure 1.3: Predictive scoring [3]



Predictive scoring is the process of classifying customers according to Machine Learning generated Probabilities, describing the possibility of responding to a specific marketing solicitation. In simpler words, a score is the probability of responding to a solicitation [4].

Customer scoring is based on two main ideas:

- Not all customers are alike, their likelihood and frequency of purchase, their average basket, their preferences are different. Therefore, a higher score is assigned to customers who are more likely to respond to the observed phenomenon.
- Marketing is all about prioritization. Customer scoring allows focusing the marketing efforts on profitable client profiles.

Accordingly, predictive scoring can be used to determine the customers who are most likely to respond favorably to an offer, to identify risky ones, and as a result, to optimize the earnings of campaigns by selecting the customers for whom the expected value of gain is greater than solicitation cost only [4].

In this regard, D-AIM has developed a predictive scoring methodology (will be detailed in further chapters) as a tool of refinement of targeting strategies. The processed scoring themes are:

Appetency: Customers with usage during the scoring period, yet no usage beforehand.

Stability: Customers with stable revenue and consumption thresholds before and during the scoring period.

Fragility: Former users before the scoring period yet with no usage during the scoring period.

Upsell: Customers who increased their revenue threshold during the scoring period.

Churn : Customers without usage (data, voice, SMS) or revenue during the scoring period.

Balance : Customer who exceeds a specific usage threshold during the scoring period.

For each of the binary phenomena mentioned above, the idea is to train a Machine Learning classifier predicting the probability of belonging to one of the two predefined groups.

Machine Learning

Machine learning is a subset of Artificial Intelligence; the science of training machines to perform tasks that usually require human intelligence, by means of specific tools and algorithms[5].
In simpler terms, machine learning is the fact of acquiring to a computer the ability to extract and learn patterns from raw data and represent it in a model, useful to predict or infer from similar data that have not been modeled yet [6].

Figure 1.4: Data modeling [6]

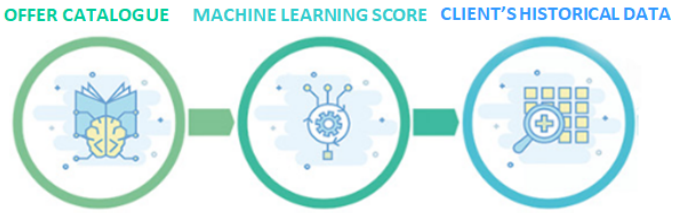
Various machine learning algorithms have been proposed to cover the wide variety of data types and problems. The choice of the appropriate method depends on the problematic, the nature of the data as well as the expected result.
These algorithms can be divided into three categories, namely supervised learning, unsupervised learning and reinforcement learning. The differences between these categories and their associated algorithms will be discussed in subsequent chapters.

Customer Targeting Strategy

Client targeting is the process of selecting only the suitable individuals for a product or service. If an appropriate targeting strategy is adopted, many advantages will be guaranteed, including optimizing the marketing action's costs by delimiting the number of prospects as well as avoiding the cannibalization risks i.e. prioritizing the cheaper opportunities over pricier ones.

In this respect and thanks to its expertise in marketing and AI, D-AIM has developed a solid and reliable targeting strategy: NBA or Next Best Action, which is the fact of creating and allocating to each client the opportunity that best fits his behavior, after exploring three different axes illustrated in the figure1.5:

Figure 1.5: D-AIM Targeting Process [6]



- The machine learning score results that help quantify the potential preference regarding a defined offer or theme.
- The customer's current characteristics and aspects.
- Estimation of the financial profit relying on the offer catalogue.

The concrete conception of this targeting strategy will be further elaborated in the practical chapters.

Given the diversification of offers and services in the telecommunications field, customers often appear to be eligible for two or more opportunities. Therefore, the choice of the appropriate one is based on a comparison between the associated scores.

1.2.2 Project Framework

This project is established in cooperation with a telecommunication operator who wants to further enhance the knowledge of its customers, more precisely the data consumer ones. The process of obtaining data is as follows:
First of all, the telecommunication operator engages of sending daily customer data that depict and details their actions.

Afterwards, due to small modifications, the raw data is redistributed into a unique and standard format: the Global Telco Data Structure, allowing the definition of the customer's profile, usage and revenue, as well as according a defined architecture to the manipulated data in a way that each Datamart is identified by the customer's ID: CONTRACT-ID as well as the date of the action: DATE and contains all the KPI necessary to describe the type of the action.
Moreover, we have three main types: SMS, Voice and Data. Each one could be taken through PAYG- Pay as you go; a consumption directly from the balance, without resorting to packages. Or through BUNDLE; a consumption generated by the purchase of packages

or offers.

For the sake of our project, we will study in depth consumption of Data, with a focus on the usage of Bundles with the aim of upgrading or upselling the potential bundle threshold for the eligible clients. Given that, generally, Bundles, or packages, allow the operator to conclude the potential usage in a specific period of validity. This approach allows to push customers to consume and generate income.
The idea is simple; we intend to delimit the customers who are likely to maintain their data consumption, or in D-AIM terms "Non-Fragile Data", and at the same time "Appetent" to one of the four Data destined Bundles or offers, So as to increase or Upsell their revenues by refining the targeting strategy

Accomplishing this step requires combining scores predicting different events. In our case, the scores to combine are in one hand, Non-Fragile Data and in the other hand, Appetent to a Data bundle. The figure 1.6 bellow will further explain the process.

Figure 1.6: Scores to Combine

Deploying these scores requires a different type Datamart that is identified exclusively by the customer's ID. In the respect, we proceed by aggregating the different KPIs according to time variables, i.e. day, week, month, which allows us to introduce the concept of seasonality. We therefore guarantee a detailed description of the client's profile as well as the event to be predicted.

The description of this Datamart will be detailed in the practical chapter.

Our work aims, also, to propose a methodology that allows, for each client, the comparison between the different scores, i.e. the Machine Learning generated probabilities, in order to choose the opportunity that suits him the most.

1.3 Problematic and Objectives

For all the purposes mentioned above, D-AIM proposed this End of Studies Project in order to assist answering the following problematic:

What would be the best practice of combining scores, from both the technical and business perspectives, and how can we quantify the generated financial profits?

And once combined, how to decide on the best comparison methodology between the different opportunities a client is eligible for, so he can be granted the opportunity that best suits his current preferences.

1.4 Conclusion

To answer these questions, the architecture proposed for this project is composed of two major parts. In the first one we will manipulate theoretical knowledge on Machine Learning, as well as the current scoring technique of D-AIM.

This will allow us, first of all, to demonstrate the intuitive technique of combining two scores which consists on deploying two independent Machine Learning algorithms. As well as propose a new approach that reposes on deploying a unique Machine Learning model designed to predict the occurrence of both events simultaneously.
Finally, we will establish a comparison between both approaches and select the one that assures better business revenues.

As for the second one, we will go through the different notions of probability theory in order to establish the proper methodology of comparing the scores generated probabilities and as a result refine the current marketing strategy.

Chapter 2

THEORETICAL FRAMEWORK PART1

Global Introductions to Machine Learning

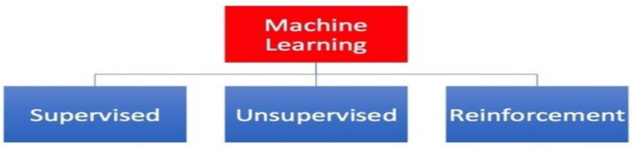
After establishing the conceptual framework, it is paramount to dedicate this chapter to present the theoretical knowledge on Machine Learning, since it occupies a large part of this project.

2.1 Definition of Machine Learning

Machine Learning is the fact of teaching a computer program or algorithm how to gradually improve a defined task assigned to it. On the mathematical and theoretical side of things, machine learning is a set of algorithms based on statistical, probabilistic and computer sciences, the purpose of which is to iteratively learn from the giving data and search for its hidden useful information [7].

As mentioned in the previous chapter, these algorithms can be classified on three major recognized categories: the supervised, the unsupervised, and the reinforcement learning. (See figure 2.1)

Figure 2.1: Machine Learning Categories [7]



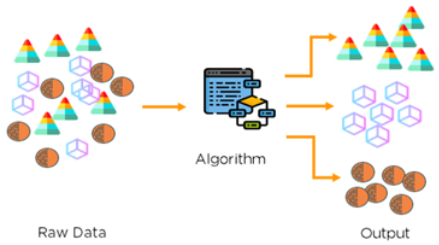
2.2 Types of Machine Learning

2.2.1 The Unsupervised Learning

This type of learning is acknowledged for being data-driven, its main purpose is to identify and group the data points into homogeneous clusters in a completely autonomous way since it treats un-labeled data. Therefore, this type of learning is commonly used in the marketing field for problems such Customers Segmenting, extracting the audience characteristics, etc.

The unsupervised learning functioning is based on a distance measure and iteratively moving similar items closer in order to form the most homogeneous cluster possible. Among its algorithms we can mention K-means and Hierarchical Clustering.
The following figure (figure 2.2) simplifies the operation of an unsupervised learning algorithm.

Figure 2.2: Unsupervised Learning Functioning [7]



2.2.2 The Supervised Learning

This type of learning is rather known as task-driven learning, given the strong dependency on the business side of the treated data. It is the most popular subset of machine learning and therefore the most suited for several marketing purposes, such as identifying highly responding prospects, predicting customer behavior, learning pattern from historical data, etc.

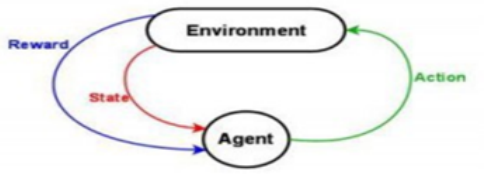
In its learning process, the supervised learning algorithm is served with a set of input features X and expected output variable Y, whose nature will specify the type of the prediction.
In the case of numerical response variable Y, the algorithm is led to estimate the output value and therefore the process is called Regression. Nonetheless, if the response variable Y is categorical, it estimates the class probability and the process is then called Classification.
The figure 2.3 is the representation of a Classification example:

Figure 2.3: Classification Example [7]

2.2.3 The Reinforcement Learning

This type of learning could be simplified as learning from mistakes. The learning process is ensured by providing a kind of signal to the algorithm which associates a correct estimation with a positive signal and a wrong estimation with a negative one [7].
Over time, and by assigning more weight to bad behaviors over the good ones, the algorithm learns to make the correct estimation.
As shown in the figure 2.4, the Reinforcement Learning process is composed of 2 main components, an agent and an environment.

Figure 2.4: Reinforcement Learning Algorithm [7]



As a matter of fact D-AIM's constructed scores reflect events in which the response variable is a binary categorical variable with two known classes: respondent (Y = 1) or non-respondent (Y = 0). In this respect, D-AIM's scoring technique is based essentially on one type of machine learning, namely the supervised learning for binary classification.

As a result, and to remain faithful to the perimeter of our project, this chapter will focus exclusively on the different binary classification algorithms.

2.3 Machine learning methods in supervised learning Classification Algorithms

Several models can be used for binary classification. Among them we can mention Logistic Regression, Gradient Boosting, Support Vector Machines, K-Nearest Neighbors, Decision Tree, Random Forest, Naive Bayes, etc. However we will not cover every one of them. These algorithms can be categorized into two families; Simplistic Methods and Ensemble Methods.

2.3.1 Simple Methods & Ensemble Methods

Simple learning, at the most basic, is the use of one predefined classifier, commonly known as weak learner or classifier, in order to receive and analyse the input data to predict the output values. To sum up, simple learning is the fact of training classifiers allowing to predict better than the random guessing,and enable to increase the accuracy of the final classification.
However, doing better than chance doesn't deny the existence of an important prediction error rate that needs to be reduced in order to guarantee accurate prediction. The proposed solution is the ensemble learning.

While ensemble learning is based on methods that combine multiple weak classifiers in order to obtain better predictive performance than could be obtained from any of the constituent classifiers alone.
The ensemble learning algorithms can be classified into tree major kind: Bagging, Boosting and Stacking. Among them, we find those that function in parallel, and those that follow a sequential approach.

In order to better understand the need of ensemble learning and its different methods, we need to define, first and foremost, the following basic notions.

2.3.2 The Bias-Variance Trade-off

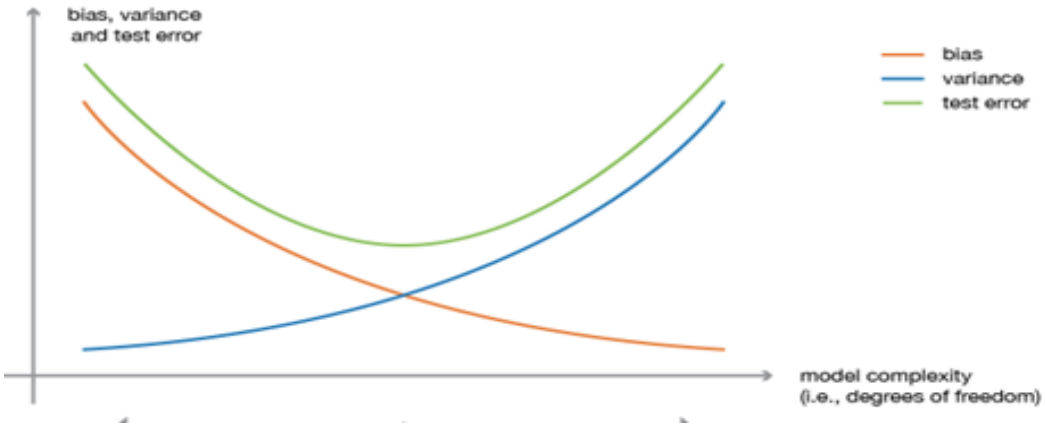
At this point, we realize that supervised Machine Learning algorithms predict future values based on historical data. And who says prediction, says prediction error. This error defined by two essential components:

$$Err(x) = Bias^2 + Variance$$

The bias error is defined by the difference between the model's predicted values and the actual ones and can be caused by either sampling or selecting errors. A high bias could cause the underfitting of the model.

Variance is defined statistically as a measurement of dispersion; it quantifies the spread of the data. Moreover, this notion measures also the sensitivity of a Machine Learning model in a way that if a model is too sensitive, i.e. has a high variance it detects inconsistent patterns which is the definition of overfitting [15].

Figure 2.5: Illustration of the Bias-Variance Ttradeoff [14]



These components are inversely correlated, i.e. if one increases the other shall decrease which creates a trade-off between them known as the Bias-Variance tradeoff, as demonstrated in the figure above.

2.3.3 Weak Learner

A Machine Learning classifier is referred to as a weak learner or a base model when it has a feeble predictive performance due to either a high bias such as the Logistic Regression or Naive Bayes, or a high variance such is the case of Decision Tree algorithms [14].
A weak learner is the building block for any ensemble learning algorithm.

The classic examples of a Weak Classifier or Learner we mention Logistic Regression, Naive Bayes and Decision Tree.

Logistic Regression

Logistic regression is one of the most known classification algorithms in machine learning. It works for both binary and multiple labels. But since D-AIM's scoring models are essentially binary classification models we will focus on binomial logistic regression only.

This algorithm is highly appreciated by marketing practitioners because of its instructive description of cause-effect relationships. Its functioning is similar to the linear regression algorithm, since they both predict problems modeled by the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + .. + \beta_d X_d + \epsilon$$

However, the Logistic Regression classifier do not model the qualitative variable Y but the probability that it will happen depending on the explanatory variables (X_i), as illustrates the following equation:

$$f(x) = p = \frac{1}{1 + e^{-}y} = \frac{1}{1 + e^{(}\beta_0 + \beta_1 X_1 + \beta_2 X_2 + .. + \beta_d X_d + \epsilon)}$$

Then, we can convert the probabilities to predictions using the concept of the threshold value, which defines a response variable Y of either 0 or 1, such as the probabilities above the threshold value tend to 1, and the ones below the threshold values tend to 0.
Binomial logistic regression is also used in the process of scoring and ranking clients, whose more likely to respond favorably to an offer? Who's more likely to refuse it?

Nonetheless, the Logistic Regression model can cause a high bias, especially in the case of numerous explanatory features.To see what the bias term $\beta 0$ represents, we simply need to set all explanatory features to 0. The resulting log odds is the bias term.

Naïve Bayes

Naïve Bayes is a probabilistic classification algorithm based on the assumption of independence among features and used for both binomial and multinomial classification problems.

Naive Bayes model is easy to build and particularly useful for very large data sets, with is the case of the telecommunication field. It operates on the basis of Bayes Theorem which provides

a way of calculating posterior probability, and written mathematically as follows [10]:

$$P(c|x) = \frac{P(x|c) \times P(c)}{P(x)}$$

Where:

- P(c|x) is the posterior probability of class (c, target) given a specific set of features: X.
- P(c) is the prior probability of the class "c", without any knowledge about the features. It is given by:
$$\frac{number of observations belonging to class \backslash c"}{total number of observations}$$
- P(x) (P(x)>0) is the prior marginal probability given a set of features X and presents the probability of similarity to X, i.e. with similar characteristics and features. Given by:
$$\frac{the number of similar observations}{total number of observations}$$
- P(x|c) is the likelihood, it represents the probability that an observation of class c has a set of features similar to X, given by:
$$\frac{number of similar observation among class" c"}{total number of observations in class" c"}$$

However, the Naive Bayes classifier uses a very simple hypothesis function to model the data. As a result, it can not accurately represent many complex situations, so it will produce high bias or errors.

Decision Tree

The decision tree algorithm is the most used algorithm in machine learning. The popularity of this method rests mainly on its simplicity and the fact that it is comparable to human logic which makes the construction process as well as the output results meaningful and easy to interpret. The most effective decision tree algorithms are CART: CART stands for Classification And Regression Trees [11].

In simple terms, a CART decision tree algorithm is a classifier that recursively partitions data according to given criteria in order to form homogeneous groups or classes. It is characterized by the construction of binary trees, that is to say that each internal node has exactly two outgoing edges and therefore two sub-trees or leafs. Furthermore, the divisions are selected in the following way: first, the algorithm finds the variable Xi which has the most predictive effect towards Y. Then, it finds the division threshold that allows the reduction of the impurity index, measured by the Gini or Entropy indexes given respectively by [12]:

- Gini: I(t) = $\sum_{n=1}^{K} pk(1 - pk)$

- Entropy: $\mathrm{E(t)} = -p\log^2(p) - q\log^2(q)$

Finally, we repeat the operation inside each segment, looking for the next important variable until we reach pure knots or the prediction is stopped due to the method's hyper-parameters.

In spite of their popularity, Decision Tree models are complex models with high variance, because if it is very sensitive to (small) changes in the training data. In case of a very large tree, it can basically adjust its predictions to every single input.

Conclusion

For many reasons, decision trees are considered a very good fit for the different ensemble learning methods, much more so than other weak learner. In fact, decision trees are non-linear, which assures better performance than the linear classifiers for bagging and Boosting algorithms. Moreover, despite their high variance, averaging the result of many decision trees reduces the variance component while maintaining a low bias. Furthermore, decision trees are reasonably fast to train and to classify, which is a crucial criteria for D-AIM. Last but not least, decision trees are configurable in a way to meet the requirement of each ensemble learning technique. In fact, we can reduce the depth of the trees hence use small/short ones with boosting and increase it to have very detailed trees with bagging.

For the reasons mentioned above, we will deploy the totality of our ensemble algorithms using the Decision tree classifier.

2.3.4 Parallel Learning, Sequential Learning

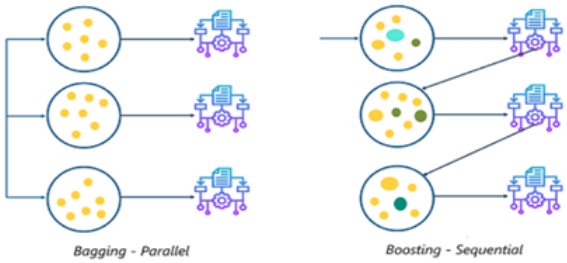
- Parallel learning is where the basic or weak learners are generated in parallel in order to guarantee the independence between the basic classifiers and significantly reduce the variance. Parallel learning is used by the Bagging and Stacking algorithm, its operation is ensured thanks to the Bootstrap sampling technique.

  Bootstrap sampling is the process of drawing with replacement n observation from a dataset of size N in order to form representative and independent samples of the initial data distribution [14]. A simplified example is illustrated in the figure 2.6.

Figure 2.6: Illustration of Bootstrap Sampling [14]

- Sequential learning is mainly used by the Boosting algorithms. It is the process of learning from the previously grown model or algorithm by assigning more weight to misclassified observations, as shown in the following figure (figure 2.7).

Figure 2.7: Parallel v.s Sequential Learning [13]



2.3.5 Bagging & Random Forest

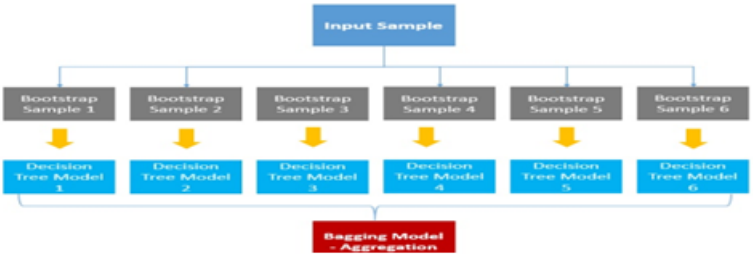
Bagging

Bagging is a very powerful ensemble method that combines, in a parallel way, the predictions from multiple homogeneous machine learning algorithms, also referred to as weak learners, in order to make more accurate predictions (figure 2.8). Bagging's aim is to reduce the variance for algorithms, such as decision trees, using the Bootstrap Aggregation as follows [14] :

First we begin by selecting the number of iterations B and then we generate the bootstraps samples of size n from an initial dataset. Afterwards, the algorithm adjusts a weak learner on each bootstrap sample and we obtain the model: f(x).

Overall we get B models: $f_1 \ldots \ f_i, .. f_b$ and the final estimator will assign to each observation, the class that received the majority of votes.

Figure 2.8: Bagging Learning Process [14]



Although bagging learning help reduce the variance, it's undeniable that the weak learners could overlap strongly since they are calculated on the basis of a draw with discount. They are strongly correlated.

Random Forest

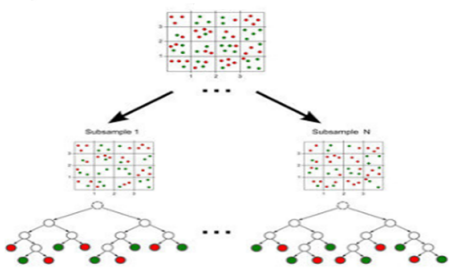
Random Forest algorithms are a refinement of Bagging algorithms that solve the correlation problem.

The Random Forest classifier searches over all the original features to find the best ones for each tree, or weak learner, and as a result these trees will be built upon a different set of features and observations. Hence they make independent, more accurate predictions.

The operation of Random Forest algorithms is similar to that of bagging regarding the selecting the number of trees and generating the Bootstrap samples. The only particularity is that for each tree a random sample of m features is drown and only those are considered for splitting. Typically m = $\sqrt{M}$, where M is the initial number of features [15].
At the end of the process, the algorithm aggregates the votes from different decision trees to decide the final class.

Let's consider the simplified example illustrated by the figure bellow (figure 2.9):

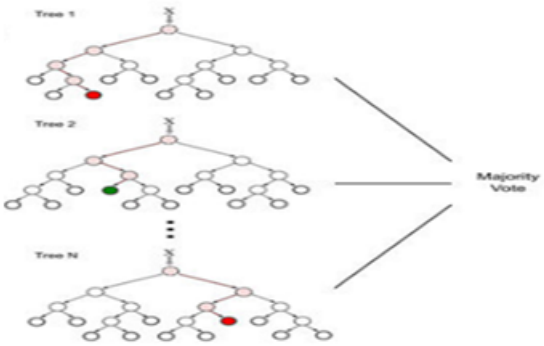
Figure 2.9: Random Forest Example-Sampling[16]



As explained earlier, bootstrap samples are drawn of the original data, and contains both labels; red and green so that each Decision Tree is built upon one of them.

For each new data point, the procedure carried out by the random forest is as follows:

Figure 2.10: Random Forest Example-Classification [16]



- The algorithm starts at the root node of a Decision Tree and traverse down through it testing the variables values in each split nodes, according to which it selects the next branch to follow until a leaf node is reached, which assigns a class to this instance.

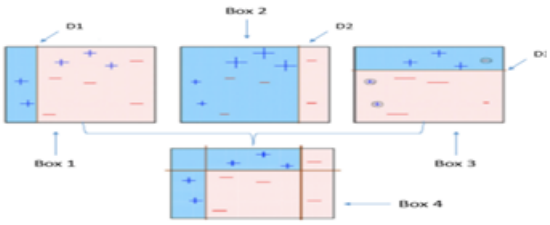
- At the end of the process (figure 2.10), each tree casts a vote for the preferred class label, and the mode of the outputs is chosen as the final prediction.

2.3.6 Boosting & Gradient Boosting

Boosting

Boosting is a very successful technique for solving binary classification problems. Contrary to the bagging method, boosting algorithms aim to reduce the bias error, by combining weak classifiers in a sequential way so that each new classifier is a fit on an improved version of the original data set, as depicted in Figure 2.11

Figure 2.11: Boosting Learning Process [20]



At first the Boosting functioning begins similar to bagging, by choosing the number of classifiers or weak learners B and generating the Bootstrap samples. Then for each classifier a model b is developed using the selected sample and the error rate of the model is calculated, given by:

$$\epsilon_l = \sum_{n=1}^{N} \omega_i 1(y_i \neq \hat{y_o})$$

If $\epsilon_l > 0.5$ or $\epsilon_l = 0$, the algorithm stops, otherwise, a weight coefficient is calculated in a way that a higher likelihood of being selected in the next dataset is assigned to the wrongly classified data points. The weighted error rate is given by:

$$\alpha_l = \ln(\frac{1-\epsilon_l}{\epsilon_l})$$

In this way, boosting sequentially builds B random datasets using the learning gained from the previously chosen instances. Finally, a weighted vote on the decisions of the classifiers is used to make the final prediction [17].

Once the general principle of Boosting is explained, we move on to introduce one of its most known variants; the gradient boosting algorithm.

Gradient Boosting

Gradient boosting is an algorithm designed upon the principle of boosting weak learners iteratively by shifting focus toward the problematic observations i.e. the wrongly predicted ones.

The gradient boosting functioning is based on minimizing the loss function. In each round of training, the weak learner is built and its predictions are compared to the correct expected outcome, in a way that the difference between them represents the error rate of the model.

This error can be used to calculate the gradient, which is basically the partial derivative of the loss function. And according to which the parameterization of the model is adjusted in a way to maximally reduce the error in the next round of training [18].
The process is summarized by the following steps:
For each iteration i, we compute the cost function j() that quantifies the error between the actual observation of the response variable y and the ones predicted by the model f_i. J(), the global loss function is given by [19]:

$$J(y, f) = \sum_{n=1}^{N} j(y_i, f(x_i)$$

In order to minimize J() we need to adjust the hyper-parameters of the model f() as follows:

$$fl(x_i)) = fl - 1(x_i) - \eta\nabla(y_i, f(x_i))$$

With η is the learning constant allowing the convergence of the process and ∇ is the partial derivative of the cost function, given by:

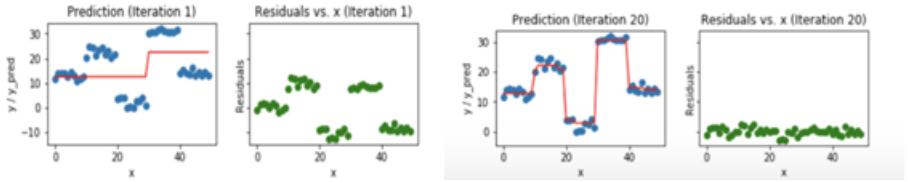
$$\nabla(y_i, f(x_i)) = \frac{\partial(y_i, f(x_i))}{\partial f(x_i)}$$

Let's consider the example illustrated in the following figure 2.12 :

Where:

- Blue dots plots represent the input (x) vs. output (y)
- Red line shows predicted values by a weak learner (Decision Tree)
- dots show residuals vs. input (x) for the iteration

Figure 2.12: Gradient Boosting Example [20]



We observe that after 20th iteration, the residuals are distributed around 0 and our predictions are very close to true values. This would be a good point to stop the training otherwise our model will start overfitting.

2.3.7 Stacking

The idea of Stacking is to combine several heterogeneous model or weak learners in a parallel way and leverage the predictions returned by them by the means of a meta-model instead of using an empirical weight function. So, we need to define two things in order to build our Stacking model: the L learners we want to fit and the meta-model that combines them [14].

The staking algorithm is summarized in the following steps:

- Choose the number and type of the weak learners and fit them to the training dataset.
- For each weak learners, make predictions for observations in the test dataset.
- Fit the meta-model, using predictions made by the weak learners as inputs.

2.3.8 Summary of Ensemble Methods

The following table summarizes the points of difference between all the ensemble learning principles.

Figure 2.13: Comparing Ensemble Learning Algorithm [21]

	Bagging	Boosting	Stacking
Partitioning of the data into subsets	Random	Giving mis-classified samples higher preference	Various
Goal to achieve	Minimize variance	Increase predictive force	Both
Methods where this is used	Random subspace	Gradient descent	Blending
Function to combine single models	(Weighted) average	Weighted majority vote	Logistic regression

As illustrated in the table above, a wide selection of ensemble methods, thus algorithms, is provided. However, as part of our project, we will apply the Bagging using the Random Forest algorithm, and Boosting, using the Gradient Boosting algorithm, only.

At this point, and based on some useful and simple examples, we walked through some widely used machine learning algorithms and assessed the importance of each one of them. Yet without the evaluation step the model development process cannot be integral. In this regard the following section will detail the basic metrics used to evaluate our models.

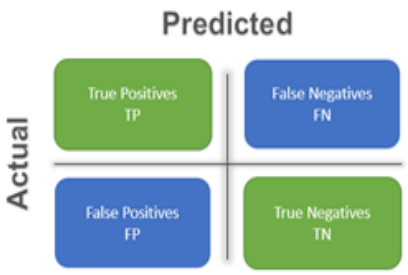
2.4 Evaluating Models

Evaluating an already trained machine learning model allows us to estimate its performance and thus compare it to other trained models. Many metrics are conceived to establish this evaluation. In this section, we will present the different indicators and metrics to evaluate machine learning models as well as the use cases of each one of them.

2.4.1 Confusion matrix

The confusion matrix is used to get an overview of the model's performance. As Figure 2.14 shows, it is a table representing the four types of outcomes that could occur while performing classification predictions [9]:

Figure 2.14: Confusion Matrix [22]



- True positives TP: Positive prediction Label was positive.
- False positives FP: Positive prediction Label was negative.
- True negatives TN: Negative prediction Label was negative.
- False negatives FN: Negative prediction Label was positive.

How can we use the information in the confusion matrix to measure the performance of models? We can compute the following indexes

Sensitivity : Also called "Recall", it informs how good the model is in avoiding the false negatives.

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity : it informs how good the model is in avoiding the false positives.

$$Specificity = \frac{TN}{TN + FP}$$

Accuracy : Accuracy determines is the proportion of the total number of correct predictions out of the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

However, accuracy can misevaluate the quality of the model in case of a high class imbalance, which is known as accuracy paradox.
For instance, if our model correctly predicts the majority of observation of the "larger" class, our accuracy index will be high, yet the model's performance won't be at best since the observations of the "smaller" class are wrongly estimated.

Precision : Percentage of positive instances out of the total predicted positive instances.

$$Precision = \frac{TP}{TP + FP}$$

A machine learning model is considered valid if it is both accurate and precise.

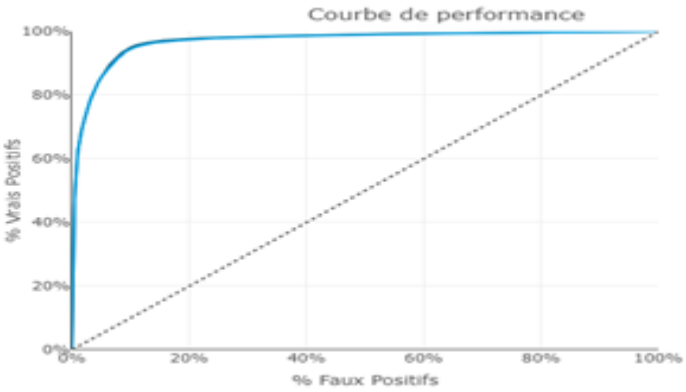
2.4.2 ROC Curve

The ROC curve (receiver operating characteristic) is a performance measure for a binary classifier. It is a graphic representation of the relationship between specificity and Sensitivity, i.e. a measure of the rate of true positives as a function of the rate of false positives (among the classified evils how many classified goods).

The inverse of the specificity (1-Sp) is on the x-axis while the sensitivity is on the y-axis of the graph bellow.

The ROC curve allows the visualization of the model's prediction error. It represents, for different thresholds, the rate of true positives as a function of the rate of false positives: TPR=f(FPR). The indicator is widely used for comparing different models [22].
The figure 2.15 is a general representation of a ROC curve.

Figure 2.15: ROC Curve



2.4.3 AUC

AUC (Area under Curve) is an index that summarizes the overall quality of the ROC curve.

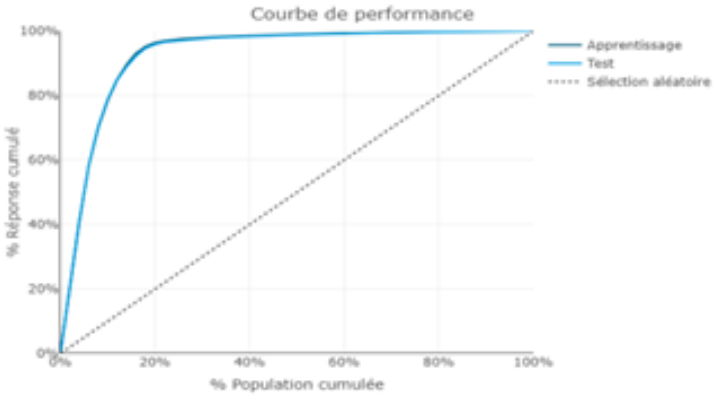
The model is considered the most performing if the AUC is close to 1 which means the separability measure is at best, whereas it is the least performing when the AUC is closer to 0 (which means it classifies 0s as 1s and 1s as 0s) [22].
An AUC equals to 0.5 means that the model is as good as random.

2.4.4 Concentration Curve of Lift

Lift is a measurement of the effectiveness of a classification model. In the same context, Lift curves, or Concentration curves, are visual aids evaluating the performance of classification models. However, in contrast to the confusion matrix that evaluates models on the whole population, Concentration or lift curves evaluate the model's performance in a portion of the population.

As we can see in the figure 2.16, on the X axis we have the proportion of our population that corresponds to a certain Lift, plotted on the Y axis. The Lift is viewed as the predictions that a random algorithm would be making and presents the share of respondents in the selection.

Figure 2.16: Concentration or Lift Curve



To measure the model's Lift, several methods are provided, however the Gini index is the most common because it has the benefit of boiling lift down to a single number. It is calculated as twice the area between the Lift Curve and the Line of Random selection.

2.5 Regulating Machine Learning Models - Cross validation

Cross validation is one of the regulating techniques of ML algorithms. It is essentially used to validate the algorithm stability as well as its generalizability on different subsets of the input data. All in the aim of reducing the overfitting and tuning the model's hyper-parameters.

Among the several forms of cross validation, the most applied is K-Fold Cross Validation in which a given data set is split into a K number of sections/folds where each fold is used as a testing set at a certain point.

Figure 2.17: 5-Fold Cross Validation [23]

Let's take the scenario of 5-Fold cross validation (K=5). As illustrated by the figure 2.17 above, the dataset is split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model, and so on. This process is repeated until each fold of the 5 folds has been used as testing set [23].
Which is a very useful technique for assessing the effectiveness of the model, since by applying, at each fold, the technical evaluation indicators, so we can ensure that the model gets most of the patterns from the data without excessive noise absorption. Moreover, we can guarantee that the performance is stable, i.e. not due to a particularly advantageous split.

2.6 Conclusion

This chapter was dedicated to explicit, by the means of simple examples, different techniques and algorithms of machine learning, precisely, the supervised learning. Nonetheless, we brought our focus essentially to a specific set of algorithms that were chosen thanks to their superior performance and popularity in the field of Marketing.

The next chapter will focus on the mathematical and probabilistic aspects allowing us to further exploit and manipulate the outputs of these models in concrete business cases.

Chapter 3

THEORETICAL FRAMEWORK PART2

Probability:The Mathematical Foundation of Machine Learning

This chapter intends to explain the essentials of probability theory, its basic concepts, as well as some advanced notions necessary to the deployment of machine learning algorithms which helps with the scope of this project.

3.1 The Probability theory

Probability theory is known as the science of uncertainty. It is a branch of mathematics concerned with the analysis of random phenomena and the prediction of their future likelihood.

Hence, whenever there is any doubt of an event occurring, the concept of probability is involved to estimate its likelihood. For instance if we want to predict the future preferences of a customer or whether he will respond favorably to a given offer we're ought to involve the mathematics of probability.

Probabilities can be expressed as percent (50%), in fractions (5/10) or in decimal form (0.5).

3.2 Probability basic concepts

In order to better understand probability we need to define some of its basic and fundamental concepts [24].

3.2.1 Experiment, Outcome, Event

Experiment

A trial or Experiment is the uncertain situations which could have multiple outcomes. For instance, whether it would rain on a daily basis is an experiment.

Outcome

An outcome is the result of a single trial. So, if it rains today, the outcome of today's trial from the experiment is "It rained".

Event

An event is one or more outcomes from an experiment. "It rained", "It did not rain" are the possible events for this experiment but only one could occur.

3.2.2 Probability space

Probability space is one of the foundations of probability theory. It is a mathematical concept used to model experiments and is formed of 3 components namely the sample space, the events and the probability measure.

Sample space

The sample space is a set of unique collections regrouping all the possible outcomes of the trial. It can contain a finite or an infinite set of values depending on the observed phenomena.

For experiments like throwing a dice or tossing a coin, the sample space contains finite possible outcomes equals respectively to 1, 2, 3, 4, 5, 6 and head, tail. Whereas infinite sample sets are the result of experiments with infinite or uncountable possible outcomes that can't be written down in a list such as the population of a city, a real number between [1, 2] etc.

Event set

The event set is a collection of all the possible subsets of the sample space, it contains a combination of outcomes and each element is called an event.

As an instance, the subset events of throwing a dice are {1}, {2}, {3}, {4}, {5}, {6}, {1, 2}, {1, 2, 3} and so on.

Probability measure

Probability measure is a function that assigns to different events the probability of their occurring.

In other words, the probability measure is a function that takes an event, i.e. an element from the sample space and maps it to a non-negative real number taking a value from 0 to1, in a way that the more likely it is for the event to occur the higher is the probability, and the sum of all these numbers equals to 1. Hence, an event with probability equals to 1 is guaranteed to occur whereas an event with probability equals to 0 will never occur.

3.2.3 Random variable

The world is full of random variables. For instance, the population of the world is dependent on time, throwing a dice, flipping a coin, the interest rates, exchange rates, price of gold, etc.

To this end, in order to calculate the likelihood of occurrence of an event, we need to put a framework to express the outcome in a measurable and observable way. We often use random variables in order to map a sample space or the possible outcomes to a set of numbers.

For instance, if we define A as the outcome of a coin toss, the originally possible outcomes would be heads, tails but we can map them as 0, 1.

A random variable can be discrete or continuous.

Continuous Random Variable

A continuous random variable has an infinite set of possible outcomes that cannot be counted. For instance, the population of the world, the interest rates, price of gold, rainfall in millimeters, etc.

Discrete Random Variable

A discrete random variable is one that has a finite set of possible outcomes and if we pick any two consecutive outcomes, we cannot get an outcome that's in between. As an instance throwing a dice, flipping a coin, days of a week, gender, months of a year, all these are examples of discrete random variable.

3.2.4 Probability distribution

Probability distribution is a notion assigned to random variables. It is a powerful tool to estimate the movements of the variable or to help understand the behavior of another random variable having the same distribution.

A probability distribution is a statistical function that describes all the possible values and likelihoods that a random variable can take within a given range. This range will be bounded between the minimum and maximum possible values.

In simpler words, a probability distribution is a list of all of the possible outcomes of a random variable along with their corresponding probability values. To give a concrete example, here is the probability distribution of a fair 6-sided dice d illustrated in the figure 3.1.

Probability distributions are used mainly for reasons such as:

- Creating the cumulative distribution functions which is a function that adds up the prob-

Figure 3.1: Probability Distribution of a Six-sided Dice [26]

Outcome of die roll	1	2	3	4	5	6
Probability	1/6	1/6	1/6	1/6	1/6	1/6

ability of occurrences cumulatively and will always start at 0% and end at 100%.

- Depicting the expected outcomes of possible values for a given data generating process.
- Defining the mean and the standard deviation.
- Anticipating the future returns on assets such as client preferences

Each probability distribution has its own shape, behavior and properties and it could be either discrete or continuous. Among the most common probability distributions we can point out to: Uniform Distribution, Normal Distribution, Binomial Distribution, etc.

The different continuous distributions as well as their expected values and variances are summarized in the table below:

Figure 3.2: Continuous Distributions Table [25]

Distribution name	Distribution symbol	Probability density function (pdf)	Mean	Variance
		$f_X(x)$	$\mu = E(X)$	$\sigma^2 = Var(X)$
Normal / gaussian	$X \sim N(\mu,\sigma^2)$	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	μ	σ^2
Uniform	$X \sim U(a,b)$	$\begin{cases} \frac{1}{b-a} & ,a \le x \le b \\ 0 & ,otherwise \end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Exponential	$X \sim exp(\lambda)$	$\begin{cases} \lambda e^{-\lambda x} & x \ge 0 \\ 0 & x < 0 \end{cases}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$

The different discrete distributions as well as their expectations and variances are summarized in the table below:

Figure 3.3: Discrete Distributions Table [25]

Distribution name	Distribution symbol	Probability mass function (pmf)	Mean	Variance
		$f_X(k) = P(X=k)$ $k = 0,1,2,...$	$E(x)$	$Var(x)$
Binomial	$X \sim Bin(n,p)$	$\binom{n}{k} p^k (1-p)^{n-k}$	np	$np(1-p)$
Poisson	$X \sim Poisson(\lambda)$	$\frac{\lambda^k e^{-\lambda}}{k!}$ $\lambda \geq 0$	λ	λ
Uniform	$X \sim U(a,b)$	$\begin{cases} \frac{1}{b-a+1} & ,a \leq k \leq b \\ 0 & ,otherwise \end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a+1)^2-1}{12}$
Geometric	$X \sim Geom(p)$	$p(1-p)^k$	$\frac{1-p}{p}$	$\frac{1-p}{p^2}$

3.2.5 Distribution function & Density function

Distribution function

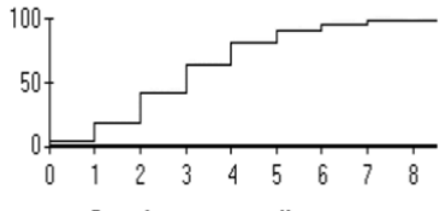
The distribution function represents the accumulation of individual probabilities. It addresses problems such the probability for that the random variable X taking a value less than x and this function depends on the type of the random variable [25] :

A function that represents a discrete probability distribution is called a probability mass function, it's given by:

$$F(x) = P(X < x) = \sum_{i=1}^{x-1} P(X = i)$$

By plotting its graphical representation we have the following curve (figure 3.4).

Figure 3.4: Staircase Curve [25]

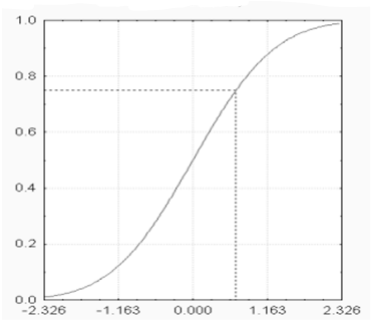


Whereas,a function that represents a continuous probability distribution is called a probability density function, it is given by:

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

And the graphical representation is illustrated by the figure 3.5:

Figure 3.5: Cumulative Curve [25]



Density function

The probability density function is a function that describes the relative likelihood for a random variable to take on a given value. It differs according to the distribution of the random variable, as well as its type.

When we use a probability function to describe a continuous probability distribution we call it a probability density function (commonly abbreviated as pdf), it is given by: $f(x) = \partial F(x)/\partial x$. Or, in the case of discrete variables equivalent to P (X = x) Where F(x) is the distribution.

3.2.6 Central tendency characteristics

The quantiles

We call quantile or fractile of order α ($0 \leq \alpha \leq 1$) of a random variable X whose function distribution is F (x), the value x such that F (x) = , x is called quantile of order α.
In the case where X is a discrete variable, F (x) = α means P (X $\leq \alpha$) = α.
Here we list some specific quantiles [27].

The median

The median is the quantile of order α =1/2, in other words the median divides the population into two equal parts; it is a characteristic of central tendency [27].

Quartiles

The quartiles, denoted Qi (respectively i = 1, 2, 3) correspond to the order quantiles (α = 0.25, 0.5, 0.75). Note that Q2 = Median [27].

Deciles

The k-th decile (k = 1,.., 9) is the quantile of order k / 10. In particular, the 5th decile corresponds to the median.

The mode

We call mode (dominant value, most probable value) of a random variable, the value M for

which the frequency histogram presents its maximum [27].

When the random variable X is continuous, with a density function provided with a derivative first and a second derivative, the Mo mode satisfies f '(Mo) = 0 and f "(Mo) ¡0) (concavity down).

In the case of discrete variables, Mo is the value of X associated with the greatest probability, hence the most likely value designation.

Interquartile Range (IQR)

The first quartile (Q1) is the 25th percentile of a data set; the second quartile (Q2) is the 50th percentile (median); and the third quartile (Q3) is the 75th percentile.

The IQR measures the difference between 75th and 25th observation using the formula: $IQR = Q3Q1$ [27].

Mean, Expected Value

In probability and statistics, the expectation or expected value, is the weighted average value of a random variable [27]. It is given by:

Expectation of discrete random variable is:

$$E(x) = \sum_{i=1}^{\infty} xiP(x)$$

Expectation of continuous random variable is:

$$E(x) = \int_{-\infty}^{\infty} xP(x)dx$$

Standard deviation, Variance

Standard deviation measures the deviation of the random variable from its mean or expectation [27]. The variance is the square of standard deviation it is denoted as σ^2 and given by:

$$\sigma^2 = \frac{\sum(x-\mu)^2}{N}$$

Covariance

The covariance gives some information about how two random variables X and Y are statistically related [27]. It is given by:

$$Cov(X,Y) = E[(X-EX)(Y-EY)] = E[XY] - (EX)(EY)$$

Correlation coefficient

The correlation coefficient is defined by [27]:

$$\rho(XY) = \rho(X,Y) = cov(X,Y)$$

3.3 Advanced probability concepts: Event Types

The probability calculation is directly linked to the type of event being reconciled. Therefore, it is essential to know the different types of existing events [29].

Mutually Exclusive Events

Two events are mutually exclusive when they cannot occur at the same time. For instance the result of tossing a coin, getting Heads and Tails at once are mutually exclusive or disjoint events.

Independent Events

Two events are independent when the occurrence of one does not impact or alter the occurrence of the other. For example, if we roll a dice twice, getting a five on the first roll does not affect the probability of getting a two on the second roll. Therefore, we can say that these two events are independent.

Dependent Events

Two events A and B are Dependent if the occurring of event A changes or causes the occurring probability of the event B, we can say that the probability of event B is dependent on event A or vice versa. For example, the rain amount and the agriculture situation of a given country.

3.4 Advanced probability concepts: probability types

It is useful to know all the different types of probabilities in order to better model the situations and have accurate results [28]. Among these types we mention:

Empirical probability

The empirical probability, also known as experimental probability, illustrates the likelihood of an event occurring based on historical data. The empirical probability of an event is given by:

$$Empiricalprobability = \frac{NumberofTimesOccurred}{TotalNo.ofTimesExperimentPerformed}$$

The main advantage of using empirical probability is that the probability is backed by experimental studies and data away from assumptions and hypotheses.

Theoretical probability

Theoretical probability on the other hand is an expected probability based upon

knowledge and intuition of the experiment. It is given by:

$$Theoreticalprobability = \frac{Numberoffavourableoutcomes}{Numberofpossibleoutcomes}$$

Theoretical and empiric probabilities do not always meet. But according to The Law of Large Numbers, Empirical (experimental) probability approaches theoretical probability when the number of trials is extremely large.

Marginal Probability

The marginal probability of an event A is the probability of its occurrence, and denoted P(A). Assuming that we have 20 cars in a parking-lot among which 5 are red, an example of a marginal probability would be the probability that a car from that parking-lot is red is equals to P(red) = 0,25.

Joint Probability

Joint probability is a statistical measure that illustrates the likelihood of two events occurring together and at the same point in time. It is the outcome of the multiplication of two events, denoted by P (A ∩ B) where A and B are the two events to intersect, and the outcome depends of the type of events.

Conditional Probability

The conditional probability is the probability of occurrence of an event B given the knowledge that an event A has already occurred. If the probability of events A and B are P(A) and P(B) respectively then the conditional probability of B such that A has already occurred is P(A|B).

The concept of independent and dependent events comes into play when we are working with conditional probability since the type of the events under consideration is the key to denote the conditional probability formula. As a matter of fact, if the occurrence of the prior event A does cause of effect the occurrence of the posterior event B, i.e. A and B are two dependent events, then the conditional probability is given by:

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

Whereas, if the occurrence of event A doesn't impact the occurrence of event B then the conditional probability verifies:

$$P(A \cap B) = P(A)$$

$$P(B \cap A) = P(B)$$

Other types of probability

- Subjective probability is based on beliefs. For example, one might "feel" a lucky streak coming on.

- Axiomatic Probability: a type of probability that has a set of axioms (rules) attached to it.

3.5 Advanced probability concepts: Counting Techniques

In analogy with the previous parts, defining the probability counting rules seems to be paramount at this stage [29].

The Multiplication Rule/ Joint Probabilities

Given two events A and B, the multiplication rule indicates the probability of accruing of both A or B at the same point in time. This notion is highly linked to joint probability and is given by P (A ∩B). The outcome depends on the type of the events to intersect.

- For Mutually Exclusive Events the join probability is given by: P (A ∩B) = 0.
- For Independent Events the join probability is given by: P (A ∩B) = P (A) × P (B)
- For Dependent Events the join probability is given by: P (A ∩B) = P (A) × P (B | A)

Where P (B|A) is the probability of B given A has happened i.e. conditional probability.

The Addition Rule

Given two events A and B, the addition rule indicates the probability of occurence of either A or B at a certain point in time. The outcome depends on the type of the involved events.

- For mutually exclusive events the joint probability is equal to 0, so the additional rule given by: P(A or B) = P(A) + P(B)
- For independent and dependent events the joint probability P(A ∩B) is greater than 0, so the addition rule is given by: P(A or B) = P(A) + P(B) - P(A∩B)

Recapitulation

The following table recapitulates and summarizes the methods of combining probability (figure 3.6)

Figure 3.6: Summary Combining Probabilities

Table 6.11 Summary of Combining Probabilities

***And* probability: independent events**	***And* probability: dependent events**	***Either/or* probability: non-overlapping events**	***Either/or* probability: overlapping events**
P(A and *B)* = *P(A)* × *P(B)*	*P(A* and *B)* = *P(A)* × *P(B* given *A)*	*P(A* or *B)* = *P(A)* + *P(B)*	*P(A* or *B)* = *P(A)* + *P(B)* − *P(A* and *B)*

3.6 Advanced probability concepts: Standardization

A common statistical way of comparing probabilities is standardization, since it redistributes the data on a single scale so a comparison can take place. So what is Standardization and when do we use it?

Standardization

Standardization is a very useful process because it enables us to compare two scores of different kinds and measured on different scales by converting their different distributions (in other words, standardizing) into a standard normal distribution SND using the z-score formula.
A standard normal distribution (SND) is a normally shaped distribution with a mean of 0 and a standard deviation (SD) of 1.

What is a Z-Score?

Simply put, a z-score, also called a standard score gives us an idea of how far from the mean a data point is. But more technically it's a measure of how many standard deviations below or above the population mean a raw score is. The z-score is positive if the value lies above the mean and negative if it lies below the mean [27].

Z-Score Formulas

To standardize a variable, we need to convert an observed value for a random variable into a z-value as follows [27]:

$$z = \frac{observedvalue - populationmean}{standarddeviations} = \frac{x - \mu}{\sigma}$$

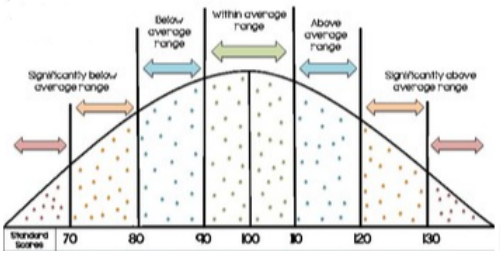
How do we interpret z-score?

Z-scores help reciprocate 4 questions [30]:

- The value of the z-score tells us how many standard deviations an observation is away from the mean. If a z-score is equal to 0, it is on the mean. A positive z-score indicates the raw score is higher than the mean average. For example, if a z-score is equal to +1, it is 1 standard deviation above the mean. A negative z-score reveals the raw score is below the mean average. For example, if a z-score is equal to -2, it is 2 standard deviations below the mean.
- The percent chance of an event happening beyond a certain point: This is the number under the curve beyond the z value.
- The percent chance of an event happening under a certain point: This is the number under the curve up to the z value.

- The percent chance of an event happening between two points: This could be the number under the curve bounded by two points.

Figure 3.7: Z-score Interpretation [30]



The figure 3.7 highlights the possibility of answering questions such as:

- How appetent a single client is in comparison to all customers?

Supposing this client has a z-score of 067. By means of the standard normal distribution table we can identify the probability of scores above or below our z-score score.
For our example, the probability that a score is greater than 0.67 is 0.2514, i.e. 25.14%. Consequently, we can conclude that the client in consideration behaves better than 75% of the client base [31].

- How to identify the top 10% best clients?

After converting the frequency distribution into a standard normal distribution, and since the percentage we are trying to find, the top 10% of clients, corresponds to 0.9. The procedure should be as follow: we first need to find the z-score correspondent to the value 0.9 in standard normal distribution table.
Afterwards, and since we already have the key information (the mean score, the standard deviation, and z-score, z), we can deduce the score allowing the clients to be classified among the top 10%.

3.7 Application of probabilities in Machine Learning

Probability theory provides a useful framework in which to model machine learning algorithms and make precise statements about their effectiveness. In fact, it is undeniably a pillar of the field of machine learning. In this part, we will introduce the importance of probabilities machine learning [32].

3.7.1 Class Membership Prediction

Predictive classification models can be framed as a probabilistic “class-membership” model, where the probability of a given observation belonging to each known class is

to be predicted. The model functioning will be assured by generation for each observation a probability of membership to each known class. Then, these probabilities can be transformed into the corresponding class label by choosing the class with the highest probability.

3.7.2 Designing Models

Some algorithms are specifically designed to harness tools and methods from probability theory. As an instance we can mention Naive Bayes algorithm, which is constructed using Bayes Theorem with some simplifying assumptions.

We can also refer to the linear regression algorithm since it's a probabilistic model that minimizes the mean squared error of predictions, as well as the logistic regression algorithm that can be seen as a probabilistic model that minimizes the negative log likelihood of predicting the positive class label.

3.7.3 Models Are Evaluated With Probabilistic Measures

For algorithms where a prediction of probabilities is made, evaluation measures are required to summarize the performance of the model.

As an instance let's consider binary classification models where a single probability score is predicted, many measures used to summarize the performance of these models are based on predicted probabilities. These evaluation measures have already been explained in the previous chapter (the one dedicated for machine learning) so for now, note that the most common examples are: ROC AUC confusion matrix Precision Recall.

3.8 Conclusion

While the previous chapter focused on the operation of machine learning methods, this one established, on one hand, some basic and advanced probability concepts useful for the deployment of these methods, on the other hand, the mathematical theories of combining and manipulating the probabilities generated by these models.

All these concepts will be used in further chapters, in order to study in detail the construction of the different scores for a specific client, as well as manipulating the generated probabilities for client targeting purposes.

Chapter 4

PRACTICAL FRAMEWORK

Application and Manipulation of Predictive Scoring

In this chapter, we will start the processing of our goal. To this end, we'll go through the following sections: at first we will deeply understand the business problem at hand as well as the data at our disposal. Afterwards, we will present the intuitive and basic approach, discuss its limitations and finally propose an alternative. Later in this chapter we will go through the different steps of conception and application of the predictive scores, compare their performance from the practical and business perspectives. And finally establish a comparison between the basic and the proposed approach.

As for the second part of this project, we will highlight the need of Machine Learning probabilities in the targeting strategy of D-AIM.

Note that the work was implemented under s R 3.5.1 using different libraries such as "data.table" and "lubridate" for easily manipulating the tables, as well as python by means of the "scikitlearn" library for modeling scores, tuning the hyper parameters as well as post processing of data.

4.1 Application of Predictive Scoring

4.1.1 Business Introduction of The Project

As explained in the introductory chapter, the first part of this project is developed in co-operation with a telecommunication operator who wants to enhance the knowledge of its data consumer customers in order to push or up-sell their usage threshold.

Actually, accomplishing this step requires combining scores of different events, namely the Non-Fragility to Data in one hand, and in the other the Appetency to any of the four Data Bundles: Bundl3, Bundle5, Bundle8 and Bundle12.

The Bundles at our disposal are Data destined bundles, called after the purchase price. Bundle3, for instance, is 3 unit of currency (UoC), and so on.

After detecting the customers who are Non-Fragile Data and at the same time Appetent to one of the previously described Bundles, we will proceed by targeting them with the Bundle with the higher price grid.

In this first section, we will present the intuitive and basic approach of combining scores, which consist on deploying simple scores each one predicting a single event, and then combining the outputs of these scores to detect the customers who respond to both events at once. Then we will present the alternative proposed approach which consists of deploying Machine Learning algorithms predicting the occurrence of both events at once, called combined scores.

Moreover, in order to distinguish between the simple and combined scores, we created a specific nomenclature for each score. For instance, the simple score predicting the appetency to the Bundle3 will be referred to as Simple_Bundle3, while the combined score that predicts the Appetency to the Bundle3 and the Non-Fragility to Data will be called Combined_Bundle3.

Later in this section we will go through the different steps of the conception and application of the predictive scores, compare their performance from the practical and business perspectives. And finally establish a comparison between the basic and the proposed approach.

4.1.2 Machine Learning Scoring Process

To model any score, it is preferable to have general and credible information about each individual, which means an integrated and complete Explanatory Feature Datamart. The next section of this chapter will first detail the process of creating the Datamart of explanatory features, confront the issue of missing values, as well as the feature selection process for each score. Then we will go through the choice of models as well as the scoring methodology of D-AIM.

Explanatory Feature DTM

Automating its processes has been one of D-AIM's main strategies, since it helps save time as well as human and financial resources. In this context, D-AIM created a standard format for all Telecommunication Data: Global Telco Data Structure or GTDS.

After receiving raw data from the client, the conversion process begins by creating the GTDS Tables which are the result of minor modification, merges or aggregations such as sums, counts and averages of the original tables and designed to describe the customer's daily characteristics. Among the GTDS tables, we mention: Profile, Data ,Recharge, Revenue, Voice & SMS and Subscription. At an advanced level, the GTDS tables will serve for the construction of the Datamart of the explanatory features that will serve as an input for the scores we wish to constitute since they have the same depth of history and the same explanatory variables.

This Datamart contains all the explanatory variables for the event to be predicted. These variables are varied and are not limited to the income generated, the usage (Data / Voice / SMS) and the recharge existing in the GTDS tables, but include also variables and ratios necessary and significant to the event to be predicted. The idea is to have a number of solid explanatory variables in terms of quantity and quality, which allows us to optimize our scores. For the project at hand, the distribution of the resulting Datamart is as follows:

IDENTIFIER	Quantitative Variable	Categorical Variable
CONTRACT-ID	248	27

The information is identified by the user's ID, i.e. his telephone number and expresses a continuous behavior of the customers. The variables are aggregated by day, week and month, which introduce the concept of seasonality and periodicity in the explanation of events.

Data cleaning & Feature selection

As mentioned, the Explanatory Feature Datamart is an integral and complete one that contains all the possible features describing the considered client. However, the large number of features requires some elementary data processing steps, such as imputing missing values and a

primary feature selection.

Imputing Missing values is one of the main treatments that ensure a better performance of predictive scores. In fact, in our project the missing values do not reflex the absence of the information, in this regard dealing with missing values do not require advanced imputation techniques, but a simple interpretation of its origin.
For instance, the absence of a customer of a daily table simply means that he did not perform the action in question on that day.

For quantitative variables a missing variable simply amplify the absence of usage, revenue, etc., thus it is replaced by the value “zero”. The same applies to quantitative variable. For example, for the variable that illustrate the name of the second SIM owned by a customer, missing values are explained by the fact that the customer is not a multi-simmer and thus replaced by the value ”MISSING”.

At the end of this step, we will eliminate the constant and quasi-constant variables because of their weak discriminating power.

Selecting Machine Learning Algorithms

Several machine learning algorithms are made available for data scientists. However, the nature of the problem to be solved can help us narrow down the search for the appropriate one to deploy. In the case of marketing campaigns, especially while manipulating real world data, reducing the error composed of bias and variance is one of the main factors of choice.

As mentioned in prior chapters, ensemble learning algorithms are established principally to reduce the components of the prediction error, i.e. the bias and the variance (section 2.3.2).
In this regard, our application will be limited to Random Forest and Gradient Boosting models.

At the end of the application phase, an evaluation of the performances of these two algorithms will be established.

dataset store telecom
kaggle

Configuration Machine Learning Algorithms

In addition to setting the feature space, specifying the perimeter and the response variable and choosing the adequate Machine Learning algorithm, hyper-parameter adjustment is the next crucial task for finding the model with the highest predictive ability.
The configuration or adjustment of machine learning models is a type of optimization problem. We have a set of hyper-parameters, and our goal is to find the correct combination of their values, which can help us find the minimum (e.g. loss) or maximum (e.g. accuracy) of the function.

In this section, we will first specify for each model the important hyper parameters that we need to configure, then we will present their meaning and standard configuration and finally we’ll

introduce the process of reaching the optimal configuration.

In fact, a wide selection of hyper parameters is defined for each Machine Learning classifier. For Random Forest we mention:

Figure 4.1: Random Forest Hyper-parameters

Hyper-Parameter	Meaning	Default Value
N° of estimator	The number of trees that we add to the model.	100
Max Features	Max number of features considered for splitting a node. The supported criteria{"auto", "sqrt", "log2"}	auto max_features=sqrt(n_features)
Criterion	The function to measure the quality of a split. The supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.	Gini
Min Samples Leaf	The minimum number of samples required to be at a leaf node.	1
Max Tree Depth	The maximum depth of the tree.	3
Bootstrap	The minimum number of samples required to split an internal node.	2

As for gradient boosting we cite:

Figure 4.2: Gradient Boosting Hyper-parameters

Hyper-Parameter	Meaning	Default Value
N° of estimator	The number of trees that we add to the model.	100
Learning Rate	Learning rate narrows the contribution of each learner(tree).	0,1
Row Sampling	The fraction of samples to be used for fitting the individual base learners (tree).	1
Min Samples Leaf	The minimum number of samples required to be at a leaf node.	1
Max Tree Depth	The maximum depth of the tree.	3
Min Samples Split	The minimum number of samples required to split an internal node.	2

As shown in the table above, a variety of hyper-parameters have been established for each model, but most data scientists agree on parameters with higher model adjustment capabilities. (As shown below).

Figure 4.3: Main Hyper parameters for Gradient Boosting & Random Forest

Principaux hyper paramètres	Gradient Boosting	Random Forest
Max_depth	✔	✔
Min_weight_fraction_leaf	✔	✔
Criterion	✖	✔
N° estimators	✔	✔
Learning_rate	✔	✖

For our project, we implemented the grid search algorithm, using the scikit-learn GridSearchCV() function as follows: First, through several experiments using the RF and GB algorithms, a series of values are fixed for each hyper-parameter to test and determine the value that can ensure the best performance.

The following tables will explicit the ranges fixed for the RF and GB models:

Figure 4.4: Fixed Range of Hyper-parameters

Random Forest

Hyper parametres	Value Range
max_depth	5;3
min_weight_fraction_leaf	0.005;0.05;0.01
criterion	gini;entropy
n_estimators	50;40

Gradient Boosting

Hyper parametres	Value Range
max_depth	5
min_weight_fraction_leaf	0.005;0.05;0.01
learning_rate	0.1;0.25
n_estimators	20;40

Which brings us to test 1×3×2× 2=12 combinations for the Gradient Boosting algorithm and 2×3×2×2=24 ones for the Random Forest algorithm. Then, we use 5-fold cross-validation for each algorithm, therefore the algorithm will execute a total of 12×5=60 times for the Gradient Boosting, and 24×5=120 times ones for Random Forest.

The end of the process is achieved by retaining the configuration that gave the best accuracy for the construction of the final model. The best hyper parameters for each method are represented in the figure 4.5:

Figure 4.5: Best Hyper parameters for Each Method

Random Forest

Hyper parametres	Value Range
max_depth	5
min_weight_fraction_leaf	0.005
criterion	entropy
n_estimators	40

Gradient Boosting

Hyper parametres	Value Range
max_depth	5
min_weight_fraction_leaf	0.005
learning_rate	0.25
n_estimators	40

D-AIM Scoring methodology

Scoring is one of the main activities of D-AIM. All its strategy is built upon a high quality scoring technique. A score makes it possible to model the occurrence of a binary event (yes/ no), over a configurable scope and period. AS mentioned earlier defining the scores requires defining the following terms:

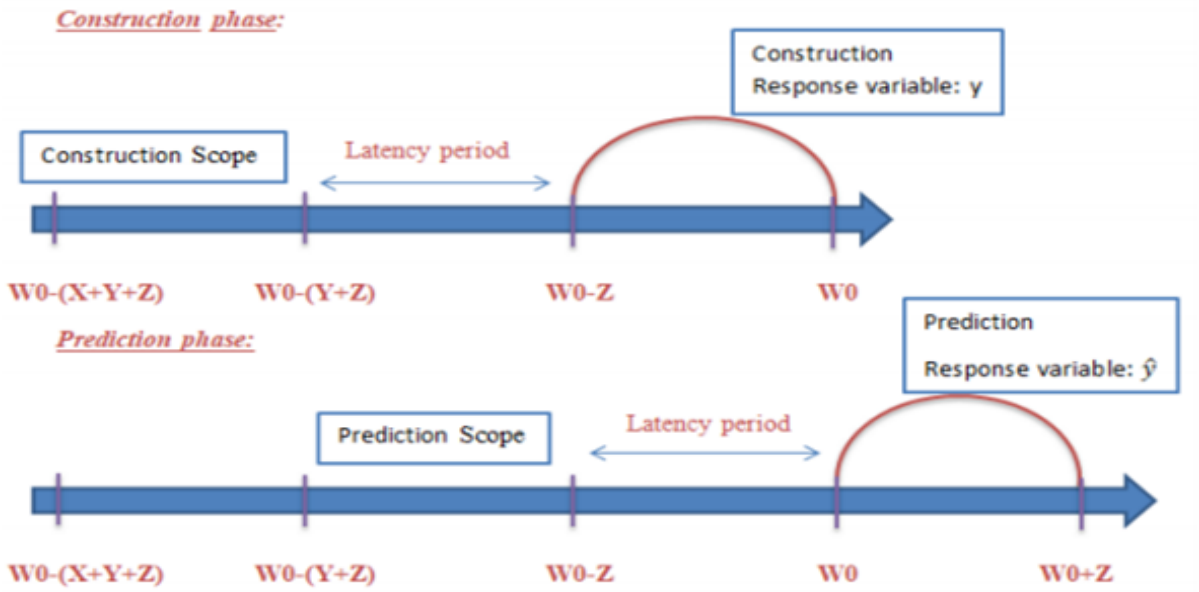
- The scope of the study: sub-population of customers to score, as well as the perimeter construction period: X.
- The Latency period: during which we will not be interested in customer behavior: Y.
- The response variable y according to the event problematic, as well as the prediction period: Z.
- The date of prediction W0.

As illustrated in the figure 4.6, D-AIM scoring technique is divided in two phases: the Construction phase and Reapplication phase, where:

Construction phase: The Machine Learning model training phase, based on historical data and contains a construction scope and construction Y.

Prediction phase: The real-time prediction phase, contains a well-defined construction scope and Y to predict.

Figure 4.6: D-AIM Scoring Steps



Application of Scoring Methodology

The scoring methodology explained earlier will be established by applying the chosen ML algorithms to deploy a total of 9 scores of two types.
To remain faithful to the business problematic defined earlier, we will deploy two different types of scores: simple scores and combined scores.
The simple score will predict the occurrence of a single event, described by the appropriate response variable and perimeter, which is the number of customers who meet the defined criteria. Moreover, for each one of the simple scores the perimeter and prediction periods are fixed by the marketing experts of D-AIM, case of the Non-Fragile Data score, or by the offer validity period, for the rest of the scores.

For sample scores only one standard is defined for the perimeter. The following table will explicit all the details relative to the simple scores:

Figure 4.7: Definition of the Simple Scores

score	Perimeter (target)	Y / Phenomenon (predictive behavior)
Non-Fragile Data	Data user in the last week	Data user in the next 4 weeks
Appetent Bund3	Non user of bundle3 in the previous 8 weeks	Subscriber to a 3kd bundle in the next 4 weeks
Appetent Bund5	Non user of bundle or user of 5/8/12/16kd bundle in the previous 8 weeks	Subscriber to a 5kd bundle inthe next 4 weeks
Appetent Bund8	Non user of bundle or user of 8/12 and 16kd bundle in the previous 8 weeks	Subscriber to a 8kd bundle inthe next 4 weeks
Appetent Bund12	Non user of bundle or user of 12/16kd bundle in the previous 8 weeks	Subscriber to a 12kd bundle inthe next 4 weeks

Whereas, for combined scores, since we're predicting the joint probabilities of two events, i.e. the likelihood that they happen simultaneously, an additional restriction must be made and the clients to score must meet two criteria at once.
The same implies for the response variable, for each one of the combined scores our response variable is characterized by two conditions, if the two conditions are ascertained, then the combined response variable will be equal to 1. Otherwise it is worth 0.

The following table will illustrate all the details relative to the combined scores:

Figure 4.8: Definition of the Combined Scores

Combination	Perimeter (target)	Y / Phenomenon (predictive behavior)
Non-Fragile Data v.s. Appetent Bund3	Non user of bundle primary in the previous 8 weeks	Subscriber to a 3kd bundle in the next 4 weeks & Non-Fragile Data
Non-Fragile Data v.s. Appetent Bund5	Non user of bundle or user of 5/8/12/16kd bundle in the previous 8 weeks	Subscriber to a 5kd bundle inthe next 4 weeks & Non-Fragile Data
Non-Fragile Data v.s. Appetent Bund8	Non user of bundle or user of 8/12 and 16kd bundle in the previous 8 weeks	Subscriber to a 8kd bundle inthe next 4 weeks & Non-Fragile Data
Non-Fragile Data v.s. Appetent Bund12	Non user of bundle or user of 12/16kd bundle in the previous 8 weeks	Subscriber to a 12kd bundle inthe next 4 weeks & Non-Fragile Data

4.1.3 Evaluation of Machine Learning Scores

As mentioned earlier, at this stage, we will deploy two types of scores for each bundle, simple score and combined with non-fragile data scores.

In addition, we will use two types of ensemble learning models of the deployments: Random Forest models and Gradient Boosting model.

In the upcoming section, we will establish an evaluation of these scores, from the technical as well as the Marketing perspectives, all in the aim of, firstly, select the appropriate model for each score. Then insure the technical performance of these scores using the evaluation metrics mentioned in prior chapters. Finally, we display the performance via some Marketing indicators in order to assess the performance of these scores as well as decide upon the optimal targeting strategy to adopt.

Most Performing Model Selection

For each of the previously described scores, namely the simple and combined ones, we trained two machine learning models, the Random Forest and Gradient Boosting, since each one of them aims to reduce a specific component of the predictive error.
This section will be interested in establishing a comparison of the performances of the two deployed models, in order to specify, for each score, the model that best corresponds to the studied data and its different patterns.

Accordingly, several indicators can be used to technically evaluate the performance of a machine learning algorithm. However, at this stage, the comparison as well as the choice of the appropriate algorithm will be based on the AUC indicator. In a way that the model with the higher AUC will be selected for the rest of the work, since it is the more capable one of distinguishing between positive class and negative one, and therefore assuring more accurate results.

The upcoming figures will illustrate the performance of the deployed models according to the AUC indicator for each score.

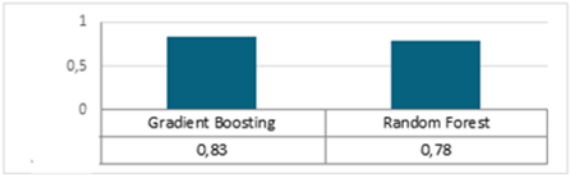


Figure 4.9: Simple_Bundle3 AUC

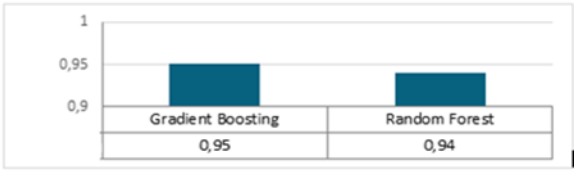


Figure 4.10: Combined_Bundle3 AUC

Figure 4.11: Comparing Both Moled's AUC for the Bundle3

Although both models established an accurate prediction of the Appetency to Bundle 3, the Gradient Boosting model is the most performing one, with an AUC of 0.83 for the Simple_Bundle3

score, and an AUC of 0.95 for the combined_Bundle3 score.

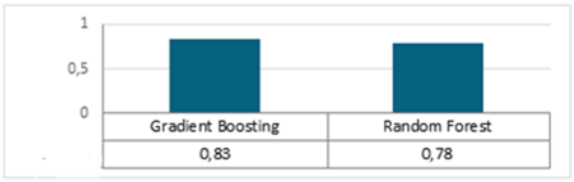


Figure 4.12: Simple_Bundle5 AUC

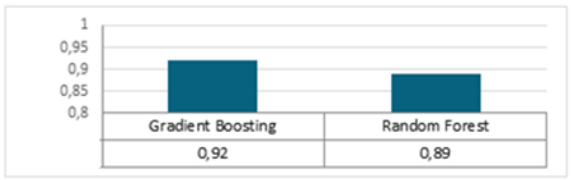


Figure 4.13: Combined_Bundle5 AUC

Figure 4.14: Comparing Both Moled's AUC for the Bundle5

The same implies for the Appetency to Bundle 5 scores, the Gradient Boosting is the algorithm that best predicts the Simple_Bundle5, with an AUC of 0.83, as well as the combined_Bundle5 score, with an AUC of 092.

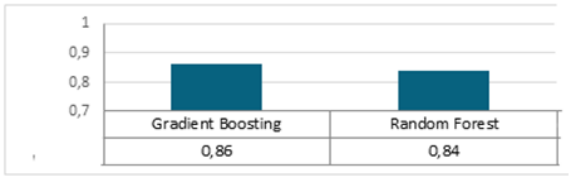


Figure 4.15: Simple_Bundle8 AUC

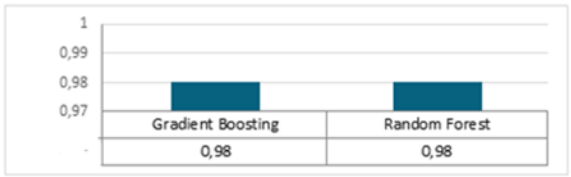


Figure 4.16: Combined_Bundle8 AUC

Figure 4.17: Comparing Both Moled's AUC for the Bundle8

As shown in the figure above, the Gradient Boosting model again, is the also the best model for the Simple_Bundle8 score with an AUC equal to 0.86 and the same for the combined_Bundle8 score with an AUC equal to 0.98.
However such a high AUC could raise the problem of over-adjustment or overfitting.

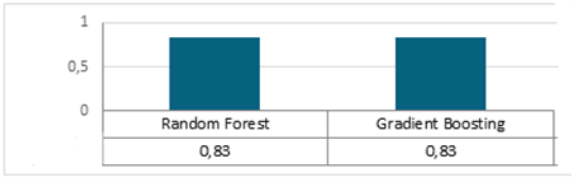


Figure 4.18: Simple_Bundle12 AUC

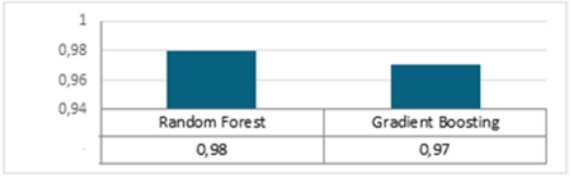


Figure 4.19: Combined_Bundle12 AUC

Figure 4.20: Comparing Both Moled's AUC for the Bundle12

However, the Random Forest algorithm is the most performing for the Appetent Bundle 12 scores, with an AUC equal to 0.83 for the Simple_Bundle12 score and to 0.98 for the Combined_Bundle12 one.
But again, we might face an overfitting issue for the Combined_Bundle12 score, advocated by an AUC equals to 98%.

Overly Gradient Boosting algorithm presented higher AUC and therefore it is the considered one for the majority of the scores, except the Simple_Bundle12 and Combined_Bundle12 scores, where the appropriate algorithm is Random Forest.

As mentioned earlier, for each score, the model with the higher performance index will be considered for use in the rest of the work.

Technical Performance Evaluation

ROC curve & AUC index

To better test the robustness of a predictive model, we can consider the ROC curve since it allows the visualization of the model's prediction error.

The idea of the ROC curve (receiver operating characteristic) is to vary the "threshold" from 1 to 0 and, for each case, calculate the false positive rate, that is the share among the total number of non-respondent (TN + FP), of individuals who do not respond to desired phenomena, yet modeled by the algorithm as respondent (FP), is displayed on the x-axis.
Whereas, on the y-axis we display the true positives rate, i.e. the share of individuals who are wrongly estimated as respondents, among the total number of respondents.
The curve is constructed by varying the threshold in order to have different couples of the specificity TPR and sensitivity FPR, and the goal is to have the most porous curve possible since it implies a false positive rate of 0 a true positive rate of 1, thus more accurate results.

Therefore, the ROC curve, same as the AUC index are considered excellent ways to compare our performance gain. The figure below shows the OCR curves for each model.

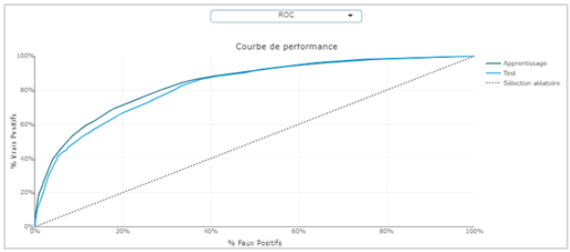


Figure 4.21: Simple_Bundle3 ROC curve

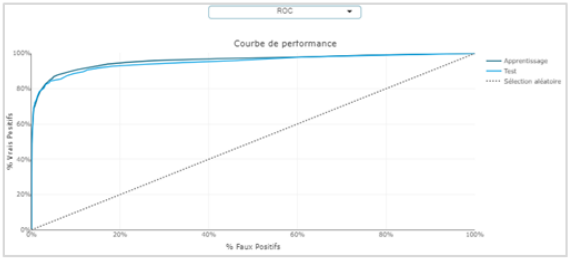


Figure 4.22: Combined_Bundle3 ROCcurve

Figure 4.23: Comparing Both Moled's ROCcurve for the Bundle3

According to the ROC curve, both models predicting Simple_Bundle3 and combined_Bundle3 scores have great performance, since for both scores the train curve is confused with the test one. Moreover, for the Simple_Bundle3 score, when the model finds 71% of respondents (rate of true positives), it is mistaken for only 20% of non-respondents by wrongly estimating them as respondents (rate of false positives). As for the combined_Bundle3 score, the performance

is more efficient, in fact, when the model finds 94% of respondents (rate of true positives), it is wrong for 17% of non-respondents by wrongly estimating them as respondents (rate of false positives).

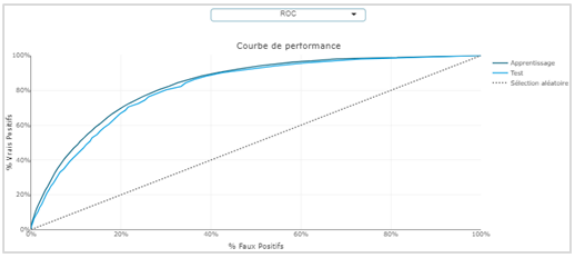


Figure 4.24: Simple_Bundle5 ROC curve

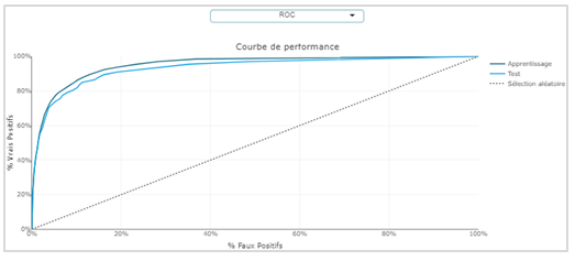


Figure 4.25: Combined_Bundle5 ROCcurve

Figure 4.26: Comparing Both Moled's ROCcurve for the Bundle5

Again, the ROC curves assure the great performance of the model predicting the Simple_Bundle5 score. As a matter of fact, when the model finds 71% of respondents (rate of true positives), it is mistaken for 20% of non-respondents by wrongly estimating them as respondents (rate of false positives). Whereas for the combined_Bundle5 score, when the model finds 94% of respondents (rate of true positives), it is mistaken for 20% of non-respondents by wrongly estimating them as respondents (rate of false positives).

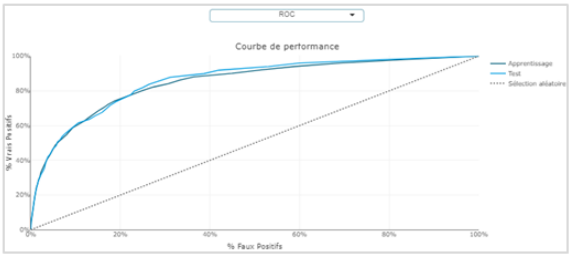


Figure 4.27: Simple_Bundle8 ROC curve

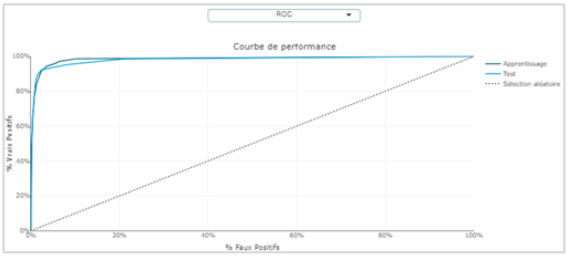


Figure 4.28: Combined_Bundle8 ROCcurve

Figure 4.29: Comparing Both Moled's ROCcurve for the Bundle8

Here, for the Simple_Bundle8 scores, when the model retrieves 78% of respondents (rate of true positives), it is wrong for 21% of non-respondents by wrongly estimating them as respondents (rate of false positives).
While for the combined_Bundle8 scores, when the model recognizes 80% of respondents (rate of true positives), it is wrong for 10% of non-respondents by wrongly estimating them as respondents (rate of false positives). Which confirms the high performance of two models.

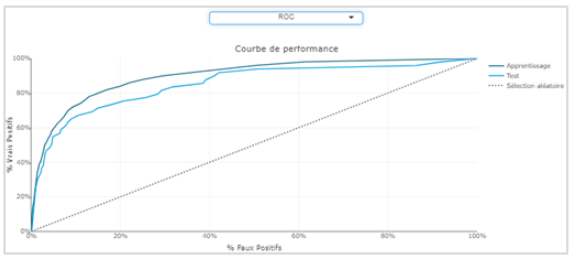


Figure 4.30: Simple_Bundle12 ROC curve

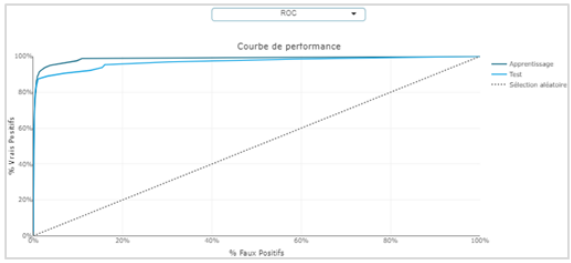


Figure 4.31: Combined_Bundle12 ROCcurve

Figure 4.32: Comparing Both Moled's ROCcurve for the Bundle12

The same implies for the Simple_Bundle12 score, since the figure above illustrates that when the model locates 84% of respondents (rate of true positives), it is mistaken for 20% of non-respondents by wrongly estimating them as respondents (rate of false positives).
While for the combined_Bundle8 score, when the model finds 80% of respondents (rate of true positives), it is wrong for 11% of non-respondents by wrongly estimating them as respondents (rate of false positives).

Overall, the ROC curves confirm our hypothesis concerning the performance of the models. Similarly we confirm the fact that the combined scores lead to better performance

LIFT or CONCENTRATION Curve

The lift or concentration curve is a technical evaluation indicator with marketing finalities. In fact, this curve can visually show the share of respondents found when selecting the best individual based on the model.
On the x-axis, we display the share of the selected population, and on the y-axis, we expose the share of the respondents in the selection. The goal is to keep the curve as far as possible from the diagonal.

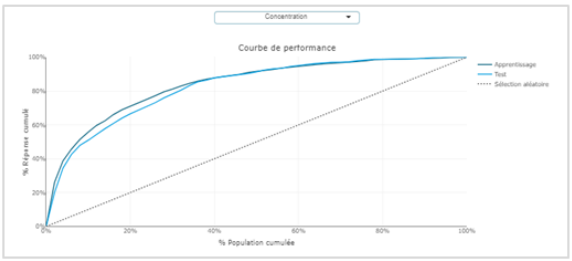


Figure 4.33: Simple_Bundle3 Lift curve

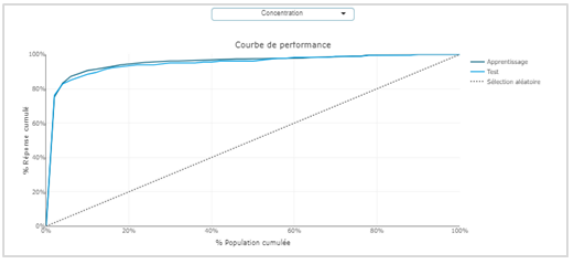


Figure 4.34: Combined_Bundle3 Lift curve

Figure 4.35: Comparing Both Model's Lift curve for the Bundle3

For the Simple_Bundle3 score, among the 20% of the best individuals according to the learning model, we find 71% of the total number of respondents.

While for the combined_Bundle3 score, among the 20% of the best individuals according to the learning model, we find 94% of the total number of respondents.

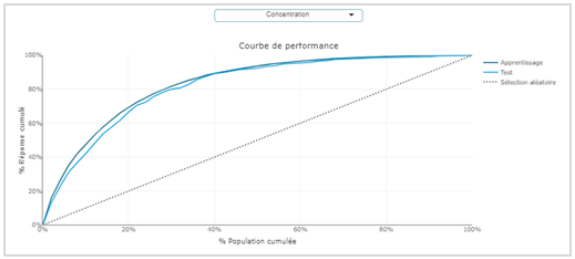


Figure 4.36: Simple_Bundle5 Lift curve

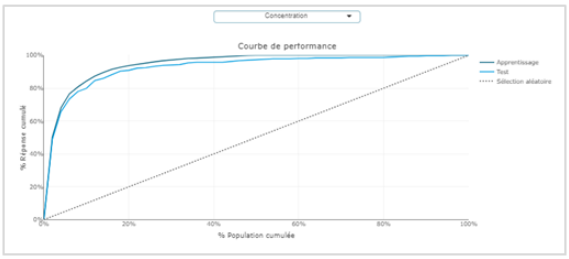


Figure 4.37: Combined_Bundle5 Lift curve

Figure 4.38: Comparing Both Moled's Lift curve for the Bundle5

For Simple_Bundle5 score, among the 20% of the best individuals according to the learning model, we find 69% of the total number of respondents.
As for the combined_Bundle5 score, among the 20% of the best individuals according to the learning model, we find 94% of the total number of respondents.

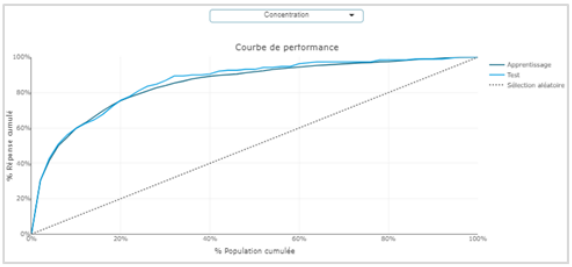


Figure 4.39: Simple_Bundle8 Lift curve

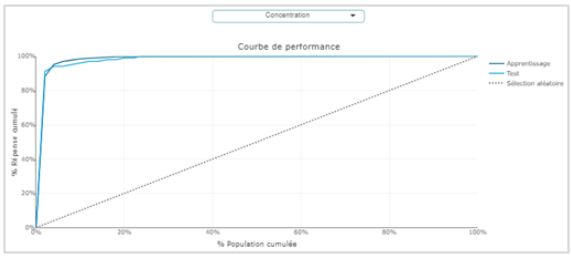


Figure 4.40: Combined_Bundle8 Lift curve

Figure 4.41: Comparing Both Moled's Lift curve for the Bundle8

For Simple_Bundle8 score, among the 20% of the best individuals according to the learning model, we find 75% of the total of respondents.
Whilst for the combined_Bundle8 score, among the 20% of the best individuals according to the learning model, we find 80% of the total number of respondents.

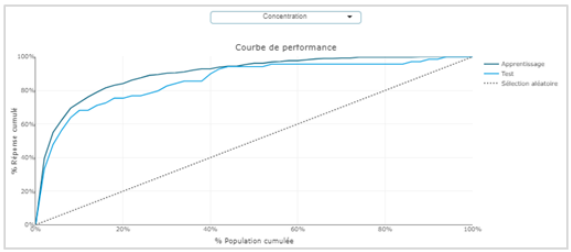


Figure 4.42: Simple_Bundle12 Lift curve

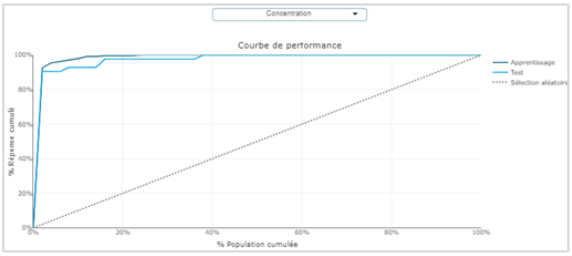


Figure 4.43: Combined_Bundle12 Lift curve

Figure 4.44: Comparing Both Moled's Lift curve for the Bundle12

Finally, for the Simple_Bundle12 score, among the 20% of the best individuals according to the learning model, we find 84% of the total number of respondents.
And for the combined_Bundle12 score, among the 20% of the best individuals according to the learning model, we find 80% of the total number of respondents.

On the whole , Lift curves confirm the performance of the models by evaluating the resulting economic value. However, for scores Combined_Bundle8 and Combined_Bundle12, we note the effects of overfitting, explained by the condensation of the totality of respondents in the last score class and curves are almost equal to 0 throughout the other classes.

Business Performance Evaluation

At this stage, even if each selected model has a good technical performance, it is crucial to evaluate its performance from the Marketing perspective using the Marketing indicators explained in the following section.

Top X & Bottom X

The first Marketing indicators to define are the TopX and the BottomX. TopX is one of the most used marketing indicators by D-AIM, it helps refining the targeting strategy by shifting the focus towards the individuals with higher chance of responding to the observed event which help reduce the targeting costs.
In fact, TopX is a standard allowing the visualization of the potential number or percentage of respondents among the X% of the total population having obtained the best scores of the model.

Three levels are selected for this key indicator 5, 10 and 20:

- Top20: Determination of the percentage of respondents if we select the 20% of best individuals.
- Top10: Determination of the percentage of respondents if we select the 10% of best individuals.

- Top5: Determination of the percentage of respondents if we select the 5% of best individuals.

Actually, it is up to the telephone operator to choose the level that suits him the most, that is to say the level that ensures the best targeting strategy under cost constraints.

As part of our project, we will consider the Top10 to compare the performance of different scores. However, we will present the results for the Top5 and Top20 as well.

BottomX on the other hand, is a marketing indicator established to better evaluate the performance of the deployed scores.

As mentioned earlier, we always tend to rank the individuals according to their score marks and we proceed by targeting the best ones. However, it is useful to quantify the percentage of respondents that are left untargeted due to modeling deficiency.

The BottomX indicator helps detect the percentage of respondents among the X% with the lowest score marks.

For this indicator as well, three levels are selected: 5, 10 and 20.

- Bottom20: Determination of the percentage of respondents if we select the 20% of individuals with the lowest scores.
- Bottom10: Determination of the percentage of respondents if we select the 10% of individuals with the lowest scores.
- Bottom5: Determination of the percentage of respondents if we select the 5% of individuals with the lowest scores.

To evaluate the performance of the different bundles, we will compare the performance (i.e. the response rate) of these different marketing indicators.

Figure 4.45: Evaluation of Simple_Bundle3 & Combined_Bundle3 according to Top10 Indicator

Score	Method	Top5	Top10	Top20	Bottom5	Bottom10	Bottom20
Bundle3 & Non-Fragile Data	Gradient Boosting	85.08%	88.40%	93.37%	0.00%	0.00%	0.55%
Bundle3	Gradient Boosting	39.28%	51.63%	66.90%	0.17%	0.69%	2.06%

Generally, we notice that the combined scores assure a better performance that the simple ones. As an instance, the figure above shows that thanks to the score combined_Bundle3 score, over 88% of the total number of the respondent can be reached by selecting only the top 10% of the population. Whereas, for the Simple_Bundle3 score only 52% of the respondent only can be scoped.

Moreover, the combined score show lightly better performance according to the Bottom indicators.

Figure 4.46: Evaluation of Simple_Bundle5 & Combined_Bundle5 according to Top10 Indicator

Score	Method	Top5	Top10	Top20	Bottom5	Bottom10	Bottom20
Bundle5 & Non-Fragile Data	Gradient Boosting	65.32%	80.11%	86.29%	0.27%	0.54%	0.81%
Bundle5	Gradient Boosting	29.92%	45.54%	67.22%	0.55%	0.66%	1.32%

The same implies for the Combined_Bundle5 score, which helps us predict about 80% of the respondents, while the Simple_Bundle5 score predicts only 45.5% of the total respondents.

Figure 4.47: Evaluation of Simple_Bundle8 & Combined_Bundle8 according to Top10 Indicator

Score	Method	Top5	Top10	Top20	Bottom5	Bottom10	Bottom20
Bundle8 & Non-Fragile Data	Gradient Boosting	94.06%	96.04%	98.02%	0.00%	0.00%	0.00%
Bundle8	Gradient Boosting	49.74%	64.02%	77.78%	0.53%	1.59%	4.23%

Similarly, the Combined_Bundle8 score helps detect over 90% of the total respondents by selecting the top 10%, unlike the Simple_Bundle8 that helps detect 64% of the respondent, only.

Figure 4.48: Evaluation of Simple_Bundle12 & Combined_Bundle12 according to Top10 Indicator

Score	Method	Top5	Top10	Top20	Bottom5	Bottom10	Bottom20
Bundle12 & Non-Fragile Data	Random Forest	90.48%	92.86%	97.62%	0.00%	0.00%	0.00%
Bundle12	Random Forest	52.17%	60.87%	66.67%	1.45%	2.90%	4.35%

At last, a similar scenario is established for the Combined_Bundle12 score, as shown in the figure above, over 92% of the respondent are scoped in the top 10% of the population, while for the Simple_Bundle12 score only 60% are detected.

Result by Score Class

A very effective way to analyze the score 's performance from the Marketing perspective is to create 20 score classes, after having classified the probabilities of the score in descending order.
The process is simple, after selecting for each score the model that predicts it the best, the scored customers are ranked according to their score marks i.e. their probability of responding to the observed event and divided into 20 classes, from 0 to 19. 0 corresponds to the customers with the lowest marks, 19 corresponds to the customers with the highest marks.

As a result, the response rate is always increasing, going from the least appetent class, to the score classes containing the most appetent customers.

This approach allows us to visually evaluate the performance by class of score which help refining

the targeting strategies by focusing on the classes with the highest response rate, i.e. the class 19.

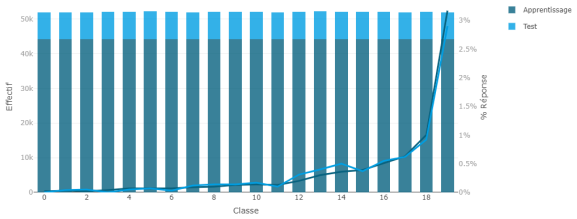


Figure 4.49: Simple_Bundle3 score class

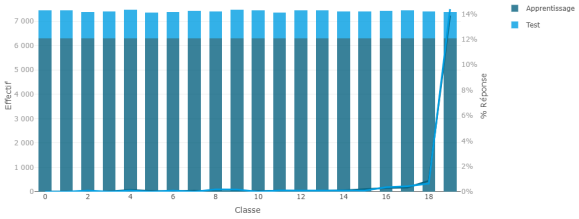


Figure 4.50: Combined_Bundle3 score class

Figure 4.51: Response Rate/score class for Both Bundle3 Models

For both scores, the two learning and validation curves are increasing in harmony with the presence of a slight elbow effect in the 18th class. This indication can help us deduce that from this class the response probability becomes higher.
However, this effect is more present in the combined scores which indicates their higher efficiency when it comes to locating the truly responding customers, (For more details, see appendix).

For the Simple_Bundle3 score, the 19th Class contains 44,256 customers on the construction sample. Among these customers, the response rate is 3.17%. As for the test sample, it contains 7,752 customers, among them, the response rate is 2.9%.
However, the combined_Bundle3 score's 19th class contains 6,303 customers on the construction sample. with a response rate of 13.88%. And it contains 1,063 customers on the test sample, with a response rate of 14.39%.

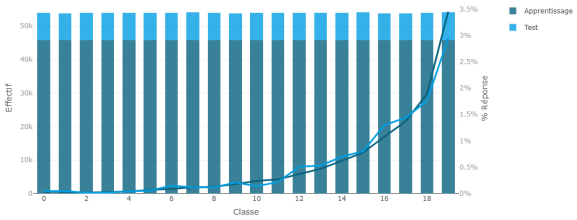


Figure 4.52: Simple_Bundle5 score class

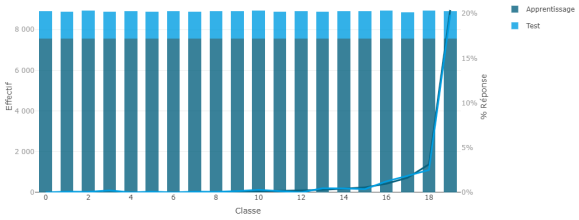


Figure 4.53: Combined_Bundle5 score class

Figure 4.54: Response Rate/score class for Both Bundle5 Models

The same goes for Simple_Bundle5 score. For the construction sample, the 19th Class's response rate is 3.45%. As for the test sample it It contains 8,311 customers with a response rate of 2.95%. The response rate is even higher for combined_Bundle5. For the construction sample it is equal to 20.4%, while for the test sample, the response rate is 19.85%.

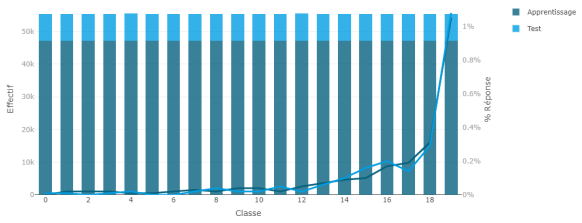


Figure 4.55: Simple_Bundle8 score class

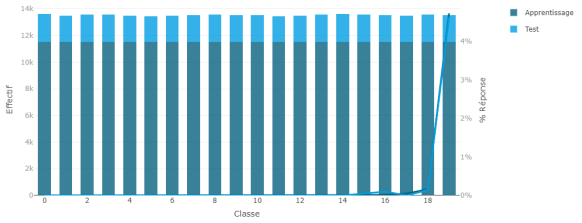


Figure 4.56: Combined_Bundle8 score class

Figure 4.57: Response Rate/score class for Both Bundle8 Models

For the Simple_Bundle8 score, the 19th Class's the response rate is 1.08% for both the construction sample and test samples.
While for the combined_Bundle8 score, the 19th Class contains 11,499 customers on the construction sample and 2,046 customers on the test sample, with a response rate of 4.73%.

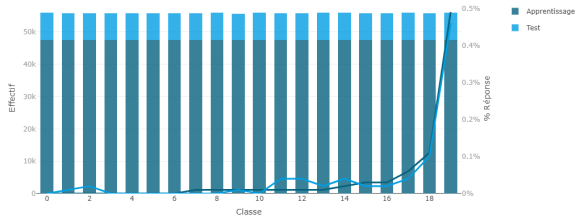


Figure 4.58: Simple_Bundle12 score class

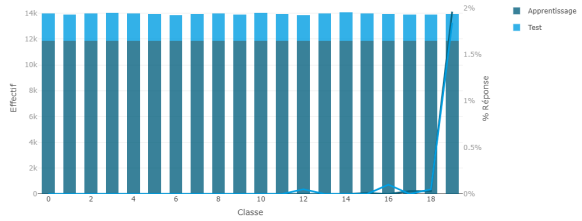


Figure 4.59: Combined_Bundle12 score class

Figure 4.60: Response Rate/score class for Both Bundle12 Models

Finally, we note that the Simple_Bundle12 score's response rate is insignificant since among the 47,463 customers on the construction sample's 19th class only 0.49% tend to be respondent. And among the 8,437 customers on the test sample only 0.46% are likely to respond.

However the response rate increases scientifically for the combined_Bundle12 score, since 1.96% of the construction sample's customers and 1.84% of the test sample's customers are likely to respond to the predicted events.

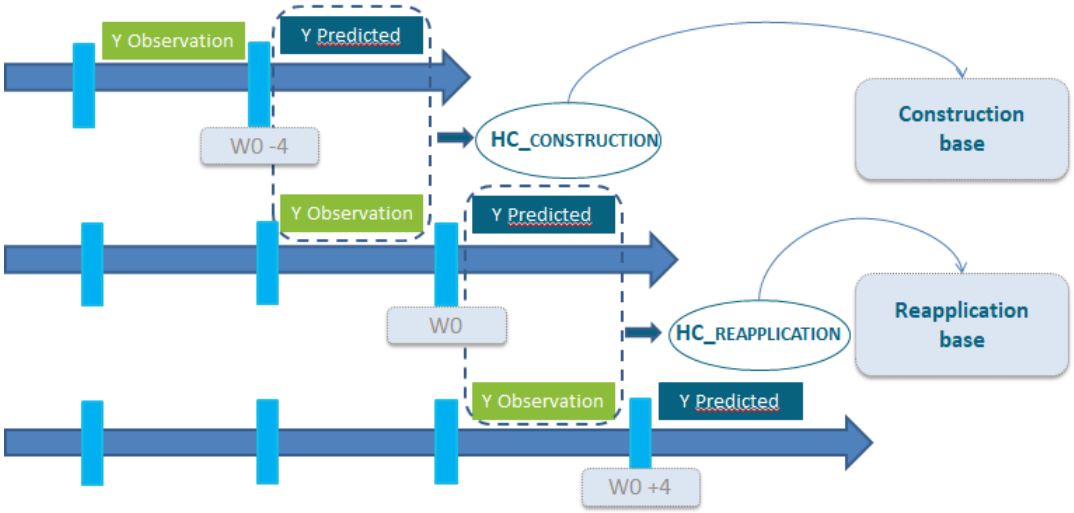
To sum up, the results by class score are consistent with those of the Lift curve; we notice a better performance for the combined scores with a possibility of over learning for the scores Combined_Bundle8 and Combined_Bundle12.

4.1.4 Validation of the Predictions: HealthCheck

In order to help adjust the overfitting problem, as well as help calibrating the machine learning model, we will recours to the health check process.

The Healthcheck is a process implemented by D-AIM, in order to validate a previous prediction based on the post-observation of the phenomenon, as illustrates the following figure:

Figure 4.61: The Healtcheck Process



Similar to the scoring methodology, the Healthcheck process is build upon two phases, the Construction phase, and the Reapplication phase.

The construction phase Healthcheck is implemented in order to, first, compare the actual output and the expected output given by a system. Then, in case of imbalance, enhance the self-learning aspect of the existing model and as a result overcome the issue of overfitting.

The reapplication phase, on the other hand, is established as post processing techniques to evaluate the real outcomes of the targeting strategy.

4.1.5 Return On Investment

The table below displays, for each bundle, the result of the combination with the Non-Fragile Data, according to both the basic and proposed approaches. The table will detail the results in terms of Response Rate, Number of targeted Clients as well as additional Revenue (measured by units of currency UoC)

Figure 4.62: RIO of the Proposed Solution

Score	Number of Class 19 Respondents Basic Approach	Number of Class 19 Respondents Proposed approach	Incremental in terms of Response Rate	Incremental in terms of Client Number	Incremental in terms of Data Revenue
Appetency Bundle 3	456	877	16,53%	421	2105
Appetency Bundle 5	181	270	3,10%	89	712
Appetency Bundle 8	54	636	4,76%	582	6984
Appetency Bundle 12	32	270	1,94%	238	3808

It should be noted that the results presented relate exclusively to the score class19 and not the entire base, because, as we have developed previously, it is the class with the highest response

rate possible.

For each combination, we can easily notice that the precision offered by the new approach shows that the customers who will be finally solicited and who will be more likely to be reactive are more numerous than those detected by the old approach. Which ensures us an incremental in terms of probability of departure, as well as the income to generate.

In numbers, the Combined_Bundle3 score allowed us to predict 421 more customers. As mentioned earlier, these clients are Non-Fragile Data which justifies our interest to solicit them with the Bundle5 offer.

Therefore, this approach will allow us to generate an additional income of $\mathbf{5} \times \mathbf{421} = \mathbf{2105\ UoC}$.

To the same extent, the Combined_Bundle5 score allows us to predict 89 more eligible customers, and therefore achieve an additional revenue of $\mathbf{8} \times \mathbf{89} = \mathbf{712\ UoC}$.

For the Combined_Bundle8 score as well, a higher number of eligible customers is located. The associated revenue is equal to $\mathbf{12} \times \mathbf{582} = \mathbf{6984\ UoC}$.

As for the Combined_Bundle12 score, 238 additional customers are scoped and after targeting them with the Bundle12 an additional revenue of result $\mathbf{16} \times \mathbf{238} = \mathbf{3808\ UoC}$ is assured.

Overall, an additional revenue of $\mathbf{2105} + \mathbf{712} + \mathbf{6984} + \mathbf{3808} = \mathbf{13609\ UoC}$ is assured thanks to the proposed approach.

4.1.6 Conclusion

Throughout this section, we firstly introduced and applied the different stages of conception of the predictive scores that predict the desired events.

Secondly, we compared the performance of the scores deployed using technical indicators (introduced in the previous chapters) as well as marketing ones.
At the end, we established a comparison of the performances of the two proposed approaches in order to be able to choose the approach which ensures better performance.

4.2 Exploitation of Predictive Scoring

4.2.1 Introduction

As stated in the introductory chapter, client targeting, or the process of selecting the suitable individuals for each product or service, is the main activity of D-AIM, since it enhances the loyalty of customers and as a result the guarantees a growing market share.

In this regard, this section of the practical chapter is dedicated to introducing the targeting strategy of D-AIM, highlight the importance of the predictive scoring output it the process of targeting and finally exploit these outputs to further refine the targeting strategy.

4.2.2 NBA : D-AIM's Individualized Targeting Strategy

NBA, or the Next Best Action: is D-AIM's individualized targeting strategy, it's the fact of allocating to each client the opportunity that best fits his behavior at a certain time period, usually 3 to 5 days.

As known, with each telecommunication project comes an offer catalog, depending on which, D-AIM will create the targeting opportunities.

An opportunity is created based on a unique combination of an offer, a set of eligibility rules and a score.

In order to detect the best fit opportunity for each client, we go through 3 steps:

First of all, we start by detecting for each client all the scores for which he is eligible; usually we end up with more than one. Then we extract the corresponding score marks, since they will assist the choice of the proper opportunity.

Afterwards, we consult the list of eligibility rules, designed by the marketing experts. These rules are observed from the client's historical variables, such as the frequency and amount of recharging, the most purchased offers, the usage potential, price elasticity and the value segmentation as well as the potential revenue.
An eligibility rule is a 0 or 1 variable and will decide if the customer is eligible for the offer in question, or not. Here's an example of eligibility rules.

Last but not least, if we want to prioritize a specific offer, which is usually a business-driven decision, we can add a "weight" in the equation, as well as the offer's price.

Finally, an opportunity is the combination of the score, the Weight and the eligibility rules:

$$Opportunity = Score \times EligibilityRules \times OfferWeight \times OfferPrice$$

However, the principle of the NBA is based on prioritizing to each customer the offer that suits him the most and at the same time that generates the most income.
In this context, we are often forced to choose one of two or even several score, as illustrated by the following figure:

Figure 4.63: Targeting Scenario

Customer	scores	Probability
Customer X	score1	Px1
Customer X	score2	Px2
Customer X	score3	Px3

We are often obliged to arbitrate, for a single score, which people are the most eligible, or for a single client, which offer suits him the most. In both cases, we are led to compare the corresponding score marks.

4.2.3 Exploit the Score Marks or Probabilities

Before exploiting the score marks or the Machine Learning generated probabilities, we first need to understand their origin as well as some of their characteristics.

Probability Origin

It is undeniable that probability is the Bedrock of Machine Learning. As a matter of fact, Algorithms are designed, trained and configured using probability (e.g. Naive Bayes). Similarly, Model hyper-parameters are configured with probability.
Moreover, classification models must predict a probability of class membership, depending on which decision will be made.

In this context, we have deployed Machine Learning algorithms predicting the potential preference of customers. The output of these algorithms, i.e. the probabilities describing the appetency level to an offer or likelihood of responding to a solicitation, will be utilized in the Marketing strategy of D-AIM.

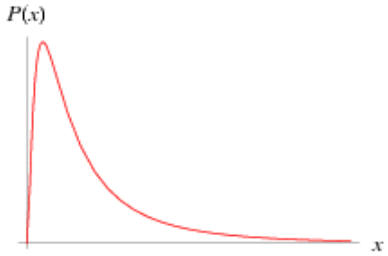
Probability Characteristics

For each predicted event, the generated probabilities have a set of characteristics, which helps to further understand and exploit them.

Among these characteristics, we mention the probability distribution; that will help us know the overall tendencies of the customers. Furthermore, the probability distribution has two parameters, namely the mean which is a measure of the central tendency of the population. And the standard deviation that measures the dispersion of a population.

All the deployed scores have a similar distribution: the Log-Normal distribution (shown in the figure below) yet with different means and standard deviations, which makes it difficult to compare different events.

Figure 4.64: Log-Normal density



4.2.4 Standardization Use Case

The process of selecting the appropriate opportunity to each customer is as follows:
First of all, we start by locating, for each customer, all the scores he is eligible to i.e. he is included in the perimeter of the score.
Afterwards, we extract his score mark, or his probability of responding to the predicted event, as well as the score class he belongs to, since customer some opportunities are defined by the score class as well as the score, the eligibility rules, the weight and the price.
Finally, we detect the opportunities that correspond to each score and score class of the client at hand and then we will arbitrate the choice of the appropriate opportunity to allocate to this client.

The idea is to evaluate, for each predicted event, the performance of the customer in consideration, compared to the overall performance, i.e. of the whole base (performance average). Accordingly, we will end up choosing the opportunity which corresponds to the score where we has the most important individual performance.

However, comparing probabilities with different means and variances does not seem legitimate and therefore moving to the standardization of probabilities is an essential step.

Standardization is the fact of expressing each score mark, or probability as the number of standard deviations unites away from the mean aka the average probability of the score. After the transformation, the distribution mean of these z scores is 0, and the standard deviation is 1. The following figure will present the mean and standard deviation of the previously deployed scores before and after the standardization step.

Figure 4.65: Score Standardization Results

Score	Mean	Standard Deviation	Mean Post Standardization	Standard Deviation Post Standardization
Appetency Bundle 3	0.0099	0.0621	0	1
Appetency Bundle 5	0.0139	0.0583	0	1
Appetency Bundle 8	0.0024	0.0274	0	1
Appetency Bundle 12	0.0014	0.0213	0	1

In order to detect how well a customer performed compared to the total performance, knowing the mean and the variance, we simply need to calculate the z-score of the considered customer, given by:

$$z = \frac{scoremark - populationmean}{standarddeviations}$$

After establishing the correspondent Z-score, we can easily deduce the number or percentage of customer having higher and lower scores than our client and as a result decide if he will be allocated the opportunity or not. To do this, we need to refer to the normalization of the z-score.

The normalized z-score can be interpreted as the probability of a score being greater than our score. Which will give us the percentage (and thus the number) of clients that could be more likely to respond to the predicted event and based on the resulting value as well as the set of eligibility rules we can decide whether or not this opportunity is the one to allocate to that client.

The Standardization process is also useful for combining the probabilities of different scores constructing the same opportunity

4.2.5 Conclusion

Throughout this section, we first of all presented the targeting strategy of D-AIM ; NBA. Furthermore, we introduced the need to compare probabilities or score marks in the Client Targeting process. And finally, we proposed the solution that will allow this comparison and subsequently, we can select for any client, the opportunity that suits him the most.

4.3 Conclusion

This last chapter was dedicated to and respond to the problematic introduced earlier. In its course, we first of all explained and compared the different approaches of combining scores and concluded by exposing the profits guaranteed if the proposed approach is adopted.

Afterwards, we detailed the targeting strategy of D-AIM, focused on the importance of the ML generated probabilities in this process and proposed a technique allowing the comparison of these probabilities in the aim of refining the targeting strategy.

GENERAL CONCLUSION AND PERSPECTIVES

The evolution of learning machines has offered great flexibility in the choice of algorithms, these data-based techniques are becoming the decision makers in what comes to the business and targeting strategies. However, during the theoretical part of our project, we decided to focus all our attention on specific types of supervised classification algorithms which is the ensemble learning after pointing out to all his advantages and added value comparing to the simple learning algorithms. Moreover, we defined the different evaluation metrics of these Machine Learning algorithms, useful to decide upon the most performing model.
These scoring algorithms are deployed to evaluate and improve a current technique that consist of combining two predictive scores for Business finalities.

In the concrete case of application, we proceeded first by presenting and applying the basic approach for combining two scores, which consists in deploying and two machine learning models to independently predict the occurrence of each event, then selecting the customers predicted responding to both events. In another measure we have proposed an innovative approach to combine the scores and which consists in deploying a single machine learning model making it possible to predict the two events simultaneously and extract the responding customers.

Finally, we also introduced the technical and business results of the two approaches so that we can establish a comparison between them and ultimately determine which one to adopt.
Similarly, we could explain the cause of the improved performance with the combined score, which is due to the restriction of the scope or perimeter, as well as the importance of the initial response rate, also known as the default probability, of the combined scores compared to the single scores.

As for the second part of this project, that relates to the exploitation of the outputs of machine learning models in order to refine the targeting strategy. We were able to define, in a broad and general way, the most fundamental concepts of the probability theory which enabled us to propose the technique of standardization and normalization of probabilities to allow the

comparison between the performance of different clients and ultimately decide about the most legitimate customer for each opportunity.

The contribution of this work has been enormous. It presented an opportunity to become more familiar more with supervised learning techniques in general and scoring in particular as well as basic concepts of probability and their utility in the scoring and targeting processes. It also allowed the acquiring of a broad business knowledge which guarantees a great professional maturity.

However, this does not exclude the fact that of encountering some difficulties related mainly to understanding the different business aspects and problematic as well as the comprehension and application of the different metrics allowing to evaluate the performance from a business point of view.

At the end, other improvements can be proposed to further evaluate this work, for instance, we can proceed by targeting the customers and compare their real-life behavior to the predicted results one, using the principle of health-Check.
In addition, from the perspective of target cost and target result, the method proves to be effective. This leads us to believe that if it is expanded and tested in several other projects and in several other scoring types, it can generate new revenue opportunities.

References

[1] D-AIM. Presentation of the company. D-aim.fr/. Last accessed on 02/2020.

[2] About D-AIM. linkedin.com/company/d-aim/. Last accessed on 02/2020.

[3] Lucy Li. What is Predictive Lead Scoring? Last accessed on 03/2020.

[4] DataRobot AI plateform. Topic:Scoring Data. Last accessed on 03/2020.

[5] Datacamp. Topic: What's Machine Learning. Last accessed on 03/2020.

[6] Paroma Varma, Braden Hancock and Chris Ré. Debug Training Data for Software. Last accessed on 03/2020.

[7] Hunter Heidenreich. What are the types of machine learning. Towards data science. Last accessed on 03/2020

[8] GASSO, Gilles. Logistic regression. 2019.

[9] ROCCA, Joseph. Ensemble methods: bagging, boosting and stacking. Towards Data Science, 2019, p. 1-21.

[10] Sunil Ray. 6 Easy Steps to Learn Naive Bayes Algorithm. Analytics Vidhya. Last accessed on 03/2020.

[11] Mehri Haghighi. Data mining and machine learning : an overview of classifiers. Department of Computer Engineering, Payam Noor University, Sosangerd, Iran,11 :76–86, 2015.

[12] Francois Denis : Chapitre 2 - les arbres de décisio. http ://pageperso.lif.univ-mrs.fr/ francois.denis/IAAM1/chap2.pdf

[13] Zulaikha Geer. Why Is Boosting Used?. Quora. Last accessed on 04/2020.

[14] ABRAICH, Ayoub, NGUYEN, Hoang Dung, et TOUNSI, Mohamed. DÉTECTION DE FRAUDE IEEE-CIS. 2020.

[15] Marie Chavent : Bagging et forêts aléatoires. Master MIMSE - Université de Bordeaux,2015-2016.

[16] Gustavo Machado and Mariana Recamonde-Mendoza. A Random Forest Approach. ResearchGate. Last accessed on 05/2020.

[17] ALSURAIHI, Waad, AL-HAZMI, Ekram, BAWAZEER, Kholoud, et al. Machine Learning Algorithms for Diamond Price Prediction. In : Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing. 2020. p. 150-154.

[18] Shinir Glander. Machine Learning Basics - Gradient Boosting XGBoost. Last accessed on04/2020

[19] Krishna Kumar Mahto. : Demystifying maths of gradient boosting. Towards Data Science. Last accessed on 04/2020

[20] GROVER, Prince. Gradient Boosting from scratch. Retrieved from Medium, 2017.

[21] SIKORA, Riyaz, et al. A modified stacking ensemble machine learning algorithm using genetic algorithms. In : Handbook of Research on Organizational Transformations through Big Data Analytics. IGi Global, 2015. p. 43-53.

[22] WEXLER, James, PUSHKARNA, Mahima, BOLUKBASI, Tolga, et al. The what-if tool: Interactive probing of machine learning models. IEEE transactions on visualization and computer graphics, 2019, vol. 26, no 1, p. 56-65.

[23] Krishni. K-Fold Cross Validation. Meduim. Last accessed on 05/2020.

[24] Dishashree Gupta. Basics of Probability for Data Science explained with examples in R. Analytics Vidhya. Last accessed on 05/2020

[25] KYUNG, Andrew et NAM, Steve. Study on Unemployment Rate in USA Using Computational and Statistical Methods. In : 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON). IEEE, 2019. p. 0987-0993.

[26] DEAKIN, Edward B. Distributions of financial accounting ratios: some empirical evidence. The Accounting Review, 1976, vol. 51, no 1, p. 90-96.

[27] Allen B. Downey. Probability and Statistics for Programmers. Version 1.6.0. 2017

[28] Probability Rules. Boundless Statistics-Lumen Canada.

[29] TO, Statistics How. Levenberg–Marquardt Algorithm (Damped Least Squares): Definition Share on.

[30] FINLAY-JONES, Robert. Factors in the teaching environment associated with severe psychological distress among school teachers. Australian and New Zealand Journal of Psychiatry, 1986, vol. 20, no 3, p. 304-313.

[31] Standard Score. Statistics Laerd. Last accessed on 05/2020.

[32] PhD. BROWNLEE, Jason. Machine learning mastery with python. Machine Learning Mastery Pty Ltd, 2016, p. 100-120.

Annex

Result table by score class for the simple and combined scores.

Figure 4.66: simple_Bundle3 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	44 257	5.00%	0.02%	7 744	4.96%	0.00%
1	44 256	5.00%	0.03%	7 753	4.96%	0.04%
2	44 256	5.00%	0.02%	7 744	4.96%	0.05%
3	44 256	5.00%	0.04%	7 795	4.99%	0.00%
4	44 256	5.00%	0.07%	7 794	4.99%	0.05%
5	44 256	5.00%	0.07%	8 045	5.15%	0.07%
6	44 257	5.00%	0.07%	7 910	5.06%	0.03%
7	44 256	5.00%	0.09%	7 735	4.95%	0.12%
8	44 256	5.00%	0.10%	7 804	5.00%	0.14%
9	44 256	5.00%	0.13%	7 854	5.03%	0.14%
10	44 256	5.00%	0.14%	7 844	5.02%	0.17%
11	44 256	5.00%	0.13%	7 692	4.92%	0.10%
12	44 256	5.00%	0.20%	7 792	4.99%	0.31%
13	44 257	5.00%	0.30%	7 950	5.09%	0.40%
14	44 256	5.00%	0.36%	7 819	5.01%	0.50%
15	44 256	5.00%	0.39%	7 772	4.98%	0.37%
16	44 256	5.00%	0.51%	7 858	5.03%	0.55%
17	44 256	5.00%	0.63%	7 704	4.93%	0.62%
18	44 256	5.00%	1.00%	7 838	5.02%	0.92%
19	44 256	5.00%	3.17%	7 752	4.96%	2.90%
Total	885 123	100.00%	0.37%	156 199	100.00%	0.37%

Figure 4.67: combined_Bundle3 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	6 304	5.00%	0.00%	1 142	5.13%	0.00%
1	6 303	5.00%	0.00%	1 142	5.13%	0.00%
2	6 304	5.00%	0.02%	1 069	4.80%	0.09%
3	6 303	5.00%	0.02%	1 103	4.96%	0.00%
4	6 304	5.00%	0.14%	1 163	5.23%	0.09%
5	6 304	5.00%	0.06%	1 058	4.76%	0.00%
6	6 303	5.00%	0.03%	1 074	4.83%	0.09%
7	6 303	5.00%	0.08%	1 119	5.03%	0.00%
8	6 304	5.00%	0.03%	1 097	4.93%	0.18%
9	6 303	5.00%	0.02%	1 174	5.28%	0.17%
10	6 304	5.00%	0.05%	1 135	5.10%	0.00%
11	6 304	5.00%	0.08%	1 050	4.72%	0.10%
12	6 303	5.00%	0.03%	1 145	5.15%	0.09%
13	6 304	5.00%	0.06%	1 150	5.17%	0.09%
14	6 303	5.00%	0.10%	1 093	4.91%	0.09%
15	6 303	5.00%	0.19%	1 094	4.92%	0.09%
16	6 304	5.00%	0.29%	1 132	5.09%	0.35%
17	6 303	5.00%	0.33%	1 153	5.18%	0.43%
18	6 304	5.00%	0.86%	1 092	4.91%	0.64%
19	6 303	5.00%	13.88%	1 063	4.78%	14.39%
Total	126 070	100.00%	0.81%	22 248	100.00%	0.81%

Figure 4.68: simple_Bundle5 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	7 564	5.00%	0.00%	1 327	4.97%	0.00%
1	7 564	5.00%	0.00%	1 318	4.94%	0.08%
2	7 564	5.00%	0.00%	1 364	5.11%	0.07%
3	7 564	5.00%	0.00%	1 306	4.89%	0.23%
4	7 564	5.00%	0.00%	1 333	4.99%	0.00%
5	7 564	5.00%	0.00%	1 312	4.91%	0.08%
6	7 563	5.00%	0.00%	1 348	5.05%	0.00%
7	7 564	5.00%	0.00%	1 324	4.96%	0.08%
8	7 564	5.00%	0.00%	1 346	5.04%	0.07%
9	7 564	5.00%	0.00%	1 348	5.05%	0.15%
10	7 564	5.00%	0.12%	1 358	5.09%	0.29%
11	7 564	5.00%	0.16%	1 328	4.97%	0.15%
12	7 564	5.00%	0.22%	1 332	4.99%	0.00%
13	7 563	5.00%	0.29%	1 303	4.88%	0.46%
14	7 564	5.00%	0.41%	1 346	5.04%	0.45%
15	7 564	5.00%	0.56%	1 353	5.07%	0.37%
16	7 564	5.00%	0.98%	1 378	5.16%	1.24%
17	7 564	5.00%	1.64%	1 290	4.83%	1.86%
18	7 564	5.00%	3.13%	1 357	5.08%	2.51%
19	7 563	5.00%	20.40%	1 330	4.98%	19.85%
Total	151 277	100.00%	1.40%	26 697	100.00%	1.39%

Figure 4.69: combined_Bundle5 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	7 564	5.00%	0.00%	1 327	4.97%	0.00%
1	7 564	5.00%	0.00%	1 318	4.94%	0.08%
2	7 564	5.00%	0.00%	1 364	5.11%	0.07%
3	7 564	5.00%	0.00%	1 306	4.89%	0.23%
4	7 564	5.00%	0.00%	1 333	4.99%	0.00%
5	7 564	5.00%	0.00%	1 312	4.91%	0.08%
6	7 563	5.00%	0.00%	1 348	5.05%	0.00%
7	7 564	5.00%	0.00%	1 324	4.96%	0.08%
8	7 564	5.00%	0.00%	1 346	5.04%	0.07%
9	7 564	5.00%	0.00%	1 348	5.05%	0.15%
10	7 564	5.00%	0.12%	1 358	5.09%	0.29%
11	7 564	5.00%	0.16%	1 326	4.97%	0.15%
12	7 564	5.00%	0.22%	1 332	4.99%	0.00%
13	7 563	5.00%	0.29%	1 303	4.88%	0.46%
14	7 564	5.00%	0.41%	1 346	5.04%	0.45%
15	7 564	5.00%	0.56%	1 353	5.07%	0.37%
16	7 564	5.00%	0.98%	1 376	5.15%	1.24%
17	7 564	5.00%	1.64%	1 290	4.83%	1.88%
18	7 564	5.00%	3.13%	1 357	5.08%	2.51%
19	7 563	5.00%	20.40%	1 330	4.98%	19.85%
Total	151 277	100.00%	1.40%	26 697	100.00%	1.39%

Figure 4.70: simple_Bundle8 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	47 080	5.00%	0.00%	8 230	4.96%	0.01%
1	47 079	5.00%	0.02%	8 221	4.95%	0.01%
2	47 080	5.00%	0.02%	8 243	4.96%	0.00%
3	47 079	5.00%	0.02%	8 322	5.01%	0.01%
4	47 079	5.00%	0.01%	8 410	5.06%	0.02%
5	47 080	5.00%	0.01%	8 275	4.98%	0.00%
6	47 079	5.00%	0.02%	8 255	4.97%	0.00%
7	47 079	5.00%	0.03%	8 341	5.02%	0.02%
8	47 080	5.00%	0.02%	8 312	5.00%	0.04%
9	47 079	5.00%	0.04%	8 217	4.95%	0.02%
10	47 079	5.00%	0.04%	8 352	5.03%	0.02%
11	47 080	5.00%	0.02%	8 254	4.97%	0.05%
12	47 079	5.00%	0.05%	8 508	5.12%	0.02%
13	47 079	5.00%	0.07%	8 260	4.97%	0.06%
14	47 080	5.00%	0.09%	8 270	4.98%	0.10%
15	47 079	5.00%	0.10%	8 330	5.01%	0.16%
16	47 079	5.00%	0.17%	8 332	5.01%	0.20%
17	47 080	5.00%	0.19%	8 349	5.02%	0.14%
18	47 079	5.00%	0.31%	8 311	5.00%	0.29%
19	47 079	5.00%	1.05%	8 361	5.03%	1.08%
Total	941 587	100.00%	0.11%	166 163	100.00%	0.11%

Figure 4.71: combined_Bundle8 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	11 500	5.00%	0.00%	2 122	5.23%	0.00%
1	11 500	5.00%	0.00%	1 982	4.88%	0.00%
2	11 499	5.00%	0.00%	2 063	5.08%	0.00%
3	11 500	5.00%	0.00%	2 087	5.14%	0.00%
4	11 499	5.00%	0.00%	2 006	4.94%	0.00%
5	11 500	5.00%	0.00%	1 956	4.82%	0.00%
6	11 499	5.00%	0.00%	1 984	4.89%	0.00%
7	11 500	5.00%	0.00%	2 014	4.96%	0.00%
8	11 499	5.00%	0.00%	2 062	5.08%	0.00%
9	11 500	5.00%	0.00%	2 029	5.00%	0.00%
10	11 500	5.00%	0.00%	2 035	5.01%	0.00%
11	11 499	5.00%	0.00%	1 935	4.77%	0.00%
12	11 500	5.00%	0.00%	1 966	4.84%	0.00%
13	11 499	5.00%	0.00%	2 052	5.06%	0.00%
14	11 500	5.00%	0.00%	2 096	5.16%	0.00%
15	11 499	5.00%	0.01%	2 054	5.06%	0.06%
16	11 500	5.00%	0.02%	2 044	5.04%	0.10%
17	11 499	5.00%	0.05%	1 984	4.89%	0.00%
18	11 500	5.00%	0.16%	2 071	5.10%	0.14%
19	11 499	5.00%	4.73%	2 046	5.04%	4.64%
Total	229 991	100.00%	0.25%	40 587	100.00%	0.25%

Figure 4.72: simple_Bundle12 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	47 464	5.00%	0.00%	8 453	5.05%	0.00%
1	47 464	5.00%	0.00%	8 343	4.98%	0.01%
2	47 466	5.00%	0.00%	8 378	5.00%	0.02%
3	47 463	5.00%	0.00%	8 386	5.01%	0.00%
4	47 464	5.00%	0.00%	8 346	4.98%	0.00%
5	47 466	5.00%	0.00%	8 328	4.97%	0.00%
6	47 464	5.00%	0.00%	8 362	4.99%	0.00%
7	47 463	5.00%	0.01%	8 273	4.94%	0.00%
8	47 464	5.00%	0.01%	8 519	5.09%	0.00%
9	47 464	5.00%	0.01%	8 192	4.89%	0.01%
10	47 466	5.00%	0.01%	8 424	5.03%	0.00%
11	47 464	5.00%	0.01%	8 356	4.99%	0.04%
12	47 463	5.00%	0.01%	8 245	4.92%	0.04%
13	47 466	5.00%	0.01%	8 539	5.10%	0.02%
14	47 463	5.00%	0.02%	8 471	5.06%	0.04%
15	47 464	5.00%	0.03%	8 371	5.00%	0.02%
16	47 466	5.00%	0.03%	8 310	4.96%	0.02%
17	47 463	5.00%	0.06%	8 440	5.04%	0.04%
18	47 466	5.00%	0.11%	8 351	4.99%	0.10%
19	47 463	5.00%	0.49%	8 437	5.04%	0.46%
Total	949 280	100.00%	0.04%	167 521	100.00%	0.04%

Figure 4.73: combined_Bundle12 score class

Classe	Apprentissage			Test		
	Effectif	Proportion	Taux de réponse	Effectif	Proportion	Taux de réponse
0	11 860	5.00%	0.00%	2 139	5.11%	0.00%
1	11 860	5.00%	0.00%	2 045	4.89%	0.00%
2	11 860	5.00%	0.00%	2 134	5.10%	0.00%
3	11 859	5.00%	0.00%	2 172	5.19%	0.00%
4	11 860	5.00%	0.00%	2 125	5.08%	0.00%
5	11 860	5.00%	0.00%	2 071	4.95%	0.00%
6	11 859	5.00%	0.00%	2 009	4.80%	0.00%
7	11 860	5.00%	0.00%	2 097	5.01%	0.00%
8	11 860	5.00%	0.00%	2 138	5.10%	0.00%
9	11 859	5.00%	0.00%	2 019	4.82%	0.00%
10	11 860	5.00%	0.00%	2 151	5.14%	0.00%
11	11 860	5.00%	0.00%	2 087	4.99%	0.00%
12	11 860	5.00%	0.00%	1 997	4.77%	0.05%
13	11 859	5.00%	0.00%	2 106	5.03%	0.00%
14	11 860	5.00%	0.00%	2 224	5.31%	0.00%
15	11 860	5.00%	0.01%	2 119	5.06%	0.00%
16	11 859	5.00%	0.00%	2 093	5.00%	0.10%
17	11 860	5.00%	0.03%	2 046	4.89%	0.00%
18	11 860	5.00%	0.03%	2 018	4.82%	0.05%
19	11 859	5.00%	1.96%	2 070	4.95%	1.84%
Total	237 194	100.00%	0.10%	41 858	100.00%	0.10%