Handelshøyskolen BI

GRA 19703 Master Thesis

Thesis Master of Science 100% - W

Predefinert informasjon

Startdato:	16-01-2022 09:00	**Termin:**	202210
Sluttdato:	01-07-2022 12:00	**Vurderingsform:**	Norsk 6-trinns skala (A-F)
Eksamensform:	T		
Flowkode:	202210||10936||IN00||W||T		
Intern sensor:	(Anonymisert)		

Deltaker

Navn:	Espen Andre Iversen og Humberto Andres Trevino Flores

Informasjon fra deltaker

Tittel *:	Classifying profitable customers based on meta data with machine learning
Navn på veileder *:	Jonas Moss

Inneholder besvarelsen konfidensielt materiale?:	Nei	**Kan besvarelsen offentliggjøres?:**	Ja

Gruppe

Gruppenavn:	(Anonymisert)
Gruppenummer:	133
Andre medlemmer i gruppen:	

Contents

Acknowledgments

This thesis was written as part of the Master of Science program in Business Analytics at BI Norwegian Business School.

We want to thank Victor Lazarin, and Ricardo H. Phillips Greene, for the trust, resource allocation, and data. As well as Luis Callejas for the attention, time, and availability beyond regular working hours making sure the time-zone difference was not an issue.

We want to extend special recognition and gratitude to Jonas Moss for the academic and theoretical guidance and the challenges in developing the topic further.

Humberto would like to thank His wife Mia, his son Max, his gestating daughter, parents, friends, and colleagues for the support during his master's studies. Additionally, Humberto would like to thank Espen for being an inspiration and outstanding partner in this research.

Espen would like to thank his colleague Philipp Jung for insightful discussions about data cleaning and pre-processing, his family, friends, and Humberto for being an excellent thesis partner.

We hope this research's outcome allows more people in Mexico with adverse socio-economic contexts to access top-quality higher education opportunities such as UIN's and foster social mobility.

BI NORWEGIAN BUSINESS SCHOOL

Humberto Andres Treviño Flores

Espen André Iversen

Oslo, Norway, June 2022

Abstract

The objective of this master's thesis is to explore if a machine learning model can predict sale outcomes from the metadata generated by potential leads as they navigate the company's website. Additionally, the research aims to identify if this model can be used to create value by improving their commercial process.

The data used to answer the research question in this thesis was obtained by Universidad Insurgentes during the period between January 2020 to March 2022. The initial dataset contained more than 0.5 million samples and 41 attributes. The data preparation consisted of several techniques to address challenges such as high cardinality, missing values, and feature engineering. The final dataset used for training and testing consisted of ~250,000 samples with 56 features.

We train and evaluate the performance of three machine learning models: eXtreme Gradient Boosting (XGBoost), CatBoost, and Light Gradient Boosting Machine (LightGBM), which were all compared and evaluated against a simple logistic regression and the default model profit.

Our study concludes that there is a theoretical potential for profit gain when using machine learning to predict sales on CRM metadata. LightGBM is identified as the best-performing algorithm in the context of this thesis. We recommend a heuristic approach for profit and enrollment maximization and include a nuanced discussion about the implied costs of implementing machine learning to predict sales.

Keywords – Machine Learning, Sales prediction, Metadata, Commercial process, BI

Definitions

Lead – In the context of this thesis, a lead is a potential new student that has shown interest in one of the Universidad Insurgentes (UIN) programs by interacting with any of UIN's digital assets such as websites social media and/or digital ads. In this sense, a lead can be considered a potential new customer.

CRM – Customer Relationship Management. CRM systems allow an organization to manage its interactions with its customers, aggregate the data from multiple communication channels, and use data analysis to study large amounts of information to learn more about their target audiences and how to best cater to their needs.

lead contacting (awareness), lead convincing (intention), sales appointment (desire), and enrollment (action)

Lead Contacting – The attempt of a business to establish direct one-to-one communication with a lead, usually through telephone, SMS, or WhatsApp.

Lead convincing – Conversation and interaction between Universities (business) and leads intending to convince the lead to take an early step into getting to know more about a product.

Sales Appointment – One-to-one interaction made physically at the sales point or university with a lead to showcase the product with the ultimate goal of enrolling the lead.

Enroll / Enrollment - If a lead enters one of UIN's programs, this lead is now considered to be enrolled. It can be used as a synonym for "Sale." The goal is to maximize the number of enrolled leads, generating revenue.

Imbalanced data – Imbalanced data refers to a dataset consisting of two or more classes where the classes are unevenly distributed. For example, in our thesis, a class represents the action of enrolling into an education program, which translates into a binary action of "yes" or "no." Our data consists of a vast number of leads with the class label "no" and relatively few leads with the class label "yes."

Asymmetric cost – relates to the cost associated with wrongly classifying the minority class, compared to incorrectly classifying the majority class. In our thesis, the cost associated with misclassification of the majority class is substantially lower than a misclassification of the minority class, which makes a misclassification of the minority class more costly.

1 Introduction

1.1 Area of study

Predictive analytics using machine learning has become one of the most relevant technologies in recent years, quickly embraced for its capacity to consume vast amounts of data and extract non-linear relations that allow to forecast or predict the desired outcome and outperform other models. As a result, this approach is quickly becoming a mainstream technology that more and more businesses seek to harness and leverage to improve their decision-making.

1.2 Motivation for research

In many countries globally, but particularly in Brazil and Mexico, the private education industry has been undergoing a significant digital transformation, leading the industry to question and redesign its processes to adapt to the new realities, consumer preferences, and technologies and to enable and become scalable companies with considerable growth potential (Mahsood Shah, 2016).

This trend, undergoing over the last twenty years, has been particularly accelerated due to the COVID pandemic and a market undergoing consolidation into conglomerates of what once was a small fragmented market (Marinoni, G. et al., 2020 and Mahsood Shah, 2016).

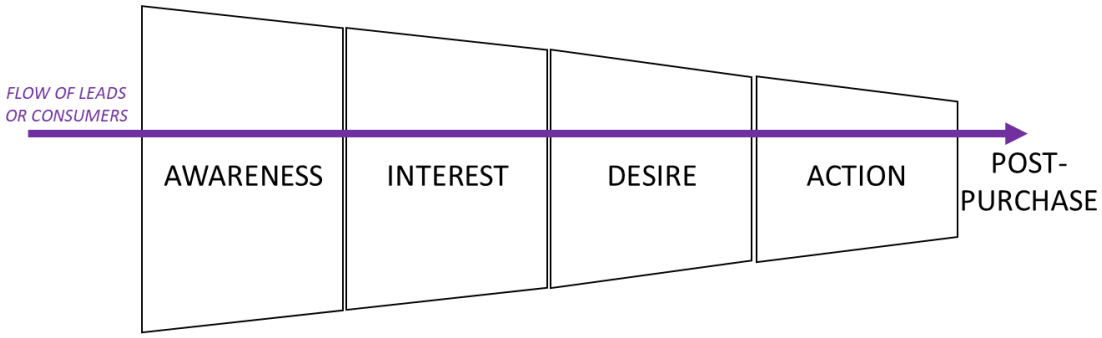
Scaling becomes an imperative, which forces them to transform their models into purely digital or hybrid education platforms to operate and sell their products. As a result, their commercial processes tend to be among the first parts of the company they need to digitize to ensure a scalable platform that allows them to grow their revenue and fund the capital-intensive digital transformation process (Mahsood Shah, 2016).

The main players in the market and best-performing companies have a similar structured approach to sales and commercial efforts (new student enrollment). Their commercial process is carefully structured and managed through Customer Relations Management (CRM) systems (Daradoumis, 2010; Guy-Emmanuel et al., 2016) and team efforts that process high volumes of leads and take them through an aggressive push-approach sales effort to sort and convince students to enroll in their universities.

This commercial approach is highly dependent on the input of leads and the efficiency in each process step to guarantee the planned conversion rates (Tanner, Ahearne, 2005).

The process is inspired by a consumer disposition model proposed initially by Elias St. Elmo Lewis, known as the AIDA (awareness/attention, interest, desire, action) model (Lewis, 1899, 1903).

Figure 1 The awareness/attention, interest, desire, action (AIDA) model which graphicly explains how a consumer moves from an unaware situation, through all the phases in the customer funnel and eventually ends up with an action.



The process is typically structured in the same manner across the most relevant players in the market; lead acquisition, lead contacting (awareness), lead convincing (intention), sales appointment (desire), and enrollment (action) (Michaelson, D., & Stacks, D. W. 2011; Vieira & Claro, 2020). The commercial actions in each step are carefully designed and managed to ensure a high conversion rate, measured as the percentage of leads that transfer from one step into the next. Therefore, a single percentage point of improvement in the conversion rates of earlier steps of the process can translate into substantial gains in later stages, directly impacting and improving the commercial performance of the companies (Vieira & Claro, 2020).

It is no surprise that the companies try to improve as much as possible in the initial steps, particularly in lead acquisition and initial contact. Companies then develop different lead acquisition strategies. E.g., direct prospecting, digital channel lead purchasing, referrals, etc. Previous years' leads, which have not yet enrolled, are also carried over to the current year if the leads are recognised as likely to enroll. These different sourcing strategies come with different costs, constraints, or implications;

some are more time and resource-intensive, some have essential time window constraints.

The acquisition of leads via digital sources is known in the industry as the "digital origin." The digital origin has been gaining traction and becoming one of the most relevant sources of lead volume, and overall sales as these leads tend to have higher conversion rates and faster conversion time from lead generation to sale (Constantinides, 2012). The way companies source their digital leads varies greatly, but most tend to rely on at least two of the leading players in the market: Facebook and Google (Vieira & Claro, 2020).

These players sell leads to companies via different services. Most depend on contests where competitors can bid on keywords to be advertised as interested consumers search for these. The price of each lead then becomes variable. It is then critical for companies to extract as much value from this origin and convert each lead quickly. (Dempster, C., & Lee, J. 2015).

In this industry, conversion rates from lead generation to sale tend to be in the low single-digits. Consequently (Lazarin & Urduain, April 2022), these education companies acquire large amounts of leads, ranging from hundreds of thousands to millions, to obtain their expected sales objectives. And rely on Customer Relationship Management systems (CRM) to process all this data, execute, and monitor their digital campaigns.

1.3 Research question - The business problem

Predicting a lead's probability of conversion could mean a substantial competitive advantage to companies in the industry, and particularly to our research subject. Understanding the probability of sale at the moment of lead generation allows them to develop their strategies and allocate resources to maximize the return on the investment. The key question at hand is then

Can we leverage machine learning to predict the conversion probability using the available data at the moment of lead generation?

If so, how does this compare with their current process, and what business efficiencies could this provide?

1.4 Value of research

Companies often spend substantial budgets on the earlier steps as they try to contact and follow up on each of the purchased leads. Their contact efforts are executed through means such as email, SMS, telephone, and WhatsApp. Even though messaging services are gaining relevance, the nature of the education purchase makes telephone and one-to-one conversation the preferred method of communication for the leads, and companies expect it will continue to be a highly important communication channel (Vieira & Claro, 2020).

Predicting which leads have the highest probability of conversion before the contact attempts would allow allocating special attention to accelerate the conversion to those leads, increasing efficiency in the process, which would free resources to be allocated to those other leads who may need more work. Additionally, it could help explain why and what paths and touchpoints in the lead creation increase the probability of conversion and develop commercial strategies to leverage those. Finally, being able to rate or grade a lead quality from this perspective may provide the company with tools to negotiate costs and raise the expected quality of each lead before even attempting to start contacting.

1.5 Literature review

Our problem at hand can be considered as a classification problem on imbalanced data with asymmetric cost, which is the topic within the machine learning community that is primarily explored. Somasundaram and Srinivasulu (2016) have investigated the process of classification on large and highly imbalanced data. They argue that the topic of imbalanced data is especially relevant for real-life data, as the probability for certain real time events tends to produce skewed data. This imbalance is often coherent with asymmetric cost, as the cost of misclassification is higher for the false negatives.

Within the literature focusing on the classification of imbalanced data with asymmetric cost, there is an extensive collection on the financial topics of credit default and fraud

prediction. These two topics are naturally imbalanced, as very few instances of the positive target variable (default or fraud) are represented in the data. However, the cost of not recognizing these instances soon enough is relatively high (Calabrese, 2014, p. 1). Calabrese (2014) proposes a method to determine the optimal cut-off for imbalanced data and shows that the cost-sensitive unweighted accuracy can outperform the iso-performance line method.

Another instance of default prediction is done by Zhou J. et al. (2019) in peer-to-peer (P2P) lending. They propose a decision tree model-based heterogeneous ensemble default prediction model to predict customers defaulting in the P2P lending platform accurately. Their model is compared with benchmark models to show that the model can achieve decent predictions on the high-dimensional and imbalanced data.

Furthermore, Ahmed Mohammed, R. et al. (2018) explores scalable machine learning techniques for highly imbalanced credit card fraud detection. Comparing several popular machine learning techniques, they observe that many detection algorithms perform well under medium-sized datasets while struggling to maintain the same level of precision when the dataset becomes massive.

Another area of study that shows similarities with the imbalanced data and asymmetric cost that we face is the prediction of click-through rates (CTR). The CTR is the binary classification of whether an online advertisement is clicked by the target or not and has gained a lot of attention lately (Zhang, S. et al., 2018, p. 1). Zhang et al. (2018) propose a combination of extreme learning machines to increase the performance of the binary classification problem compared to related algorithms. In another study, Gupta and Pal (2018) use a tree-based model to classify the successful and unsuccessful clicks, showing that a tree model can attain high accuracy.

Based on this research on different topics and industries, but with the same characteristics as our problem having highly imbalanced classes and asymmetric costs. We want to explore if machine learning algorithms can be used as an approach to predict a lead's probability of conversion before being contacted by the recruitment team, solely based on the meta data generated by a lead. Given the large operation and costs implied by treating every lead the same, correctly classifying the low probability

conversion leads from the high probability of conversion leads can result in significant cost reductions and effectiveness of the sales and recruitment team.

2 Data

In this chapter, we explain the data collection process of Universidad Insurgentes (UIN) CRM provider and briefly describe the raw data generated between 2019 to 2022, consisting of 41 columns and 549,754 rows. Then, we do a thorough exploratory analysis of the raw data before proceeding with data cleaning and pre-processing. The final dataset used in the empirical analysis includes 628 columns and 249,468 rows related to four different sales cycles.

2.1 Data collection

The data we use for our research is provided to us by UIN. The datasets contain the metadata of each lead UIN has had over the last two years and is extracted from HubSpot.

HubSpot is a commercial management software that serves as a CRM, digital marketing campaign manager, inbound and outbound sales, and customer service platform. This software tracks each lead's journey from a specified interaction which would trigger the tracking of a lead, now defined as a lead. E.g., an interaction that would initiate tracking of a lead could be the lead searching for 'higher education' on Google, within a relevant geographical location. From this point on, the data and interactions with the company is recorded as metadata related to the interactions with UIN, from that specific lead.

HubSpot defines metadata as "data that describes other data within a database or a data warehouse". Therefore, traditional data is the information of each lead, such as contact data, personal data, interests, and any item filled by the lead in any form from the company displayed in their digital platforms and campaigns. Metadata is the information that describes the behaviour or activity of each lead in terms of how said lead engaged with the digital platforms.

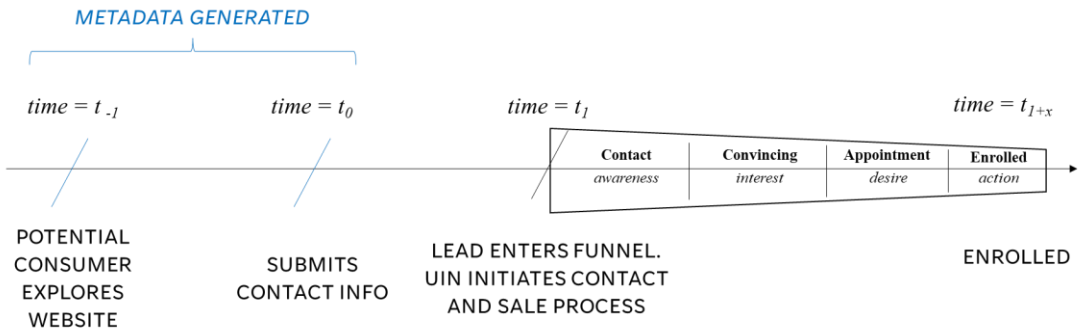
While "data" contains specific answers to direct questions that the lead answers, the "metadata" describe the lead's behaviour as the lead engages with the platform to answer those questions.

For example, if the lead visits a website that asks the lead which product, he is interested in knowing more about. The data would be the product that the lead selected. Metadata is the time it took for the lead to choose, the hour of the day at which the engagement happened, or the number of different pages the lead visited while the session was active.

The relevance of the metadata over data is that with very little data, we can collect far more metadata that describe the user's behaviour, potentially discovering trends or traits that better predict the leads tendency to enroll. For example, UIN collects only four data points for contacting purposes and initiating its sales cycle. Name, Email, Phone, Program of interest. While at the same time, UIN collects more than 50 metadata points by simply accessing its website.

This volume of data increases the probability that crucial informative value for predictive purposes may be discovered and, in turn, enables us to predict purchase before the contact phase begins.

Figure 2 Timeline of Metadata generation moments



HubSpot collects the metadata automatically as the aggregator of leads provided by different sources and origins. The origins of these leads are the initial sources that led each lead to get involved with the company. For example, some leads may reach through organic search, others may reach through paid advertisements, and others may be sourced through specialized companies that develop complex methods to identify

and acquire them. As a result, these different sources may impact the scope and breadth of available metadata.

2.2 Raw data description

The raw data is described in the data dictionary presented in Table 5. The dataset contains data types representing time, boolean, categorical and numerical, segmented into different categories describing the types of interaction that the lead has with the platform.

- *Active data sent:* number of forms filled in by the user
- *Call to actions responded:* Capturing the dates when the lead responded to a prompted action (e.g., "click here for…").
- *Lead's interaction with the company's websites:* This kind of data marks the moment in time when the lead met an important milestone in its digital engagement with the company. E.g., the date of first or last visit to a website, the time last visit session lasted, the first time the lead responded to a call for action in a website.
- *Email engagement:* This category reflects the interaction with the company's emails. The date the first marketing email was sent or the date the lead clicked an email marketing, the number of emails sent to the lead or similar.
- *Location:* Location of the IP used by the lead from continent level to city level.
- *Source of data:* origin or source through which the lead reached the website. E.g., a lead can reach through typing directly the URL, searching for a keyword in a search engine, social media, following a blog, or through paid advertisement.

Given the nature of metadata and the different sources of leads, some origins may not generate all the metadata fields. For example, suppose a lead was bought from a specialized source but has never actively been engaged in a website. In that case, the company will have contact information and metadata of contact attempts but no

metadata that presents the interaction of said lead on their website, as the lead has never been actively navigating the site.

This missing metadata (Appendix A) is presented as incomplete columns and will pose a challenge as we pre-process the data. However, missing data may have informative value for the algorithms; it is fair to assume that a lead who has been navigating the website and is familiar with the products offered may be more inclined to purchase than those who have no previous knowledge of the company and whose only knowledge of the company stems from being cold-called.

2.3 Exploratory analysis

The dataset used for training consisted of a merge of two datasets. The first dataset contained the metadata of approximately 0.55 million leads obtained between 2019 and 2022. This data, as described previously, represents the interaction of the lead with the company's CRM platform. The unprocessed dataset contained 41.3% missing values and 41 variables to each lead.

We use the Python (Python version 3.9. documentation available at https://docs.python.org/3/) library `pandas_profiling` (pandas-profiling version 3.2.0 documentation available at https://pypi.org/project/pandas-profiling/), and the function `profile_report()` to create a report of the raw data for explanatory data analysis (Appendix G). We could assess data completeness (Appendix H), distinct values, and histograms, and we were able to identify trends and patterns in data completion. In addition, the dendrograms (Appendix I) and heatmaps of missing value correlations allowed us to understand that missing URL values had information and should not be ignored or deleted.

A second dataset contained descriptive information about the sales process, whether the lead enrolled, if it was contacted, the different steps in the funnel each lead got through, and the sales cycle each lead got enrolled to. For the most part, this was a descriptive dataset, containing information related to a later stage of the sales funnel. Most of the variables in the second dataset posed the threat of target leakage, as the data being present for a specific lead, indicates that the lead is close to enrollment. For

this reason, only the class label of enrollment “yes/no” and the information to which sales cycle the lead related to were kept.

This second dataset contained ~250,000 leads from the sales periods of 2021 and 2022. This dataset became the reference for segmenting and selecting the leads and data to be considered in the training stage. By recommendation of UIN, given the covid pandemic, focusing on these two years would factor out the pandemic's effect on the company sales. The dataset was complete without missing values and contained mostly binary attributes with one categorical variable (enrollment cycle). The only relevant attributes considered from this dataset were the target feature “enrolled” and the cycle of interest of enrollment, which is known at the time of lead generation.

2.4 Data cleaning & pre-processing

The performance of a machine learning algorithm is only as good as the data used to train it; therefore, predictive modelling is mostly data preparation (Brownlee, 2020b, p. 13). Data quality is one of the most critical problems in any predictive modelling project since dirty data often leads to inaccurate results and bad business decisions (Ihab F. Ilyas, Xu Chu, 2019, p. xiii). We have spent a considerable amount of time ensuring that our raw data is prepared for our chosen algorithms. A detailed description of pre-processing operations made to the raw data is attached in the appendix, under Data Cleaning and Pre-processing (Appendix C). In short, the data cleaning operations relate to missing values and categorical variables with high cardinality. We also identified the need for some feature engineering, described in the appendix.

After cleaning and pre-processing, the data no longer have missing values, while maintaining the number of instances. The cleaned data does not have any high cardinality categorical variables, and the skew and kurtosis of the numerical features are reduced. All categorical variables are one-hot encoded, since this provides us with a dataset that consists of only numerical features which enables us to train most machine learning model on the same dataset.

No normalization or standardization preparations were made during the data cleaning and pre-processing. This is to ensure that we do not risk data leakage, as applying data

preparation to the entire dataset before splitting the data for training and test purposes increase the chance for data leakage (Brownlee, 2020b, p. 27).

The data cleaning and pre-processing resulted in a dataset that contains 56 columns and 249,468 rows, before one-hot encoding the categorical variables. The one-hot encoded dataset has the same number of rows, with 628 variables. The size of the final dataset is greatly reduced which lowers the hardware requirements needed to train a machine learning classifier.

3 Empirical analysis

The profit of UIN's current operation is compared with the maximized profit when predicting which leads are most likely to enroll using three machine learning classifiers and a simple logistic regression model. The data was first partitioned into a training and holdout dataset, where we used the cycles of enrollment to separate the training and testing dataset. We maintain the chronological order of the data and train all three classifiers on past data to evaluate the performance of the models on future data.

After training all classifiers and evaluating the models on the holdout data, LightGBM was selected as the best performing model when it comes to distinguishing the two classes according to the profit performance. The results obtained in this empirical analysis lay the ground for our more pragmatical recommendation and discussion in the next chapter.

3.1 Supervised learning

Our problem is a classification problem, where the output variable is the confidence of enrollment for each lead based on the available data related to the lead. This confidence representation is then converted into a binary variable {0,1}, based on the optimal threshold for when it is profitable to pursue the specific lead. This problem falls under the supervised learning category. We have input variables X and an output variable Y, and we use an algorithm to learn the mapping function from the input to the output.

$$f(X, \varepsilon)$$

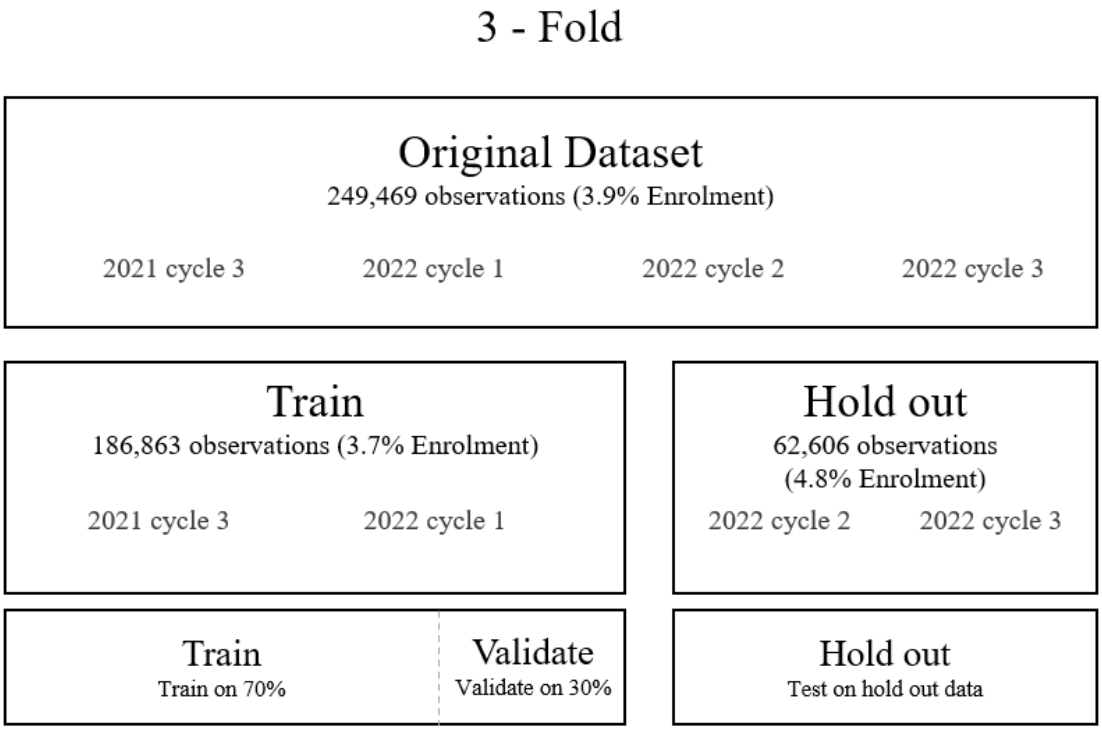
The goal is to approximate this mapping function so that new input data (X) can predict the unseen output variables. It is called supervised learning because the output variable

of the data used to train the model is known, while the goal is to generalize the model to correctly predict the output variable when running the model on input variables where the output variable is not known (Brownlee, 2016a, p. 16).

3.2 Data partitioning

The data that UIN receives can be assigned to a specific sales cycle, as UIN operates with three cycles per year. Before every cycle begins, UIN receives data on potential enrolling leads, which UIN then contacts in an effort to enroll the lead in one of UIN's programs. This creates a time hierarchy, e.g., "2021 cycle 3" comes before "2022 cycle 1". We want to use the knowledge of which leads that enrolled in a previous cycle to predict the leads most likely to convert into an enrolling lead in the current cycle.

Figure 3 Data partitioning strategy based on the time chronology of cycles.



Based on the chronological order of the data and UIN's business model, the chosen method to minimize the risk of overfitting the model is a 3-fold partitioning strategy. The full dataset is split into a training dataset consisting of the first two sales cycles, and a holdout dataset consisting of the last two sales cycles. The model is trained on 70% of the training data and validated on the remaining 30% of the training data, now named the validation data. A final hold-out dataset is then used to test the models. This

strategy is considered to be the favourable, as it is generalizable to new cycles. As the dataset of UIN expands into more cycles, the dataset used for training and validation can grow as long as the data related to past cycles stay relevant when predicting the outcome of new data.

Our choice in a 3-fold cross-validation strategy relates to the size of each cycle. The number of observations within each cycle varies and splitting the training and validation data based on cycles would provide us with an unfavourable ratio of data to train, validate and test. Overall, the chosen data partitioning strategy gives us 130,804 observations in the training data with 3.6% enrollment, 56,059 observations with 3.6% enrollment in the validation set, and 62,606 observations in the holdout data with 4.8% enrollment.

3.3 Metrics for measuring model performance

The literature on model evaluation metrics consists of multiple performance metrics used to evaluate and quantify the performance of a classifier. Because a classifier is only as good as the metric used to evaluate it, choosing the correct metric for the task at hand is crucial. This is particularly important for imbalanced classification problems. Some evaluation metrics are optimized by a high ratio of correct predictions and will therefore only predict the majority class (Brownlee, 2020a p. 39). Below we go through our chosen performance metrics that we will use to evaluate the classifiers on the imbalanced dataset.

3.3.1 Confusion matrix

A confusion matrix separates the classification made by the classifier, distinguishing how one class is being confused for another. This way, we separate the different errors, making it easy to deal with them separately (Provost & Fawcet, 2013, p. 189).

Figure 4 Confusion Matrix

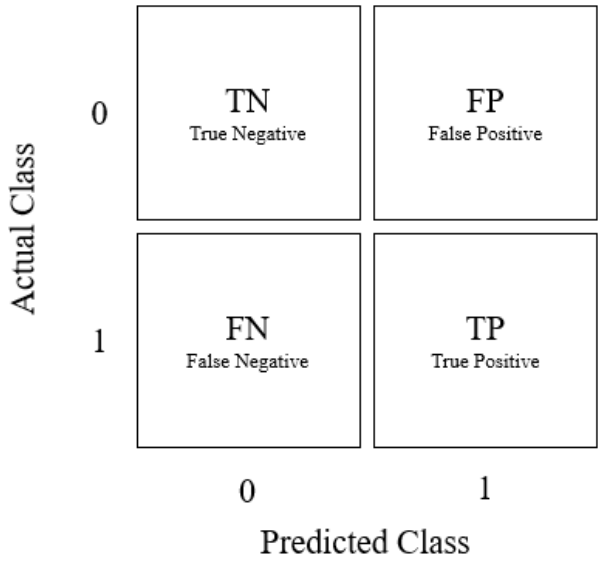


Figure 4 displays how the binary classification problem is divided into four instances that can be used to calculate a variety of performance metrics. The True Negative (TN) represents all leads that have not enrolled as students and are correctly classified as not enrolled by the model. The False Negative (FN) are the leads that have enrolled as students but were classified as not enrolled. The False Positive (FP) are the leads falsely classified as enrolling, and the True Positive (TP) are the leads correctly classified as enrolled. Performance metrics are summaries of the confusion matrix, referring to the counts in the matrix (Provost & Fawcett, 2013, p. 203). For this project, we want to optimize the classifier to have a high ratio of TP because of the asymmetric cost related to misclassifying the minority class, which we will explain in the next section.

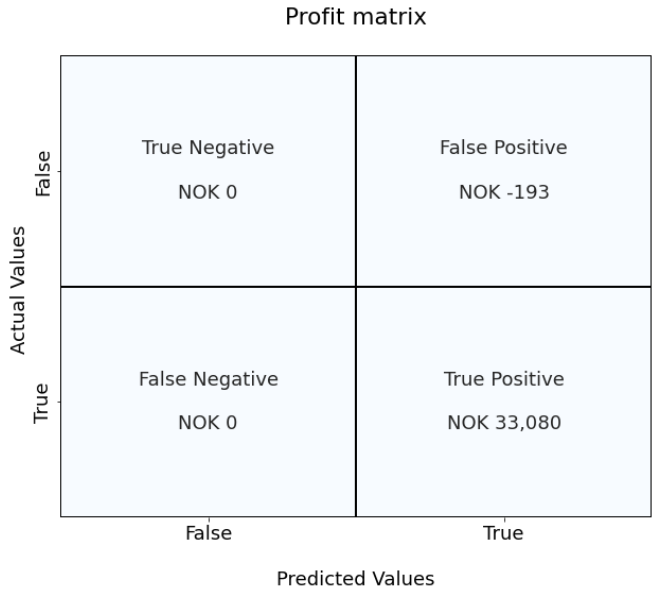
3.3.2 Profit matrix

To compare models, we need to attribute a cost and revenue to each decision depending on the confusion matrix quadrant. We have assumed a theoretical action (treatment to lead conditional on predicted values) and later recommend a practical action based on a realistic approach to current industry practices. Throughout the thesis, we assume that a lead not contacted by UIN's sales team will not enroll. Therefore, the probability of enrolling for a lead that has not been contacted will always be 0.

Table 1 Costs and revenue per lead. The variable cost elements are made into an average of the total variable cost, not to reveal the cost ratio each cost item contributes with. For our research, this has no effect on the results as we only use the variable cost per lead.

Variable costs per lead		
Talent	NOK	38.40
Telco	NOK	38.40
Software	NOK	38.40
Marketing	NOK	38.40
Hardware	NOK	38.40
Total	NOK	192.63
Revenue per enrolled lead		
Total	NOK	33,273

Figure 5 The profit matrix used to evaluate the performance of the different approaches throughout our research.



3.3.3 Threshold selection

Our goal is to maximize the profit, where profit is defined as the difference between revenue and cost. The maximum profit is obtained by the output level q, where the difference between revenue and cost is greatest (Pindyck & Rubinfeld, 2017, p. 295).

Revenue and cost are determined by how many leads that are classified as either true or false.

The classification models will predict a probability-like score of the model's confidence in the lead to enroll in one of UIN's programs. This probability-like score will then be converted into the binary class label, based on a determined threshold for when it is profitable to pursue engagement with a lead based on the probability-like score. We therefore need to identify which threshold to use when converting the classifiers probability-like score into a class label.

Changing the classification threshold is called threshold-moving and is a common strategy for problems with severe class imbalance (Brownlee, 2020b p. 245). Since we know the cost associated with a contacted lead and the profit associated with an enrolling lead, we can calculate the profit-maximizing threshold, which we will name the theoretical threshold. The theoretical threshold will be used to define the class label of a lead. E.g., we assume the optimal threshold for profit-maximization is *t*. If our classifier gives a lead a probability $p < t$, the lead will receive the class label of 0, which indicates a negative profit expectation.

For each lead, we want to make the decision that maximizes profit in expectation. The decision is made based on the predicted confidence for each lead to enroll, where a profit maximizing threshold is determining the decision. We find this profit maximizing threshold based on the profit and costs associated with each decision. Let X be a binary random variable with a success probability of enrollment $p = P(X = 1)$. Considering a bet with payoff π if $X = 1$ and $-c$ otherwise. The expected value of the bet is, therefore

$$E\left[\pi 1[X = 1] - c1[X = 0]\right] = \pi p - c(1 - p)$$

We should take the bet if the expected value is greater than 0, i.e., $\pi p - c(1 - p) > 0$. This gives us the theoretical threshold, which will yield the highest profit.

$$p > \frac{c}{(\pi + c)}$$

Since we have the cost and profit associated with the different class labels, explained in more detail in chapter *3.3.2 Profit Matrix*, we can calculate the theoretically optimal threshold:

$$p = \frac{192.63}{33{,}080.37 + 192.63} = 0.0058$$

Implying that any lead with a probability score greater or equal to 0.0058 will have an expected positive profit and therefore receive the class label of 1. This theoretically optimal threshold assumes that the classifiers predictions, reflect the true conditional probability. That is, the predicted class probability needs to be well-calibrated. To be well-calibrated, the probabilities must effectively reflect the true likelihood of the event of interest (Kuhn & Johnson, 2013, p. 249).

3.3.4 ROC AUC

We will use an evaluation method that enables us to visualize the performance of a classifier over the entire operating range and all possible imbalance ratios. The area under the Receiver Operating Characteristic Curve (ROC AUC) is one evaluation metric concerned with how effective the classifier is at separating classes.

The ROC curve is a diagnostic plot for summarizing the behaviour of a classifier by looking at the relationship between the false positive rate and true positive rate under different thresholds (Brownlee, 2020a, p. 41). The true positive rate measures the number of correct positive predictions made from all correct positive predictions that the classifier could have obtained. In this way, the true positive rate provides coverage of the positive class and is often used as a performance measure of imbalanced learning (Brownlee, 2020a, p. 61). Conversely, the false-positive rate shows the number of FP divided by all negatives present in the training data.

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

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

Plotting the fraction of correct predictions for the positive class on the y-axis against the fraction of errors in the negative class on the x-axis gives us the ROC curve. This

curve can be understood as the relationship between the two classes, where we can improve one of the ratios at the expense of the other and vice versa. Figure 26 in the appendix gives a visual representation of how to interpret the ROC curve.

Comparing three or more models against each other based on their curves can become a challenge. So instead, the Area Under Curve (AUC) is used to give a single score for the classifier comparable across models. This approach gives us the ROC AUC metric, a value between 0 and 1, where 1 indicates a perfect classifier (Brownlee, 2020a, p. 72). We will use the ROC AUC metric as a supplementary measurement when comparing multiple classifiers.

3.4 Model Calibration

The machine learning classifiers we will use predict a probability-like score for class memberships. We want to interpret these probability-like scores as the true conditional probabilities, which requires that the model under evaluation is well-calibrated.

To validate the assumption that our models are well-calibrated, we will evaluate each of our selected machine learning algorithms under the theoretical optimal threshold against the empirical (calculated) optimal threshold. If the machine learning models are well-calibrated, the theoretical optimal threshold should be close or equal to the empirical optimal threshold.

As mentioned in chapter *3.3.3, Threshold Selection,* our theoretical optimal threshold should match the calculated optimal threshold when evaluating the model's performance. If this holds, and the two thresholds are similar, we can assume that the performance of the model represents the true conditional probabilities and give the predicted probability score an interpretation. However, suppose these two thresholds do not match within a reasonable interval. In that case, we must assume that the model is not well enough calibrated for the predicted probabilities to receive a realistic interpretation.

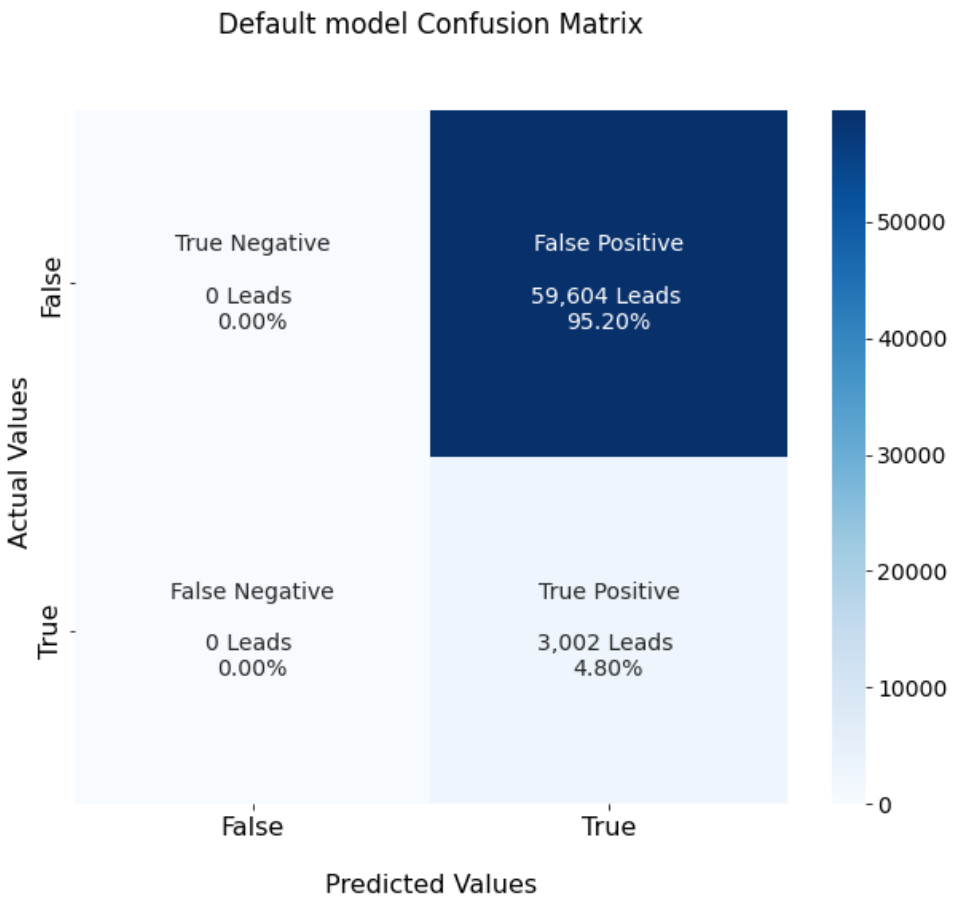
3.5 Default model

The default model is the *status quo* of the commercial operation at UIN. Under the default model, UIN assumes every single lead has the same conversion probability and predicts $y = 1$ for every lead. Every lead is integrated into the commercial process and

receives the same treatment; it enters the conversion attempt process and receives a series of phone calls, WhatsApp messages, and emails to attempt contact.

The confusion matrix for the hold out data, which will be used to measure performance throughout our research, is as follows:

Figure 6 Default Model Confusion Matrix.



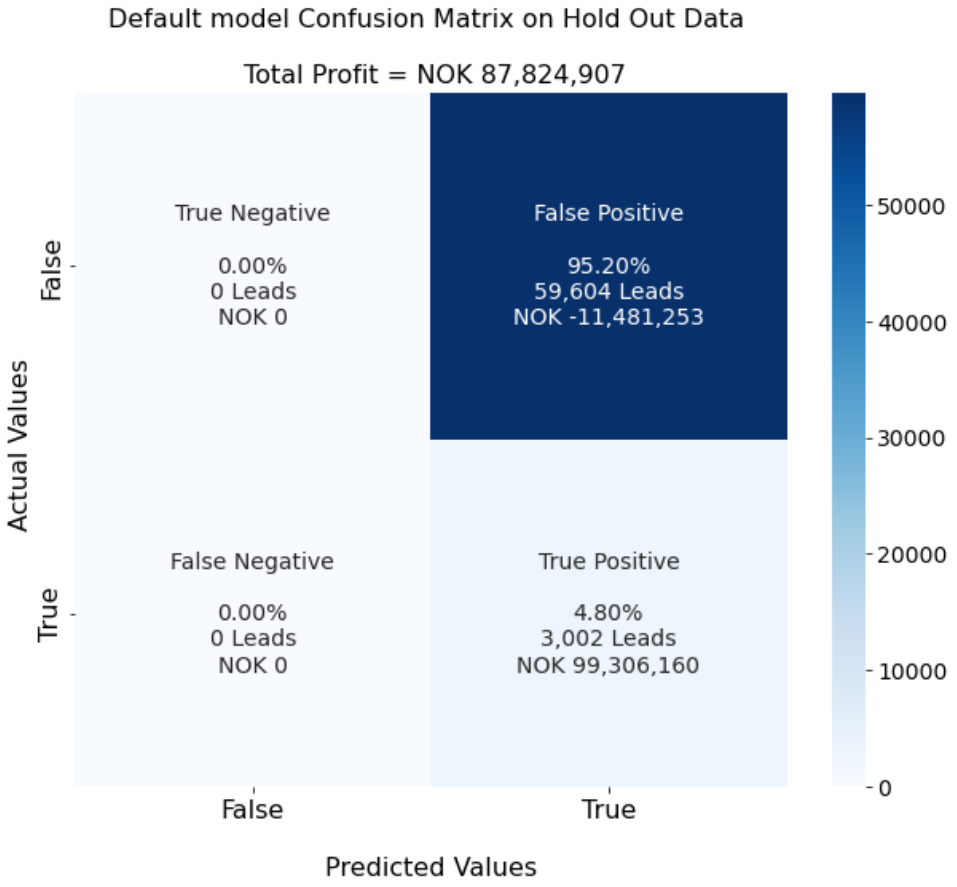
The key takeaway from the confusion matrix with the default model is that all leads are classified to enroll, and pursued by the commercial process.

3.5.1 Profit of the default model

The profit of the default model is defined by adding the count values of each of the default model's confusion matrix cells weighted by the cost or revenue of its corresponding cost-matrix cell.

This profit is estimated for the hold-out data since this will be used to evaluate the performance of the logistic regression and machine learning models.

Figure 7 Default model confusion matrix on hold out data with profit calculation.



3.6 Selection of algorithms to train

When deciding on which machine learning algorithms should be considered for training and evaluation, we had to consider that our dataset consists of both numerical and categorical variables. This led us to proceed with decision tree algorithms as they have many beneficial properties for our dataset. Decision trees are simple to understand, interpret and visualize, which is essential when applying the finalized algorithm to the business problem we are trying to solve. Furthermore, the tree algorithms do not get affected by nonlinear relationships in the data, which is another crucial strength. Another favourable trait of decision trees is that they require less data

preparation relative to other machine learning algorithms and implicitly perform variable screening and feature selection (Gupta, P. 2017).

Decision trees provide the foundation for more advanced methods such as boosting (Brownlee 2016, p. 91). Boosting algorithms seek to improve the prediction power by training multiple weak models, where each consecutive weak learner compensates for the weaknesses of its predecessors. This is a particularly good feature when the data is imbalanced, as the iterative process of focusing on its predecessors' weaknesses and assigning different weights to the trees, depending on how well its performing. Boosting is, therefore, not a specific model but rather a generic algorithm that can be utilized by multiple machine learning algorithms (Zhang, Z 2019). Based on these reflections, we have chosen the three algorithms LightBMG, XGBoost, and CatBoost to evaluate for answering the research question. Additionally, we train a simple logistic regression model as a baseline model, to give us a perspective of how well performing the boosting algorithms perform.

In the following sections, we evaluate the classifiers based on the theoretical optimal threshold and compare it with the calculated optimal threshold based on predictions made on the hold-out data. Finally, we compare the results from all four classifiers to identify which classifier we suggest as the best one.

3.6.1 Logistic Regression

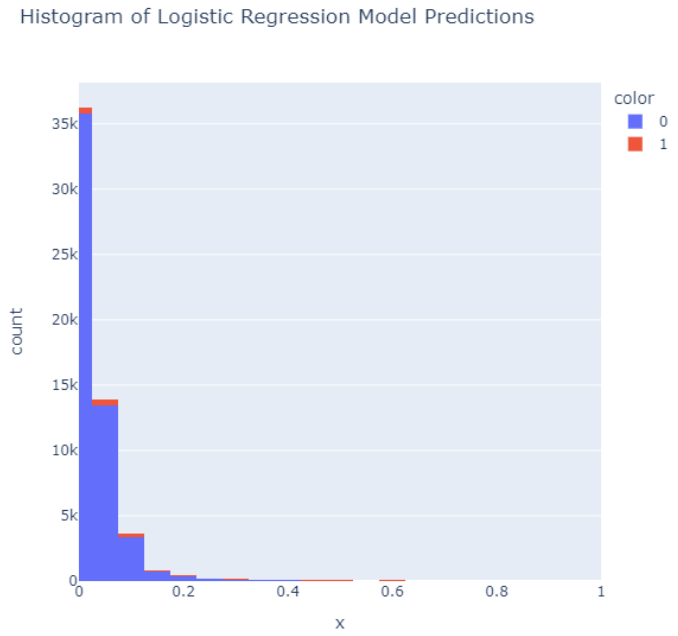
In addition to the default model, we compare the machine learning models against a logistic regression model to evaluate how well the machine learning models perform. The logistic regression model is a simple machine learning model that estimates $P(Y_i = 1|X_i)$. Where X_i is a vector with $X_{i,j}$ as elements

$$P(Y_i = 1|X_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_{i,1} + \ldots + \beta_t X_{i,t})}}$$

The model predicts each leads class probability and returns a value in (0,1) which is the estimated probability of enrollment for each lead (Provost & Fawcett, 2013, p. 96). We use `sklearn` (Pedregosa *et al.*, 2011) and the `LogisticRegression` function with the default parameters, when we train the logistic regression model (Sklearn.Linear_model.LogisticRegression – Scikit-Learn 1.1.1 documentation).

A histogram of the predicted probabilities on the validation data with the logistic regression classifier, reveals how the model's predicted probabilities are mostly distributed with values < 0.1. We know from the theoretical threshold of 0.0058, that it is the predicted probabilities in the lower range of the probability distribution that are most crucial for the profit estimate of the model.

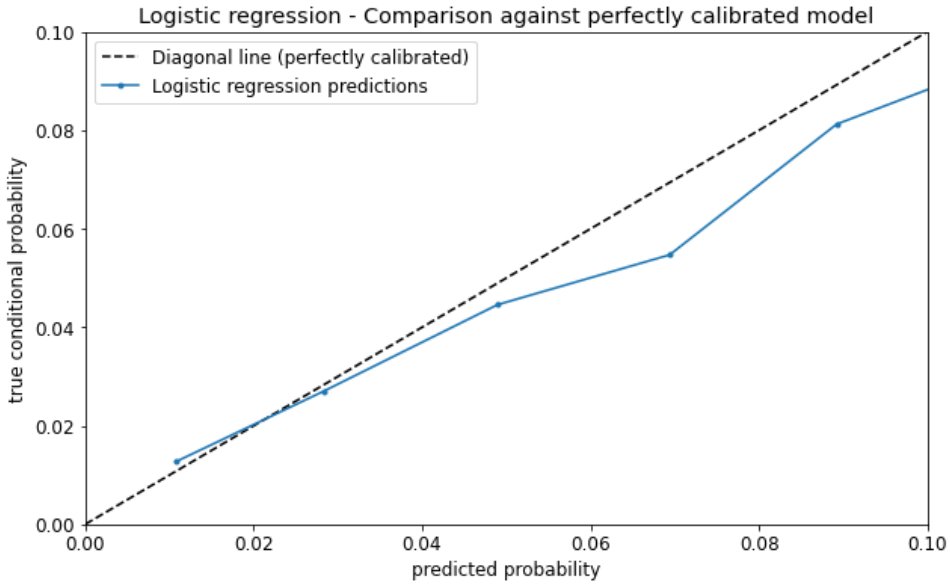
Figure 8 Histogram of the predicted probabilities on the validation data, using the logistic regression model.



To validate the relationship between the predictions made by the classifier and the true posterior probabilities, we use the Python library `sklearn` and the `calibration_curve` function to plot the predicted probabilities against the perfectly calibrated black dotted line as a reference (Niculescu-Mizil & Caruana, 2005). The x-axis of Figure 9 represent the predicted probability, while the y-axis represents the true conditional probability. We do this in the prediction interval of 0-0.1, as this is the range, we have shown that the logistic regression classifies most of the leads. The method discretizes the [0, 1] interval into bins, so examining whether the logistic regression model is well-calibrated for predicted values >0.1 provides almost pure noise, since the number of observations in this range is vanishingly small.

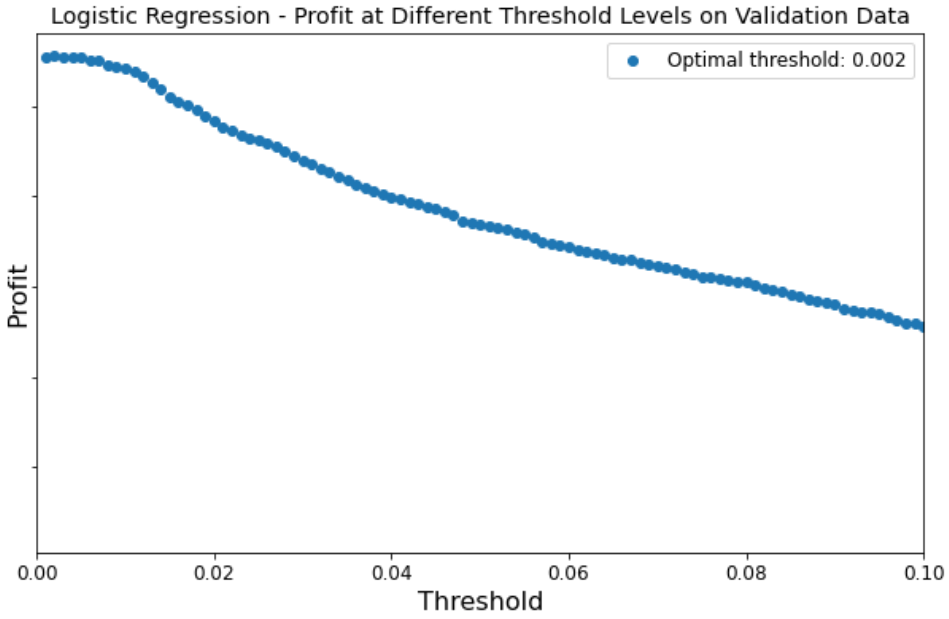
For the logistic regression model, we observe that the predicted probabilities deviate from the perfectly calibrated line. This observation indications that the predicted probabilities from the logistic regression model can't be interpret as the true conditional probability.

Figure 9 Plot of the probability curve with the logistic regression model on the validation data. The dotted diagonal line represents a perfectly calibrated model.



Using the trained logistic regression model, we calculate the profit obtained over all different classification thresholds used to predict the class label on the validation data. We then find the threshold at which the model provides the largest profit, which is nicely visualized in Figure 10. The profit maximizing threshold obtained for the logistic regression is 0.002, which will be used to evaluate the performance of the model on the hold out data. The threshold of 0.002, represents the empirical optimal threshold, mentioned in 3.4 Model Calibration, for the logistic regression model.

Figure 10 Obtained profit over different threshold levels obtained with the logistic regression model with predictions on the validation data.



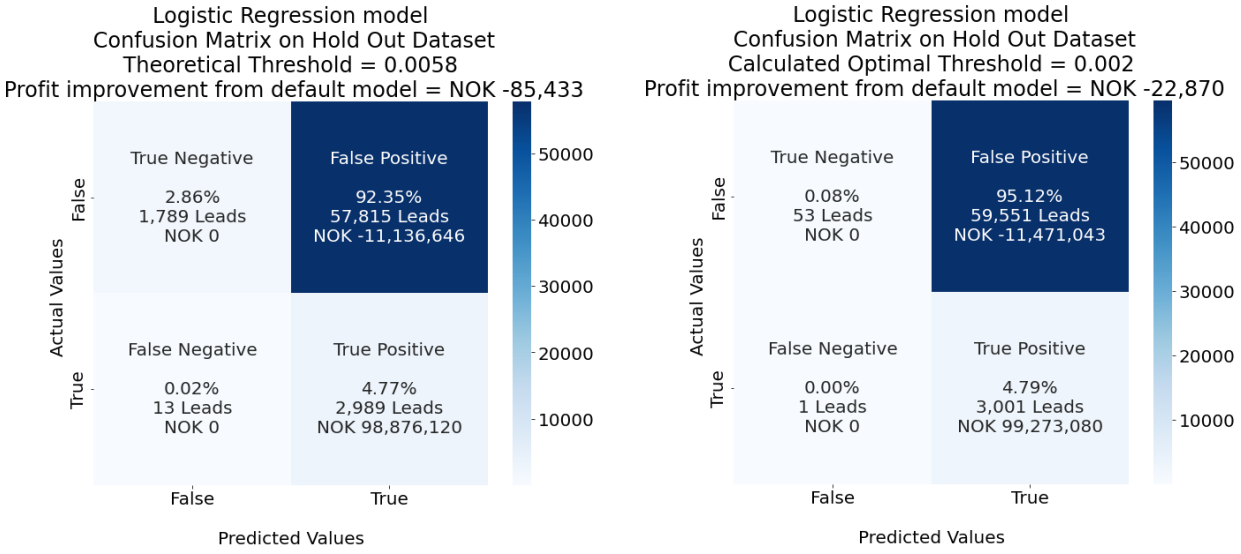
After calculating the empirical optimal threshold, we apply the theoretical- and calculated thresholds to the logistic regression model and make predictions on the hold-out data. As we can see from the confusion matrix, the logistic regression model performs worse than the default model for both the theoretical and empirical threshold. The logistic regression model yields a negative profit in both cases, compared to the default model.

In chapter 3.4 Model Calibration, we mentioned that for a well calibrated model, the theoretical and empirical thresholds should match. For the logistic regression model, we argue that the thresholds are not sufficiently similar to assume that the predictions of the model can be interpreted as the true conditional probability. Based on this conclusion, we will not go any further into examining the predicted probabilities of the logistic regression model.

The potential monetary gain from the model is determined by the difference between the profit obtained from the default model and the logistic regression model. The potential monetary gain is calculated by subtracting the missed income from the avoided loss.

$$Avoided\ Loss\ -\ Missed\ Income\ =\ Gain$$

Figure 11 The profit matrix for the hold-out data, using the logistic regression classifier with theoretical threshold and optimal threshold.



3.6.2 LightGBM

Light Gradient Boosting Machine (LightGBM) is a gradient boosting framework for machine learning developed by Microsoft in 2016. This algorithm tackles the challenge of increasing computational complexity with the number of features and data points with two techniques: Gradient-Based One-Sided Sampling and Exclusive Feature Bundling. With this, LightGBM reduces the training speed, lowers memory usage, provides better accuracy, and scales better on large datasets than conventional gradient boosting decision trees (Ekanayake, N. 2021, September 17). We use `lightgbm` and the `LGBMClassifier` function with the default parameters, when we train the model (lightgbm.LGBMClassifier – LightGBM 3.3.2.99 documentation).

We follow the same logic we used to evaluate the performance of the logistic regression model, on the LightGBM and the other machine learning models. The histogram of the predicted probabilities on the validation data with the LightGBM classifier (Figure 17), shows that the model's predicted probabilities are mostly distributed with values < 0.1.

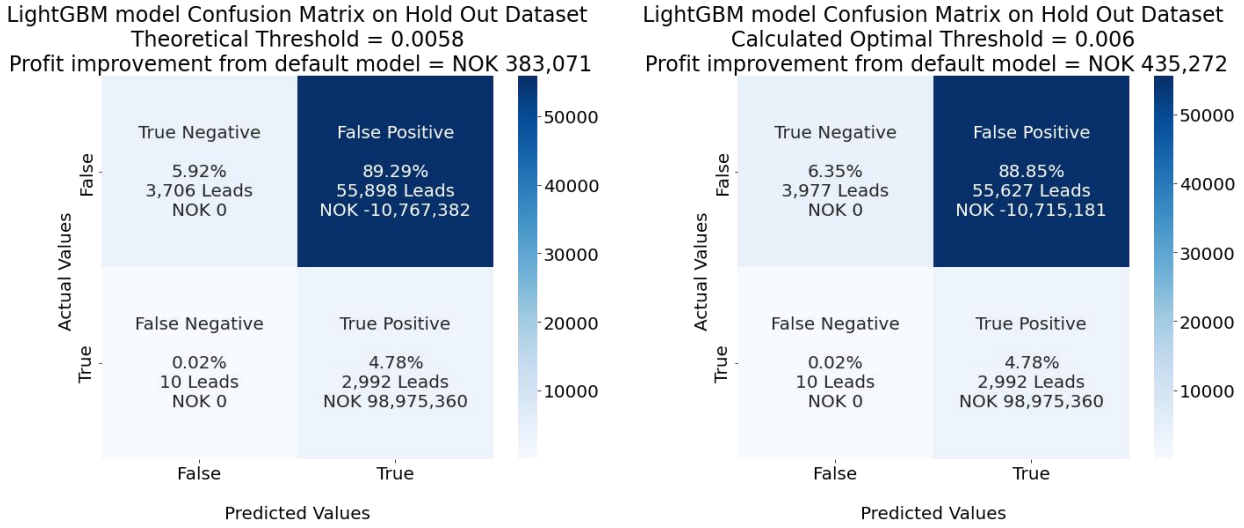
Plotting the probability curve (Figure 20) in the same chart, with the perfectly calibrated black dotted line as a reference, shows that the LightGBM classifier is well calibrated in the identified critical probability range. Especially in the lower range of 0 – 0.1, we see that the calibrated probabilities fit the perfectly calibrated line well. This indicates that the predicted probabilities, in the identified range, can be given an interpretation as if it's true conditional probability.

With the LightGBM model, we calculate the optimal empirical threshold with classifications on the validation data (Figure 23). The optimal threshold is found at 0.006. We argue that its close enough to the theoretical threshold for us to conclude that the LightGBM model is well calibrated in the range from 0-0.1. We can therefore give the models predicted probabilities an interpretation as the true conditional probabilities.

The optimal empirical and theoretical thresholds are used to evaluate the performance of the LightGBM classifier. Both thresholds perform better than the default model, when evaluated on the hold out data. Based on the conclusion that the LightGBM model is well calibrated, we can work with the theoretical threshold when evaluating the LightGBM classifier.

Using the LightGBM classifier provides profit improvements in the range of NOK 383,071 – 435,272 on the hold out data, compared to the default model.

Figure 12 The profit matrix for the hold-out data, using the LightGBM classifier with theoretical threshold and optimal threshold.



3.6.3 XGBoost

eXtreme Gradient Boosting (XGBoost) is a scalable, portable, distributed gradient boosting algorithm that provides fast and accurate parallel tree boosting. One of the unique features of XGBoost is how it handles sparsity in the data, which is common in most large datasets, ours included. Furthermore, XGBoost is built to avoid building complex trees that can cause the model to memorize patterns instead of learning, which will be necessary for this project as we aim to train a model on past data to predict the outcome of future unseen data (Ekanayake, N. 2021, September 17). We use `xgboost` and the `XGBClassifier` function with the default parameters, when we train the model (xgboost.XGBClassifier – XGBoost 1.6.1 documentation).

The histogram of the XGBoost model predicted probabilities (Figure 18), shows that most of the leads receive a probability score <0.1, similar to the other models.

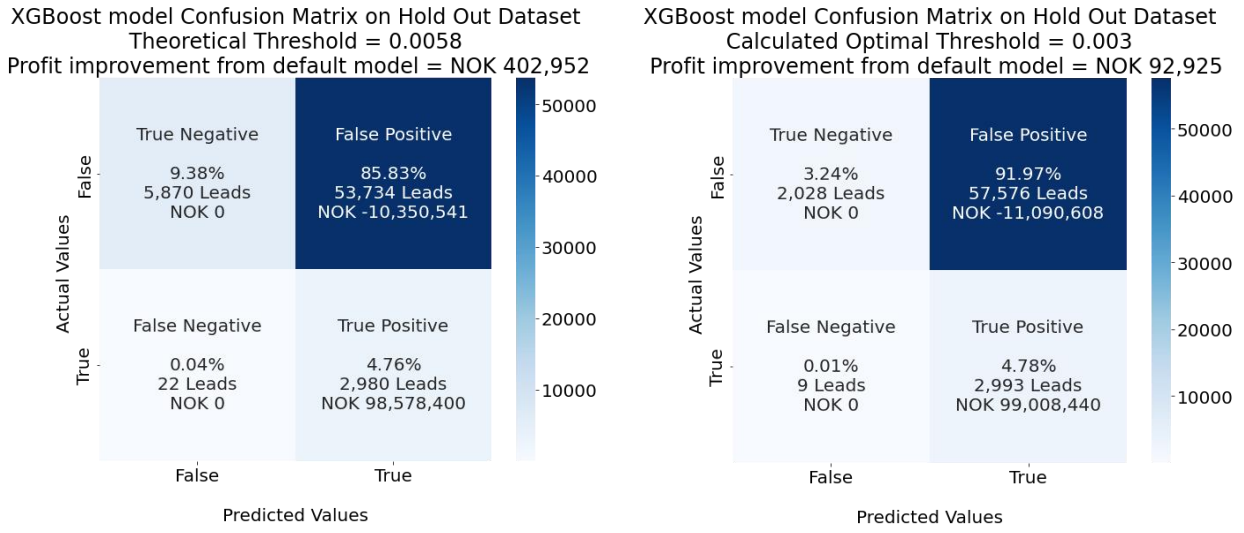
A plot of the probability curve for the XGBoost classifier (Figure 21), on the validation data, reveals larger deviations from the perfectly calibrated line, compared to the LightGBM model. It indicates that the XGBoost model is not well calibrated.

We calculate the profit obtained over all different thresholds used to predict the class label on the validation data. Plotting the obtained profit over the range of thresholds reveals a maximum value at the threshold of 0.003 (Figure 24Figure 21). The deviation

between the optimal empirical and theoretical threshold for the XGBoost classifier, leads us to the conclusion that the model is not well calibrated. Therefore, we will not give XGBoost's predicted probabilities any further interpretations.

The trained XGBoost model is evaluated on the holdout data, using both the optimal empirical and theoretical threshold. Both thresholds yield a positive profit when compared to the default model. The results obtained by the XGBoost model differentiates from the other classifiers, as it is the theoretical threshold that delivers the largest gain in profit. Because of the difference in the optimal empirical and theoretical threshold, we evaluate the XGBoost model not to be well calibrated. Therefore, we will not go any further into the examination of the predicted probabilities of the XGBoost model.

Figure 13 The profit matrix for the hold-out data, using the calibrated XGBoost classifier with theoretical threshold and optimal threshold.



3.6.4 CatBoost

The last algorithm we have chosen to evaluate for our problem is the CatBoost algorithm. It was developed by Yandex in 2017 and is based on gradient boosting. CatBoost provides a scalable, fast, and open-source algorithm. We use `catboost` and the `CatBoostClassifier` function with the default parameters, when we train the model (catboost.CatBoostClassifier – CatBoost documentation).

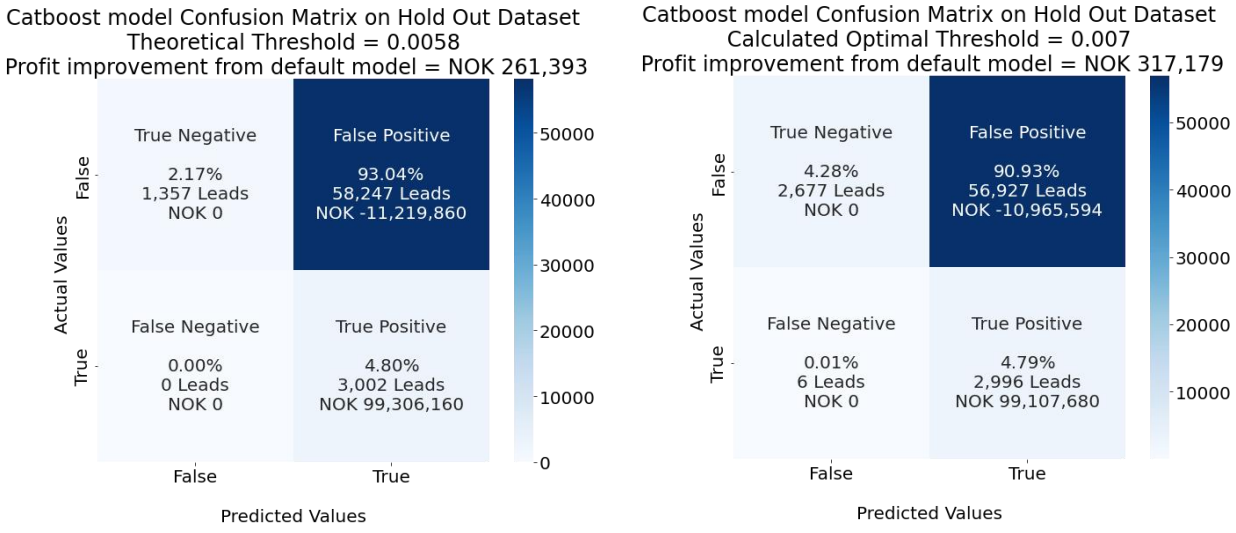
Similar results are obtained with the predicted probabilities of CatBoost, ass the other models. The histogram shows that most of the leads are predicted a probability score <0.1 (Figure 19).

CatBoost's predictions deviates slightly from the diagonal line, in for predicted values >0.05 (Figure 22Figure 22). In the lowest range <0.5, the model fits the diagonal line well. This indicates that the predicted probabilities of the CatBoost classifier is well calibrated in the lower range of the probability distribution, which we will further evaluate with the threshold selection.

The profit maximizing threshold obtained from the validation data, when using the trained CatBoost classifier is located at 0.007 (Figure 25Figure 22). The optimal empirical threshold is close to the theoretical threshold, but we argue it is not similar enough to conclude that the CatBoost classifier is well calibrated. We would therefore warrant the use of the optimal empirical threshold when working with the CatBoost classifier, with the disadvantage that we can't interpret the predicted probabilities as if they were the true conditional probabilities.

Both thresholds are applied to the model and make predictions on the hold out data. Both thresholds improve the profit from the default model, where the empirical threshold is the one that yields the highest expected profit.

Figure 14 The profit matrix for the hold-out data, using the calibrated CatBoost classifier with theoretical threshold and optimal threshold.



3.6.5 Model comparison & reflections

All results obtained from the algorithms are displayed in Table 2. These results are all obtained from classifications made on the hold-out data and compared with the default process profit performance. We can see from the results that the best performing model is LightGBM, which obtains the highest expected profit improvement when using the empirical optimal threshold.

The ROC AUC score, which we use as a supplementary measure, demonstrate that for our problem it does not coincide with the model that produces the highest expected profit improvement. Therefore, the ROC AUC score will not be part of the decision to which model is the best performer in our context.

Table 2 Model performance metrics used to evaluate the three classifiers.

		Threshold	*Profit improvement*	*ROC AUC*
Logistic	Theoretical	0.0058	-85,433	0.7894
Regression	Calculated	0.002	-22,870	
CatBoost	Theoretical	0.0058	261,393	0.8164
	Calculated	0.007	317,179	
XGBoost	Theoretical	0.0058	402,952	0.8097
	Calculated	0.003	92,925	
LightGBM	Theoretical	0.0058	383,071	0.8128
	Calculated	0.006	435,272	
Default			0	

All three boosting models manage to beat the default models profit performance, whereas the logistic regression performed worse than the default process. Based on this observation, we argue that the simple logistic regression model is not able to capture predictive features in the current data, which would enable it to more precisely predict the two classes. The more advanced boosting algorithms, on the other hand are all able to outperform the default model.

Only one of the three boosting algorithms obtain a calculated optimal threshold that is reasonable equal to the theoretical threshold for us to argue that the predicted

probabilities can be given a true conditional probability interpretation. This classifier, LightGBM, is both well calibrated and obtains the highest expected profit improvement overall. It is also the model that experiences the smallest deviation in expected profit from the three boosting models, and it has the highest average expected profit considering both threshold levels.

All three models manage to beat the default models profit performance, so we consider them all to be evaluated for operational use. That said, the best performing model in the current context is LightGBM, which we will examine further and consider in chapter 4 Discussion & recommendations.

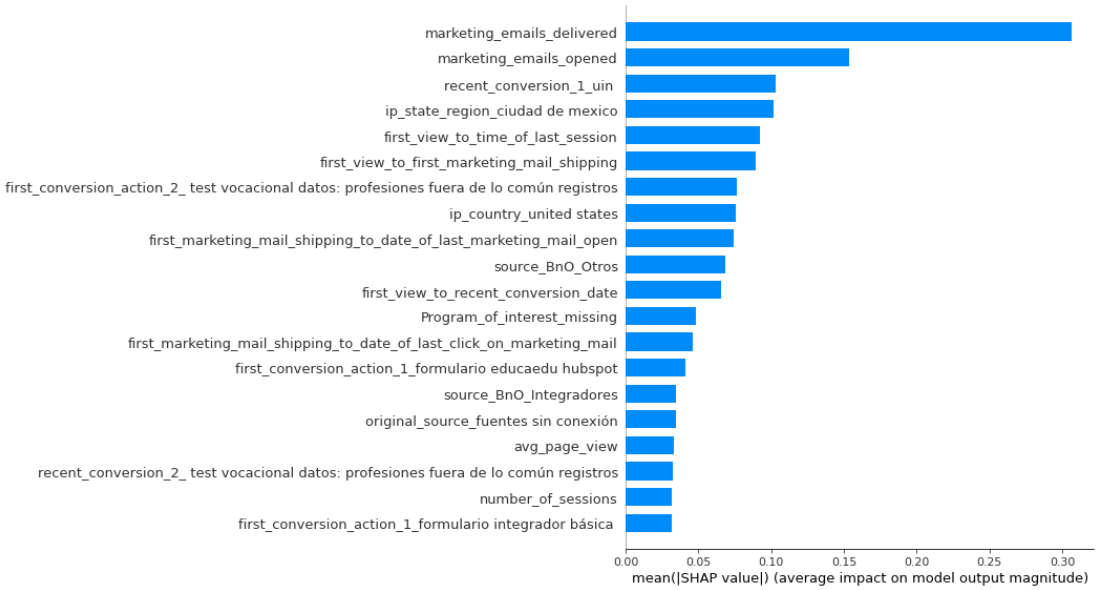
3.7 Detailed examination of LightGBM

This chapter further examines the results obtained with our chosen classifier, LightGBM. It obtained the largest gain in profit from the default process of the three boosting models in evaluation. We provide insightful information about which variables are most important when correctly predicting the class label and lay the grounds for our recommendations and conclusion.

3.7.1 Variable importance

The variables of the trained LightGBM model are evaluated using the Shapley values. Shapley additive explanations values (SHAP) are derived from the importance of each feature to the overall predictive result (Lundberg & Lee, 2017). The top 20 most important variables for predicting the minority class can be seen from the graph below, while the importance of the top 50 features can be found in the appendix. In Table 5 in the appendix, a table with descriprion of the the raw data features is included.

Figure 15 Top 20 variables for the overall predictive result of the minority class.



The features are ordered by how much they influence the model's predictions overall, where the absolute SHAP value is on the x-axis. This means that for any lead in our dataset, the importance of each feature can vary. For example, the model might evaluate the feature 'avg_page_view' as the most important feature when predicting the outcome for lead *i*. In contrast, the model can evaluate the same feature as not particularly important when predicting the outcome for a different lead *j*. It is, therefore, difficult to give an unambiguous interpretation of the significance of the features that can be extrapolated across all leads in the dataset.

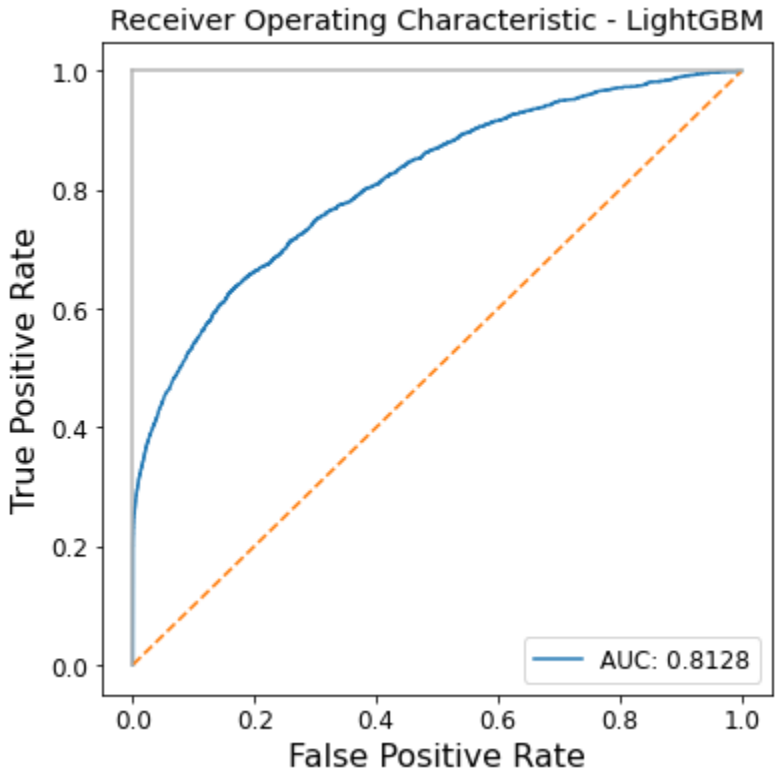
A prediction can be explained by assuming that each feature value of the instance is a 'player' in a game where the prediction is the payout. The Shapley value is the average marginal contribution of a feature value across all possible coalitions (Molnar, C. 2022, p. 215-216). This makes it possible to explain what each feature value contributes to the predicted probability of each individual lead. This is not a topic we will discuss further, as it becomes too granular for our thesis but can be used to gain further knowledge of the most important features on an individual level.

3.7.2 LightGBM ROC AUC

The Area under the receiver operator characteristic curve is a performance metric that visually displays the trade-off between the true positive rate and the false positive rate

at different thresholds. In Figure 16, we have plotted the ROC AUC curve for the LightGBM model on the validation data. It significantly improves the dotted diagonal line, which represents random guessing.

Figure 16 The Receiver Operating Characteristic curve for the LightGBM classifier, which has an Area Under Curve score of 0.817 on the validation data.



As is evident from the default model, the costs associated with the different classes are asymmetric. For example, enrolling a lead yields a substantial profit, while misclassifying a potential lead incurs a sizeable potential profit loss. On the other hand, a misclassified observation from the majority class will only inflict a small cost, and correctly classifying an observation from the majority class will save us a small cost.

The asymmetric cost associated with the two classes is highly decisive for the precision we want from the model. We know that we want to capture as many of the minority class as possible, not caring if this results in a false positive rate closer to 1. The ROC AUC plot above shows that the optimal threshold will be the threshold where the true positive rate is close to or equal to 1, which provides us with a false positive rate of

~0.9. Isolated, this does not sound impressive, as it is close to the default model, but this will, in fact, provide us with the highest profit, which is what we are after.

4 Discussion & recommendations

4.1 Verdict of machine learning approach on metadata for predicting lead conversion.

Our analytics approach has proven to be feasible, with reasonable performance, and is theoretically financially profitable. In addition, the metadata used has informative value for predicting conversion and could streamline the contact process to increase the business model's profitability.

Using the algorithm with the proposed set of actions: ignoring the lead, or contacting the lead, yields a theoretical expected profit increase of NOK ~400 thousand on the hold out data. Furthermore, this approach serves as the baseline for further iterations and exploring additional data sources to improve the model's performance.

4.2 Practical application of the solution – An enrollment maximization approach.

Given the financial value that a successful conversion brings to the organization, I's not a surprise the extent to which the company will put an effort to contact and try to convert all possible leads. Additionally, given the relatively low costs of a miss-classified false-positive, our model's profit-maximizing threshold favours the classification of the positive class, regardless of its precision or accuracy.

As a result, our initial suggested approach, which recommends ignoring ~5% of the leads, may not seem at first sight a convincing strategy for improving performance. Additionally, the idea of ignoring leads that may contain potential enrollers may seem naive. Such a shift in strategy and operation schemes may be unfeasible or have a considerable delay in being put into action.

Besides the lost revenue for leads not contacted, the reduced number in enrollment may have business consequences that are not clear to us, such as missing shareholders' mandates, expectations or legal requirements set by the ministry of education. Therefore, we present a heuristic approach as an alternative to capture the value from the false negative class.

As presented in Table 3, instead of two actions (contact/ignore), the heuristic approach considers three sets of actions based on model output, applying two cut-off thresholds. The first cut-off is the threshold that outputs a ~20% precision performance. In other words, a threshold that leaves us with a list of leads, where 20% are expected to enroll, compared to 3.9% in the full dataset. Effectively, this increases the current conversion rate five-fold. The second cut-off is the recall maximizing threshold of 0.0025 to ensure all true positives are captured.

Table 3 Heuristic approach of treatments for the leads based on two cut-offs

		Predicted labels		
		0		**1**
Threshold cut-off (range)		**0 to 0.0025**	**0.0025 to 0.08**	**0.08 to 1**
True labels	**0**	Ignore and do not purchase the lead	External low-cost contacting process	Internal High efficiency contacting process
	1			

Based on these two cut-offs, the company could engage with the highest probability leads itself to ensure a fast conversion with a highly effective experience-driven process. Moreover, an external contractor to extract the maximum value from the rest of the leads. A pay-on-conversion approach could be financially profitable if the individual cost of the solution does not exceed our estimated contacting costs for the amount in Table 5.

This approach's expected counts and potential financial value are presented in Table 4, where the thresholds are extracted from the validation data and applied to the hold out data. 55,063 leads with 2,002 false negatives at a hypothetical cost of conversion of NOK 3,500. A cost paid out only for successful conversions allows us to determine the potential incremental cost of working on these lower probability leads through an external at NOK ~7 million.

Table 4 Expected observation counts and financial profit from each action,

		Predicted labels		
		0		**1**
Threshold cut-off (range)		0 to 0.0025	0.0025 to 0.08	0.08 to 1
True labels	**0**	403 leads Profit: 0	55,063 leads Cost: 0 NOK Revenue: 0 NOK Profit: 0 NOK	4,138 leads Profit: -193 NOK
	1	0 leads Profit: 0	2,002 leads Cost: 3,500 NOK Revenue: 33,273 NOK Profit: 29,773 NOK	1,000 leads Profit: 33,083 NOK
Total		403 leads	57,065 Leads TN profit: 0 FN profit: 59,605,546 NOK	5,138 leads FP profit: -797,102 TP profit: 32,080,000 Total: 32,282,898 NOK

We want to stress the fact that this is purely an example to showcase how our suggested heuristically approach could perform. The values in Table 4 are calculated under the assumption that the performance of the external contracting team that works on a pay-per-lead salary scheme are able to successfully contact and enroll all expected enrolling leads in the second dataset with a lower proportion of profitable leads. This assumption is of course a stretch. Supposed the external sales team possesses less knowledge on UIN's different programs, which would probably decrease the success rate of enrolment within this team.

Therefore, the expected profit obtained by the external sales team in Table 4 is the maximum obtainable profit from the heuristic approach with the current model. An interval where we assume that the external sales team manages to enroll 40%-80% of the leads, is therefore more plausible.

Another assumption made in the above example is that the implementation cost of operationalizing and managing the machine learning model is zero. Considering the expenses incurred from e.g., employing a data scientist, setting up the required technical infrastructure, fetching data and the time it would take to reorganize the internal sales team, the cost of operationalizing our suggested machine learning model to predict profitable leads would be greater than zero. In a short-term perspective, this will involve an increased cost. Over a longer horizon, when the initial costs of setting up the required internal structure is paid and the maintenance costs are reduced, we could see that most of the expected profit from our approach is acquired.

Considering the requirements of operationalizing our heuristical approach and the uncertainties it brings, we cannot say for certain that it would leave the company with a higher profit. We definitely see the value this research can generate. More uncertain is how much of this increased value that can be captured in real life.

5 Limitations & further research

Throughout this paper, we have made assumptions that affect how we can interpret the results. this chapter, these weaknesses and questions are given more attention and detail.

5.1 Limitations

5.1.1 Optimal action

All the data we have worked with when evaluating the models and results is related to leads that have been in contact with UIN's sales team. Therefore, we have assumed that any lead not contacted by the sales team, or an individual outside of the dataset,

would have a probability of enrolling equal to 0. This assumption holds in an isolated research scenario, while in a real-world scenario, leads can, of course, enroll in one of UIN's programs without ever being in contact with the sales team. Thus, our theoretical optimal threshold is a simplification.

One could expand on the current information and let there also be a probability *p* for the probability of enrollment without being in contact with the sales team. If we also include the two actions, *a = 'contact lead'* and *b = 'do nothing,'* we get the equation

$$p_{ia}u(X_{ia} = 1) + (1 - p_{ia})u(X_{ia} = 0) > p_{ib}u(X_{ib} = 1) + (1 - p_{ib})u(X_{ib} = 0)$$

Here, a lead would be contacted if the expected profit from being contacted is greater than the expected profit from not being contacted. This model further improves our current theoretical threshold, with the assumption that leads can enroll without being contacted by the sales team.

5.1.2 Data

As explained, the original raw dataset consisted of 550K leads. However, due to our choice to merge the dataset with the corresponding cycle to which each lead belonged, we ended up with half of the original leads. The reason for this is that we found it correct to train on a full cycle and evaluate the models on the next, with the drawdown of losing data to train on. We cannot rule out the possibility of obtaining a more fine-tuned model, had we had more data to train on.

Additionally, most of the data we used for training and evaluating purposes relates to the cycles 2022 cycle 1 and 2. 2021 cycle 1 and 2022 cycle 3 were severely decreased in size when we merged the data with the corresponding cycle information. We see it as a potential weakness that the models were only trained and evaluated on two full cycles. This problem could have been solved by including data from additional full cycles, which we did not have the luxury of.

We identified two limitations of URL attributes. First, given the nature of the attributes, each URL address contains information that we could not explore given the limited time. For example, the kind of site each URL corresponded to could potentially open the opportunity for more feature manufacturing. The second

limitation is that if a URL has a change in the address, the algorithm will not recognize it anymore. This should be expected given the nature of URL websites and their constant updates.

Finally, the data was provided by the research subject (UIN) after a careful specification and review to avoid a target leak. However, we cannot guarantee that no target leak can be found in these attributes because of our lack of ownership, expertise, and knowledge of UIN's systems and data generation operations. This issue, however, can be explored and learned with a sandbox deployment to test the model with live data and iterate further.

5.1.3 Calibration

In our empirical analysis, we evaluate if the models are well calibrated with two methods. First, we visually explore if the model matches the perfectly calibrated line, over the most critical probability range. The disadvantage of visually evaluating if the model is well calibrated is the way the calibration curve is calculated. The method to calculate the calibration line, bins together observations with close probability predictions, which smooths out the predictions. This can hide deviations in the model's performance and only shows if the model is on average well calibrated.

With the second test, where we check if the theoretical threshold matches the optimal empirical threshold, we are able to determine if the model is well calibrated within that specific range. Here we observe that one of the models is well calibrated. For most of the models, the profit-maximizing threshold did not match the theoretical threshold. We acknowledge that for the most part, the calibrated probabilities cannot be given a true conditional interpretation, which we see as a limitation in our work.

5.1.4 The model

In the 4 Discussion & recommendations, we strive to develop a practical and pragmatic approach to operationalizing the findings in our research. We experiment with how a higher threshold for a lead to be assigned to the 'enrolling' class label can be used as a tool to obtain a list of leads with a high conviction rate. By increasing the classification threshold, the goal is to obtain a list of almost purely enrollers. On the contrary, we observe that even when we increase the classification threshold to a

level where only a fraction of the total population is labelled with the positive class label, the precision obtained does not meet the desired precision.

This lack of distinction between the two classes can relate to a homogenous dataset. After all, the dataset consists of leads that have somehow shown interest in one of UIN's programs. It can also relate to our models not being tuned with the optimal hyperparameters for the specific dataset and classification problem.

6 Conclusion

We have proven that predicting conversion using machine learning on meta-data is possible. All machine learning models explored, performs better than a simple logistic regression model when predicting conversion of leads and outperforms the default process of UIN's current customer acquisition process. The results are derived from metadata related to UIN's leads, so we do not suggest that these results are generalizable to other situations where customer metadata is available.

Financially, predicting lead conversion provides a positive profit expectation compared with the default model. The expected profit improvement is however modest, and we argue that the increased profit is only attainable in theory. This reflection is based on the consideration of expenses incurred if machine learning were to be implemented in the commercial process.

We therefore suggest a pragmatic approach that considers additional actions, with the machine learning model as a tool in the process, rather than a decision maker. This is preferred for strategically important decisions such as this, which impact revenue and sales volume.

References

Athey S, Johannemann J, Hadad V, Wager S (2021) Sufficient representation for categorical variables.

Bronshtein, A. (2017, May 17). Train/test split and cross validation in Python. Retrieved from https://towardsdatascience.com/train-test -split-and-cross-validation-in-Python-80b61beca4b6

Brown E (2017) Gender inference from character sequences in multinational first names. https:// towardsdatascience.com/name2gender-introduction-626d89378fb0#408a

Brownlee, J. (2016a) Mastering machine learning algorithms: Discover how they work and implement them from scratch.

Brownlee, J. (2016b, March 21). Overfitting and underfitting with machine learning algorithms. Retrieved from: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/

Brownlee, J. (2018) Machine Learning Algorithms from Scratch (Vol. V1.7). Author.

Brownlee, J. (2020a) Choose better metrics, balance skewed classes, and apply cost-sensitive learning (Vol. v1.2). Author.

Brownlee, J. (2020b) Data preparation for machine learning: Data cleaning, feature selection, and data transformation in Python.

Brownlee, J. (2020c, July 24). Train-test split for evaluating machine learning algorithms. Retrieved from Train-Test Split for Evaluating Machine Learning Algorithms (machinelearningmastery.com)

Calabrese, R. (2014) Optimal cut-off for rare events and unbalanced misclassification cost.

CatBoost – catboost.CatBoostClassifier documentation available at https://catboost.ai/en/docs/

Cesari A, Zheng A (2018) Feature Engineering for machine learning: principles and techniques for data scientists.

Constantinides, Efthymios & Zinck Stagno, Marc. (2012). Higher Education Marketing: A Study on the Impact of Social Media on Study Selection and University Choice. International Journal of Technology and Education Marketing. Vol 2. 41 - 58. 10.4018/ijtem.2012010104.

Daradoumis, T., Rodríguez-Ardura, I., Faulin, J., Juan, A. A., Xhafa, F., & Martínez-López, F. J. (2010). Customer Relationship Management applied to higher education: developing an e-monitoring system to improve relationships in electronic learning environments. International Journal of Services Technology and Management, 14(1), 103-125.

Dempster, C., & Lee, J. (2015). The rise of the platform marketer: Performance marketing with Google, Facebook, and Twitter, plus the latest high growth digital advertising platforms. John Wiley & Sons.

Elias St. Elmo Lewis. (1903). Catch-line and argument. The Book-Keeper, 15:124–128, February.

Elias St. Elmo Lewis. (1899) Side Talks about Advertising. The Western Druggist. February 21st. p.66

Guillaume Lemaitre, Fernando Nogueira, Christos K. Aridas, (2017) Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning, Journal of Machine Learning Research

Haibo He, Yunqian Ma (2013) Imbalanced Learning: Foundations, Algorithms and Applications.

Hamilton, R., Ferraro, R., Haws, K. L., & Mukhopadhyay, A. (2021). Traveling with Companions: The Social Customer Journey. Journal of Marketing, 85(1), 68–92. https://doi.org/10.1177/0022242920908227

Hassan, S., Nadzim, S. Z. A., & Shiratuddin, N. (2015). Strategic use of social media for small business based on the AIDA model. *Procedia-Social and Behavioral Sciences*, *172*, 262-269.

Hu Y, Hu C, Tran T, Kasturi T, Joseph E, Gillingham M (2021) What's in a name? – gender classification of names with character-based machine learning models.

Ihab F. Ilyas, Xu Chu (2019) Data Cleaning.

Jing Zhou, Wei Li, Jiaxin Wang, Shuai Ding, Chengyi Xia, (2019) Default prediction in P2P lending from high-dimensional data based on machine learning.

Kuhn M, Johnson K, (2013) Applied Predictive Modeling.

Lazarin, V. (2022, April). Personal Communication. [Personal Interview].

Light Gradient Boosting Machine – lightgbm.LGBMClassifier 3.3.2.99 documentation available at https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

Mahsood Shah, & Sid Nair (2016). A Global Perspective on Private Higher Education, edited by, Elsevier Science & Technology, 2016. ProQuest Ebook Central, https://ebookcentral-proquest-com.ezproxy.library.bi.no/lib/bilibrary/detail.action?docID=4455063.

Marinoni, G., Van't Land, H., & Jensen, T. (2020). The impact of Covid-19 on higher education around the world. *IAU global survey report*, *23*.

Mcdougal, Robert & Dalal, Isha & Morse, Thomas & Shepherd, Gordon. (2019). Automated Metadata Suggestion During Repository Submission. Neuroinformatics. 17. 10.1007/s12021-018-9403-z.

Meelis Kull, Telmo Silva Filho, Peter Flach; Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers,

Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR 54:623-631, 2017.

Michaelson, D., & Stacks, D. W. (2011). Standardization in public relations measurement and evaluation. *Public Relations Journal*, *5*(2), 1-22.

Molnar, Christoph. (2022). Interpretable machine learning: Aguide for making black box models explainable. 2nd edition.

Nadeesha Ekanayake (2021, September 17) XGBoost, Light GBM and CatBoost – A comparison of decision tree algorithms and applications to a regression problem https://medium.com/octave-john-keells-group/xgboost-light-gbm-and-catboost-a-comparison-of-decision-tree-algorithms-and-applications-to-a-f1d2d376d89c . OCTAVE – John Keells Group

Niculescu-Mizil, A. Caruana, R. (2005) Predicting Good Probabilities With Supervised Learning. Department Of Computer Science, Cornell University.

Pashootanizadeh, M., & Khalilian, S. (2018). Application of the AIDA model. Information and Learning Science, 119(11), 635-651. doi:http://dx.doi.org/10.1108/ILS-04-2018-0028

Platt, J. (1999) Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods

Prashant Gupta (2016, May 17) Decision trees in machine learning. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

Provost, F., & Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking (1st ed.). O'Reilly.

Python Software Foundation Python Language Reference, version 3.9. Available at https://docs.python.org/3/

Rafiq Ahmed Mohammed, Kok-Wai Wong, Mohd Fairuz Shiratuddin, Xuequn Wang (2018) Scalable Machine Learning Techniques for Highly Imbalanced Credit Card Fraud Detection: A Comparative Study

Rajan Gupta, Saibal K. Pal (2018) Click-through rate estimation using CHAID classification tree model.

Raman, Pushkala & Wittmann, C. & Rauseo, Nancy. (2006). Leveraging CRM for Sales: The Role of Organizational Capabilities in Successful CRM Implementation. Journal of Personal Selling and Sales Management. 26. 39-53. 10.2753/PSS0885-3134260104.

Rigo, Guy-Emmanuel et al. CRM ADOPTION IN A HIGHER EDUCATION INSTITUTION. JISTEM - Journal of Information Systems and Technology Management [online]. 2016, v. 13, n. 1 [Accessed 14 April 2022] , pp. 45-60. Available from: <https://doi.org/10.4301/S1807-17752016000100003>. ISSN 1807-1775. https://doi.org/10.4301/S1807-17752016000100003.

Robert S. Pindyck, Daniel L. Rubinfeld (2017) Microeconomics, 9th edition. Pearson.

Salto, D.J. To profit or not to profit: the private higher education sector in Brazil. *High* Educ 75, 809–825 (2018). https://doi-org.ezproxy.library.bi.no/10.1007/s10734-017-0171-8

Salto, D. (2020). COVID-19 and Higher Education in Latin America: Challenges and possibilities in the transition to online education. eLearn, 2020(9).

Sen Zang, Zheng Liu, Wendong Xiao (2018) A hierarchical extreme learning machine algorithm for advertisement click-through rate prediction.

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Somasundaram, A., Srinivasulu Reddy, U., (2016) Data Imbalance: Effects and Solutions for Classification of Large and Highly Imbalanced Data.

Sucarrat, G. (2017). Metode og Økonometri en moderne innføring (2nd ed.). Fagbokforlaget.

Tanner, Ahearne, M., Leigh, T. W., Mason, C. H., & Moncrief, W. C. (2005). CRM in Sales-Intensive Organizations: A Review and Future Directions. The Journal of

Personal Selling & Sales Management, 25(2), 169–180. https://doi.org/10.1080/08853134.2005.10749057

Uncles MD. Directions in Higher Education: A Marketing Perspective. Australasian Marketing Journal. 2018;26(2):187-193. doi:10.1016/j.ausmj.2018.05.009

Urduain, R. (2022, April). Personal Communication. [Personal Interview].

Vieira, Valter & Claro, Danny. (2020). Sales Prospecting Framework: Marketing Team, Salesperson Competence, and Sales Structure. BAR - Brazilian Administration Review. 17. 10.1590/1807-7692bar2020200025.

B. C. Wallace and I. J. Dahabreh, "Class Probability Estimates are Unreliable for Imbalanced Data (and How to Fix Them)," 2012 IEEE 12th International Conference on Data Mining, 2012, pp. 695-704, doi: 10.1109/ICDM.2012.115.

Zixuan Zhang (2019, June 26) Boosting Algorithms Explained. https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30

XGBoost – xgboost.XGBClassifier 1.6.1 documentation available at https://xgboost.readthedocs.io/en/stable/

Appendix

Appendix A Missing values per attribute

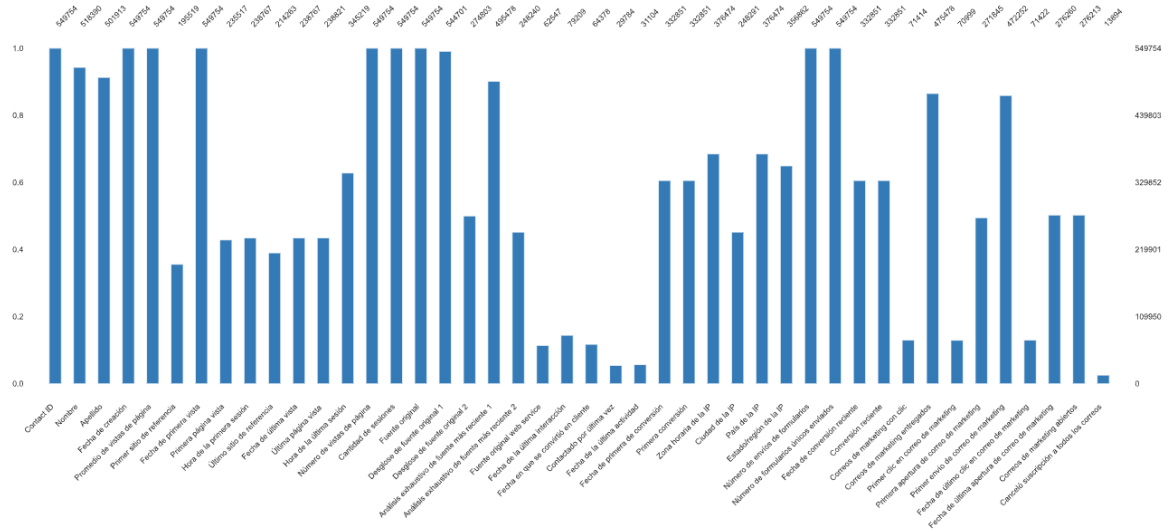


Table 5 Raw data dictionary and missing values management. Categorical features are one-hot encoded with the category included in the column name, in the cleaned data.

Original column name	Renamed column name	Description of the attribute	Data Type	How we managed Missing values
Número de envíos de formularios	number_of_forms_sent	How many data request forms have been sent to or from this user	number	no missing
Número de formularios únicos enviados	number_of_unique_forms_sent	Number of unique data request forms have been sent to this user	number	no missing
Fecha de primera de conversión	first_conversion_date	First date this user had any conversion, that is, moved into the funnel stages	datetime	median
Primera conversión	first_conversion_action	First action where the lead engaged and executed the promped action	categorical	replaced with 'missing'
Conversión reciente	recent_conversion	Most recent action where the lead engaged and executed a promped action	categorical	replaced with 'missing'

Canceló suscripción a todos los correos	cancel_subscription_to_all_emails	If the user unsuscribed to the emailing list-	bool	no missing
Primer clic en correo de marketing	click_on_marketing_mail	First date this user clicked on a marketing email	datetime	(median)
Primera apertura de correo de marketing	first_marketing_mail_open	First date this user opened an email	datetime	(median)
Primer envío de correo de marketing	first_marketing_mail_shipping	First date the company sent an email to this user	datetime	(median)
Fecha de último clic en correo de marketing	date_of_last_click_on_marketing_mail	Last date this user clicked on a marketing email	datetime	(median)
Fecha de última apertura de correo de marketing	date_of_last_marketing_mail_open	Last date this user opened a marketing email	datetime	(median)
Correos de marketing con clic	marketing_emails_clicked	Number of emails that have received a click from the user	number	replace with 0
Correos de marketing entregados	marketing_emails_delivered	Number of emails that have been succesfully delivered to this user's email	number	replace with 0
Correos de marketing abiertos	marketing_emails_opened	Number of marketing emails that his user has opened	number	replace with 0
Fecha de primera vista	first_view	Date when the user visited the websites for the first time	datetime	(median)
Hora de la primera sesión	time_of_first_session	Hour (Time) of the first session the user visited the webite	datetime	(median)
Fecha de última vista	last_view	Last time (date) the user visited the website	datetime	(median)
Hora de la última sesión	time_of_last_session	Hour (Time) of the last time the user visited the webite	datetime	(median)
Fecha de creación	creation_date	Date the lead was created in the system	datetime	(median)
Fecha de la última interacción	date_of_last_interaction	Last date the user had any kind of interaction with the company's content	datetime	(median)

Fecha en que se convirtió en cliente	date_when_it_became_client	Date when the user became a lead, a leadmeans _____ ???	datetime	(median)
Fecha de la última actividad	date_of_last_activity	Date of the last activity this user had in the CRM	datetime	(median)
Fecha de conversión reciente	recent_conversion_date	Last date this user had any conversion, that is, moved into the funnel stages	datetime	(median)
Promedio de vistas de página	avg_page_view	Average pages this lead visits within the website	number	no missing
Número de vistas de página	number_of_page_views	Number of pages this lead has visited in the website	number	no missing
Cantidad de sesiones	number_of_sessions	Number of sessions this user has accumulated in the website	number	no missing
Primer sitio de referencia	first_reference_site	First site this user got refered to	categorical	replaced with 'missing'
Primera página vista	first_page_view	First page this user saw within the website	categorical	replaced with 'missing'
Último sitio de referencia	last_reference_site	Last site this user got refered to	categorical	replaced with 'missing'
Última página vista	last_page_view	Last page this user saw within the website	categorical	replaced with 'missing'
Contactado por última vez	last_contact	Last time the lead was contacted	datetime	(median)
Zona horaria de la IP	time_zone_of_ip	Time zone of the lead's IP address	categorical	replaced with 'missing'
Ciudad de la IP	ip_city	City of the IP for the lead	categorical	replaced with 'missing'
País de la IP	ip_country	Countrye of the lead's IP address	categorical	replaced with 'missing'
Estado/región de la IP	ip_state_region	State / Region of the lead's IP address	categorical	replaced with 'missing'
Fuente original	original_source	Source this lead was generated from	enumeration	no missing
Desglose de fuente original 1	original_source_breakdown_1	Level 1 Breakdown of the lead source	categorical	replaced with 'missing'
Desglose de fuente original 2	original_source_breakdown_2	Level 2 Breakdown of the lead source	categorical	replaced with 'missing'
Análisis exhaustivo de	exhaustive_analysis_of_most_recent_source_1	Level 3 Breakdown of the lead source	categorical	replaced with 'missing'

fuente más reciente 1				
Análisis exhaustivo de fuente más reciente 2	exhaustive_analysis_of_most_recent_source_2	Level 4 Breakdown of the lead source	categorical	replaced with 'missing'
Fuente original web service	original_source_web_service	Original source if the lead was generated by web service	categorical	no missing

Figure 17 Histogram of the predicted probabilities on the validation data, using the LightGBM model.

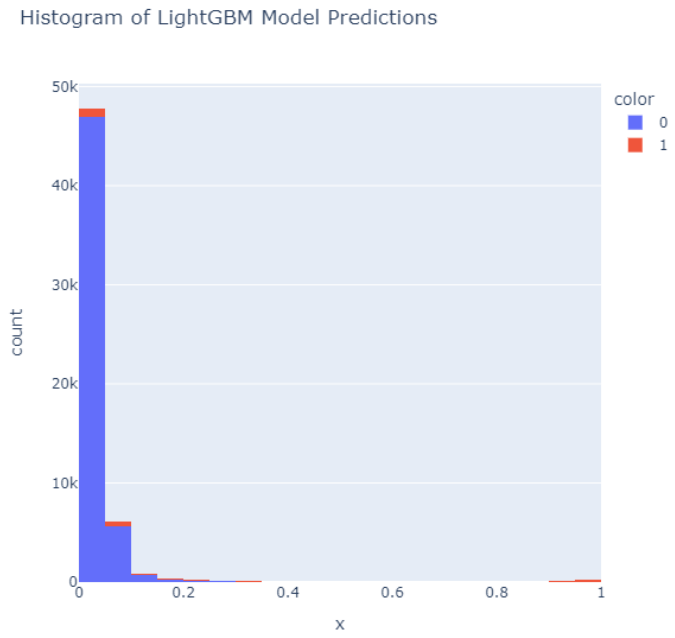


Figure 18 Histogram of the predicted probabilities on the validation data, using the XGBoost model.

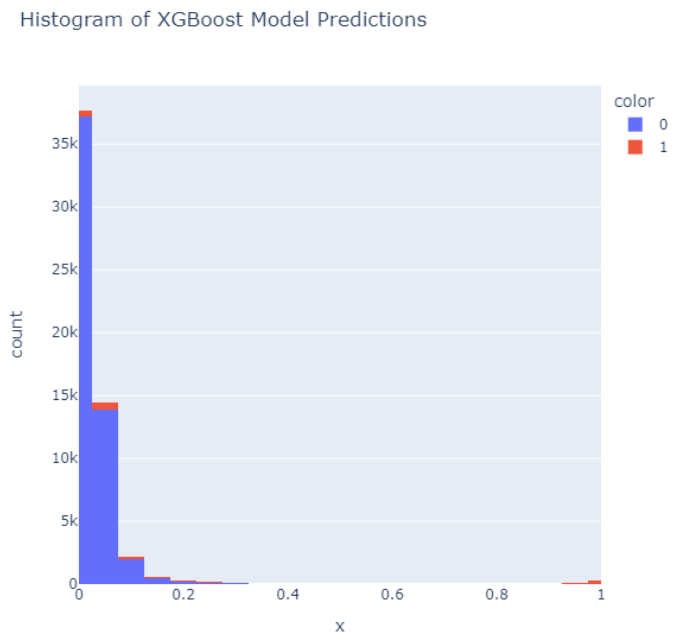


Figure 19 Histogram of the predicted probabilities on the validation data, using the CatBoost model.

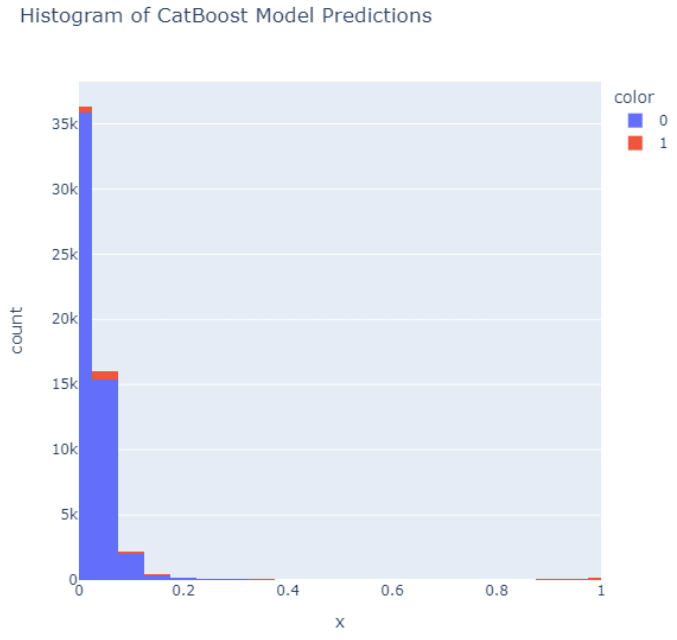


Figure 20 Plot of the probability curve with LightGBM model on the validation data. The dotted diagonal line represents a perfectly calibrated model.

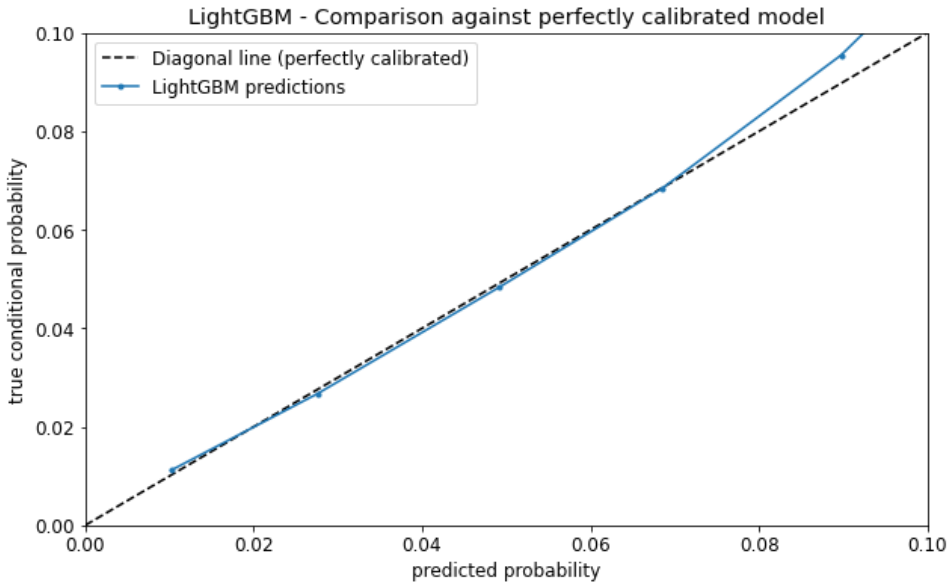


Figure 21 Plot of the probability curve with the XGBoost model on the validation data. The dotted diagonal line represents a perfectly calibrated model.

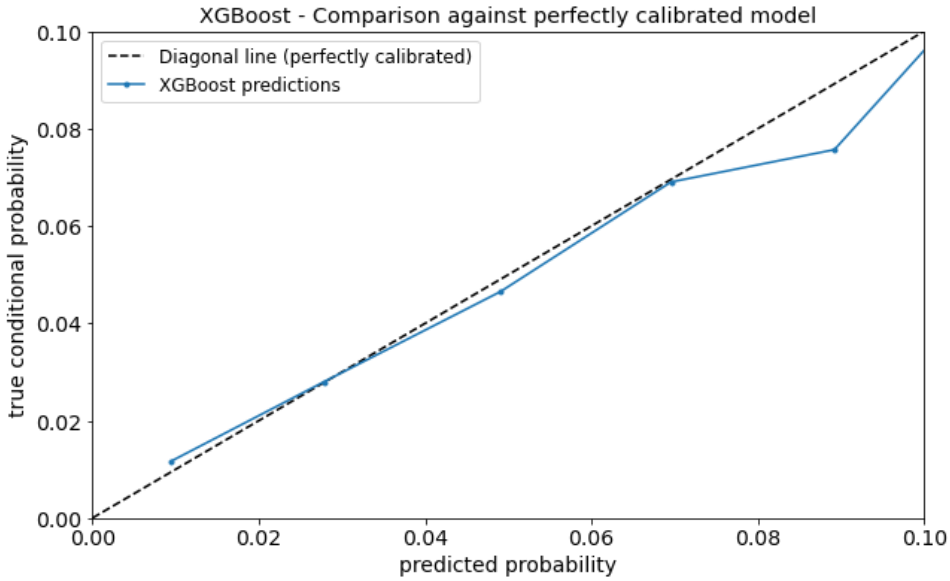


Figure 22 Plot of the probability curve with the CatBoost model on the validation data. The dotted diagonal line represents a perfectly calibrated model.

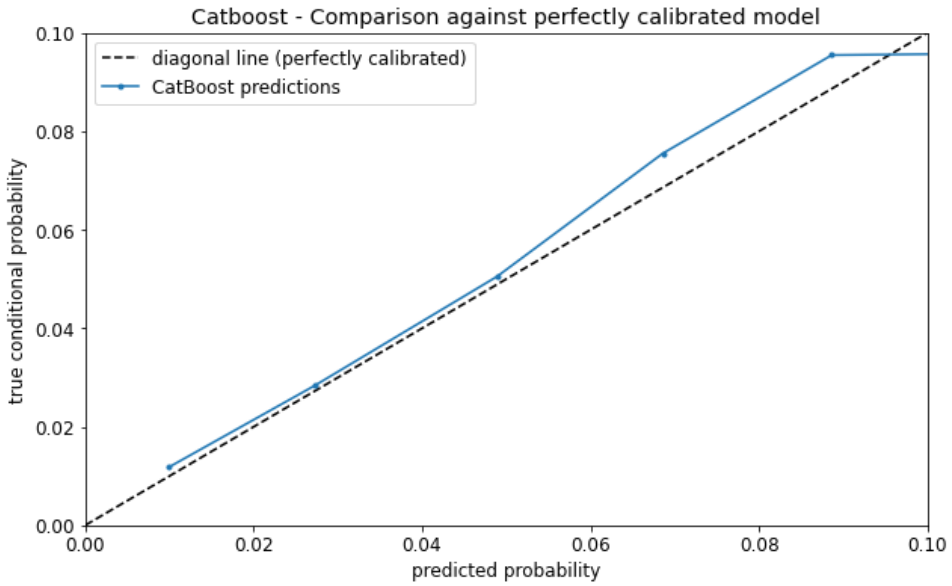


Figure 23 Obtained profit over different threshold levels obtained with the LightGBM model with predictions on the validation data.

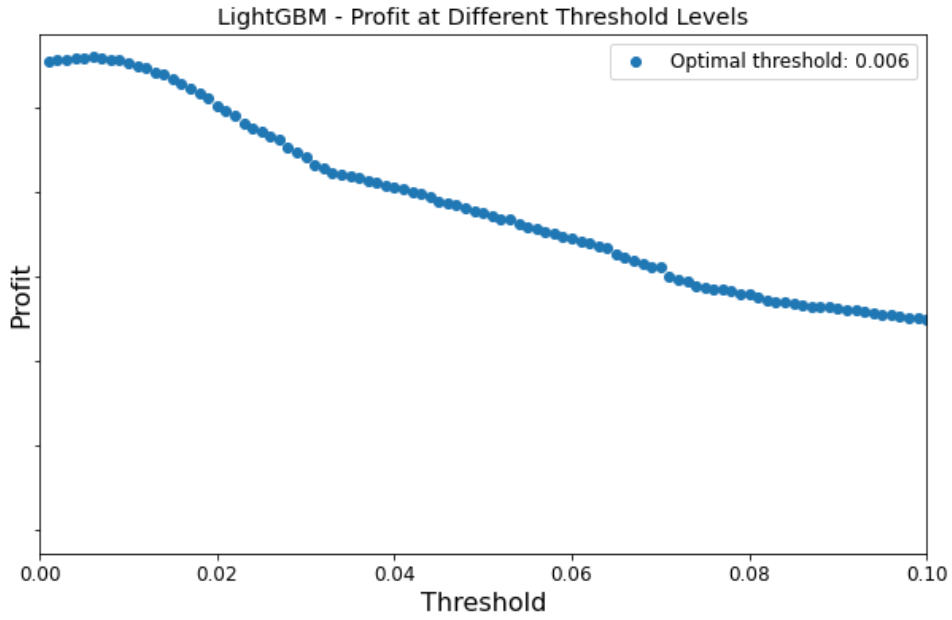


Figure 24 Obtained profit over different threshold levels using the XGBoost model with predictions on the validation data.

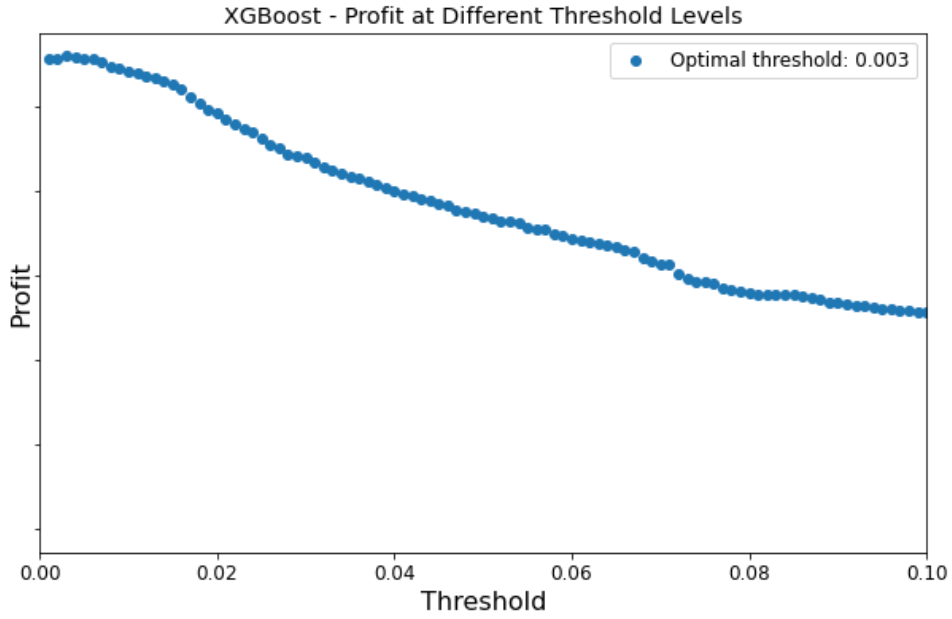


Figure 25 Obtained profit over different threshold levels using the CatBoost model with predictions on the validation data.

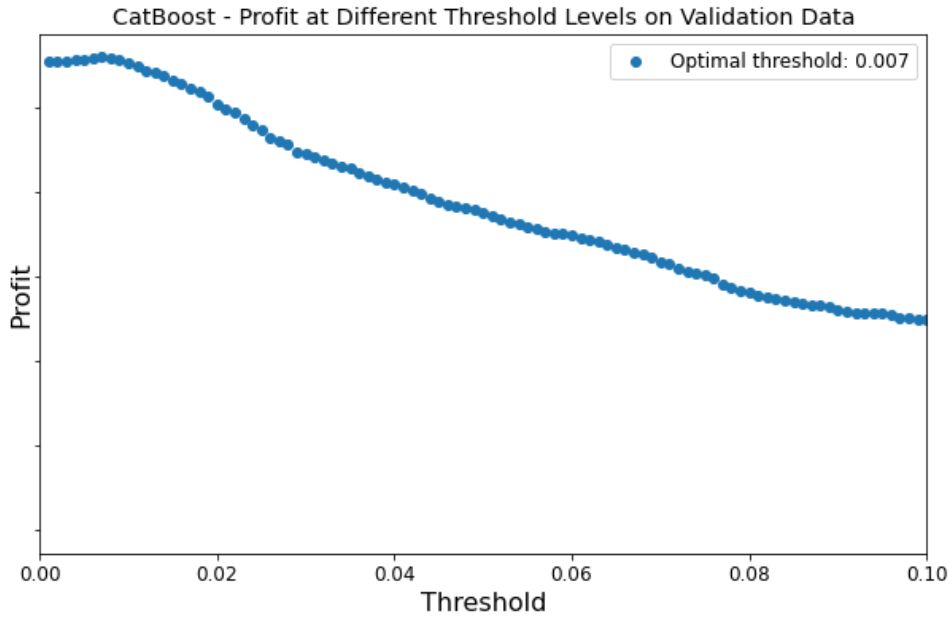
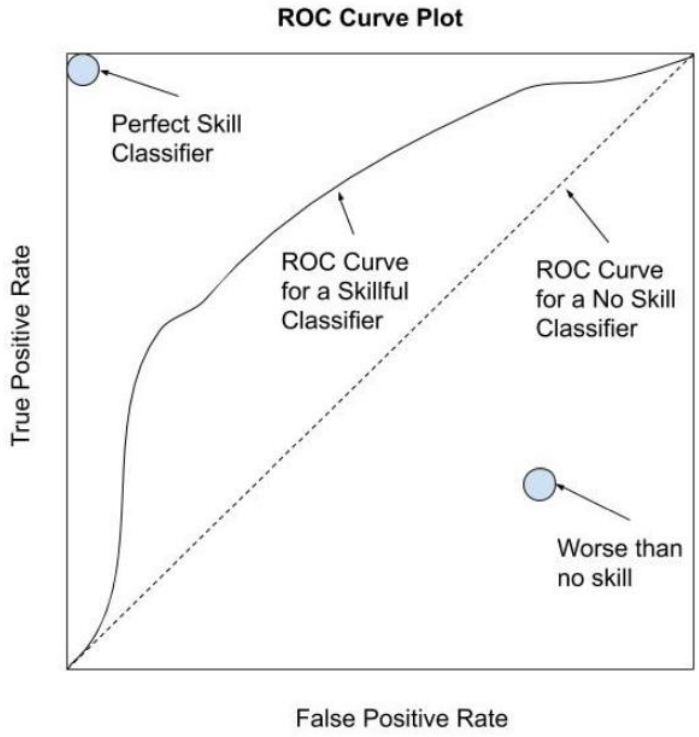


Figure 26 Depiction of a ROC Curve (Brownlee, 2020a, p. 42)



Appendix B Top 50 variable importance graph explained by LightGBM algorithm, for the minority class.

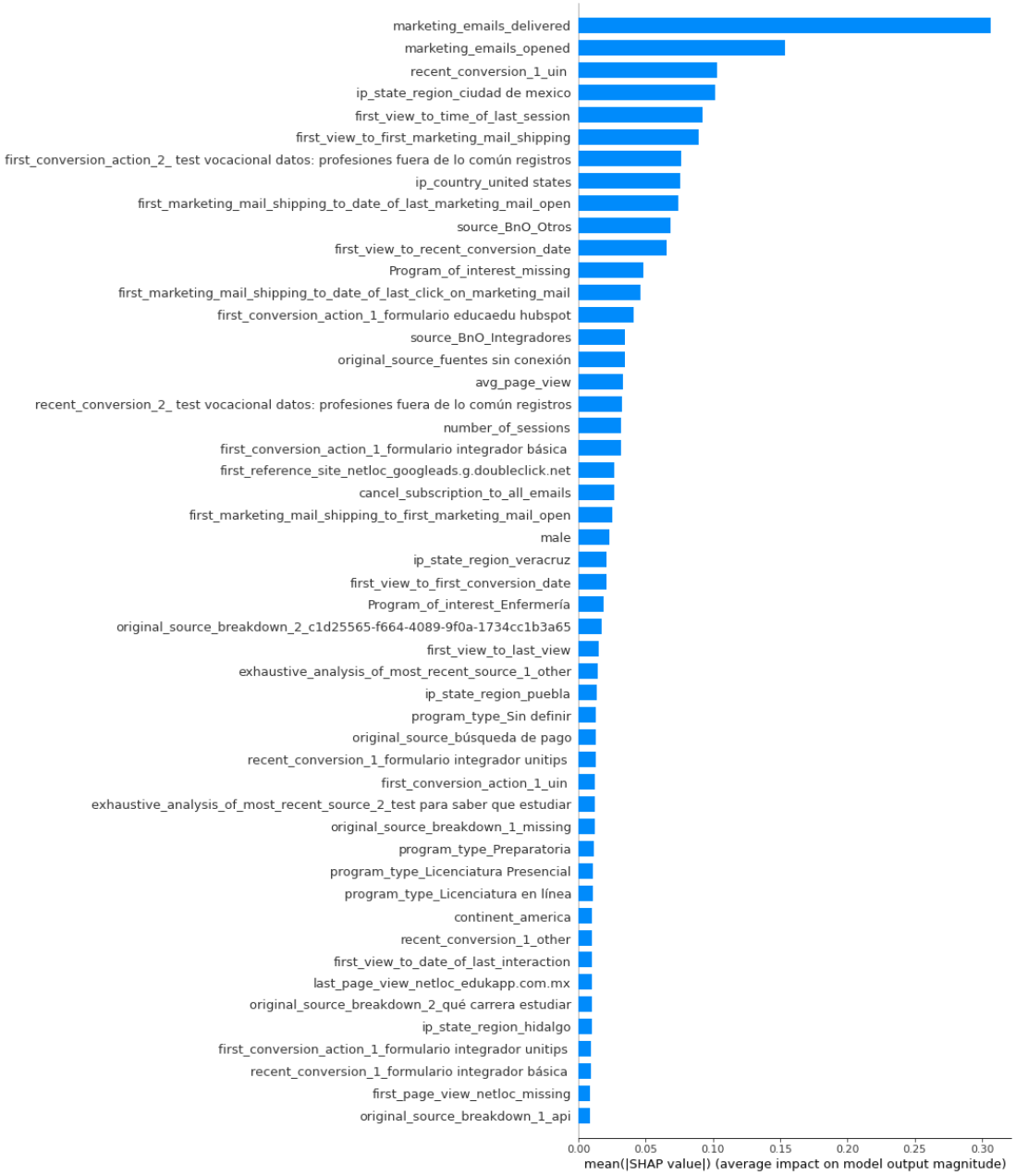
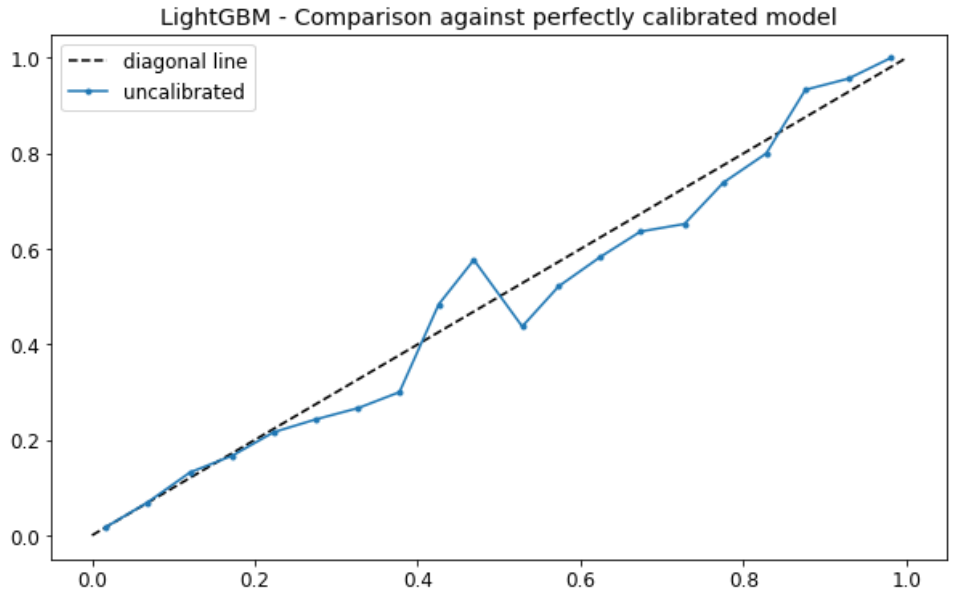


Figure 27 Predicted scores over full range of x-axis.



Appendix C Data Cleaning and pre-processing

Data Cleaning and Pre-processing

Detailed Description of raw data features

Boolean

In our raw tabular dataset, there is only one variable that contains a Boolean datatype. This variable contains the TRUE value if the lead has cancelled its subscription to all emails and is represented with a missing value where there is no information about the cancellation action. In our thesis we will assume that a missing value in this column can be replaced by the Boolean value FALSE.

Time

The dataset contains multiple variables that represent the point in time of a specific action, related to the behaviour of the lead which introduces a chronological time-hierarchy. As each of these logged time variables are related to a specific action, there are missing values where a lead has not yet finalized the full journey to the last tracked interaction. The first point of contact is always available, because the lead generation is always initiated by a specific action of that lead. The subsequent variables in the time-hierarchy have increasingly more missing values as the number of leads that reach all the way to the end of the journey is decreasing. Appendix D shows a dendrogram

of the datetime variables. The dendrogram is a visual representation of the correlation between variable completion, which also gives a visual impression of the chronological order of the datetime variables.

Categorical

A categorical feature is a type of variable that can have two or more groups. In our dataset we have multiple categorical variables that hold information related to geography, name, URL, origin, and conversion. Some of these categorical features represent an action that is again tied to a logged time. Because of this relationship between an action and the logged time of that action, there is a similar chronological hierarchy of these actions and consequently also missing values where a lead has not yet reached that phase of the tracked journey.

Not all the categorical variables are linked to a point in time, some hold information related to the lead and others are aggregations of the journey of the lead. Where the data provider, HubSpot is not able to capture the categorical feature, that variable contains a missing value.

Another important characteristic of the categorical variables in the dataset is the high cardinality of the data. High cardinality is defined as the number of unique values a variable can have, where a high cardinality feature describes a variable that can take the number of *n* values where *n* is large. Names are a common high cardinality feature, as names can come in many different variations. In our dataset we have high cardinality in the name, URL, geography, origin, and conversion variables.

Numerical

Numerical features represent the last type of variables in the dataset. These variables describe information about how many emails or forms a lead has received, page views and sessions. This information is represented in integers in the raw data. As some of the leads have not yet reached the stage where they receive marketing emails, the values here are missing rather than having a value of zero.

Another important feature of the numerical variables is that their distributions are all right skewed and leptokurtic.

Detailed Description of data cleaning and pre-processing

Appendix E Detailed description of data cleaning and pre-processing

Numerical Variables

Missing values is a frequent problem in real-world data and is often caused by corrupt data or failure to record data. The handling of missing values is a particularly crucial step in the pre-processing of data, as many machine learning algorithms do not handle missing values. Another aspect when dealing with missing values is that it needs to be handled in a way where it does not add features that were not originally in the dataset, where under other circumstances the missing values represent some information that needs to be included.

The numerical variables in our dataset mostly consist of variables with no missing values, except for three columns related to marketing email activity. These columns have missing values when no marketing email activity is recorded. We therefore assume that no marketing email activity can be considered as a value of 0, replacing all missing values with 0. In the dataset we also have a Boolean variable, where only the variable for TRUE is recorded. We therefore assume that missing values in this column can be replaced with the value FALSE. The boolean variable is then converted into a binary variable [1,0].

The main task of any machine learning model is to minimize the loss function. Some machine learning models assume normally distributed features, while others are not affected by skewed distributions. Since our numerical features are both right skewed and leptokurtic, we want to reduce the variance of the data and make it more normally distributed for algorithms to give equal importance to all samples. We obtain this by utilizing a power transformation, where we have chosen the log +1 transformation of our numerical features. Log transformation is an example of transformations known as power transformations, that in statistical terms are known to be variance-stabilizing transformation (Cesari, 2018, p. 23).

Time Delta

Our dataset consists of rows, where every row represents a lead and the journey of that lead from the point of generation. As the datetimes related to each lead represent a journey in time from the point of generation, they do not relate to the datetimes of other leads and their journey in time after generation. We therefore need to generalize the time features of the dataset so that they convey time as a value from generation.

We do this by calculating the time difference from time *a* until time *b*, and so forth. For each lead, we will now, instead of having variables containing datetime features, get new variables that tell the time difference from generation to the next interaction of that lead. Converting the datetime features of the dataset enables us to compare the dime difference in minutes from one interaction to the next across the leads, unrelated to when the lead was generated.

Appendix F displays the time hierarchy of the interactions of a lead and can be interpreted as actions after generation by reading from left to right and top to bottom. In the processing of the new time-delta variables we use the first interaction of a lead as the base to calculate the time differences as this variable is complete, without missing values. This way we manage to maintain more of the information as many of the datetime variables contain a high percentage of missing values and measuring the time-delta between each action would lead to more missing values as both datetimes must be present for the time-delta to be calculated.

As a further step we convert the time delta features from seconds to minutes and convert them to absolute values. The reason we take the absolute values of the time deltas is that we have outliers that are negative, meaning that a lead has somehow been generated before its first interaction, for instance. Because the time delta still represents a time difference, we use the absolute value, not to get outliers on the left tail of the distribution. The time features are then log +1 transformed.

Categorical variables

Categorical variables can be divided into two categories, nominal, and ordinal. A nominal categorical variable means that there is no inherent order in the categories,

while an ordinal categorical variable is a set of variables where there is an inherent order between the values. In our dataset there are multiple categorical features and all of them belong to the nominal group of categorical variables.

The next distinction for the categorical variables is whether they have a high or low degree of cardinality. Depending on the degree of cardinality determines how one must deal with the variable in the cleaning and pre-processing of the data and in our dataset, we have variables of both high and low degrees of cardinality.

Low cardinality variables

There are two variables in our dataset that we consider to be nominal low cardinality features and they both relate to the origin of how a lead was generated. One consists of 9 unique categories with no missing values and the other, 7 unique categories with missing values. As explained previously, the missing values in our dataset relate to one of two categories, that lead has not yet reached that stage in the tracked journey, or the data provider was not able to capture the desired feature. Either way, a missing value represents information that we want to pass on in the cleaned dataset. The low cardinality features are left untouched in the cleaning and pre-processing stage.

High cardinality variables

Many learning algorithms require categorical data to be transformed into real vectors before it can be used as input. Often, categorical variables are encoded as one-hot or dummy vectors. However, this mode of representation creates sparsity in the dataset with many low signal regressors, especially when the number of unique categories is large (Athey et al. 2021). Another problem with converting high cardinality features into a binary representation is that as the number of features in the dataset grows, the amount of data we need to accurately distinguish between features and to generalize the model grows exponentially. This is popularly mentioned as the curse of dimensionality. As most datasets contain a combination of numerical and categorical features, the challenge of high cardinality is not new, so a lot of research is done on the topic. The answer to dealing with high cardinality is ambiguous, but many proposed solutions to several types of cardinality challenges are available. We have explored some of them to deal with high cardinality in our dataset.

Gender classification

Included in our tabular dataset we have two columns representing the name of each lead. One column with the first name, and one column with the last name. The name information is in string format and represents a high cardinality data, also referred to as a dirty variable problem. Many machine learning algorithms cannot handle categorical information and in those who do, high cardinality is a problem when the number of categories increases. An important task in the data cleaning process is therefore to simplify the name information and at the same time keep valuable information that holds predictive properties.

As gender information is not necessarily a mandatory information input when registering for online accounts and not typically an information property that is captured by data providers operating with metadata capturing, a lot of research is made on how to infer gender from known data, as the gender information is assessed to have predictive properties. Brown (2017) proposes to use features from the first name, as this is believed to be the most telling indicator of a person's gender. The described features are first/last letter, count of letters and suffixes (last 2, 3, 4, first 2, 3, etc.) of a name. The features are then used to train a NLTK Naïve Bayes classifier. Hu Y et al. (2021) goes a step further by proposing a character-based machine learning model that utilizes both first and last names to improve classification accuracy, as the first names may have different gender connotations across cultures. They also show how content information (page view, search, clicks etc.) could be a complementary feature to the first name for improved accuracy, as the disclosed name might not be the true name of an individual.

In our thesis we have chosen Brown's suggested approach, as it proves to have high accuracy compared to the simplicity of the model. Our first name column consists of 131,332 unique names, counting combinations of multiple first names as well. There are also 5.7% of the first names that are missing in the dataset, which results in no predicted gender for our chosen approach.

To prepare the first name column for the Naïve Bayes classifier we separate the column on any spaces and keep only the first of the words included. This leaves out any middle

names so that we are left only with the first names. The next step of the classification is to define the features that we want to use from the first name to train the classifier. After some testing we ended up with the highest accuracy score with the last letter, last 2 and 3 letters and the first 3 letters as our features. The classification model shows an 83% accuracy on the test data, which we find substantial enough.

After the gender classification exercise on the first name column, we are left with the last name column unused. This column has 8.7% missing values, which is an increase from the first name column. To maintain the richness of our dataset we chose to convert the last name column into a dummy variable column, where 1 indicates a present last name and 0 indicates a missing last name, as we believe there is a chance that this information can improve the model.

Geographical variables

There are four variables in the dataset related to the geographical location of the lead, at lead generation. They can all be categorized as high cardinality features because they contain many unique values in the form of time-zone, country, city, and region. These variables are all highly concentrated on a few values related to the geographical location of Mexico, much expected since this is where the business operates. They also have a high percentage of missing values which is not negligible since this information potentially has predictive properties as well.

We perform two operations on geographical variables to reduce cardinality. Firstly, we extract the continent from the time-zone variable which in its original form was a concatenation of continent and city. Next, we merge low frequency categories and replace them with “other”, to create a new value that contains all low frequency values. The threshold we use for this operation is any variable that occurs less than 0,2%. After these two operations we are left with a significantly reduced cardinality in the geographical variables, while maintaining most of the information in the dataset.

URL’s

Uniform Resource Locator’s (URL) are string components that represent an address on the web. These strings are built up by addressing scheme, network location, path etc.

As these strings represent a unique location on the web, the variety of unique URLs related to a single webpage can grow large. Our tabular dataset has four variables that hold URL strings, and these variables are by far the features with the highest cardinality in our data with number of unique observations counting above 100,000.

As URL's consists of a universal structure of components, we want to split the full URLs into its different components to reduce the cardinality. For this we use one of the standard libraries in Python named *urllib*. In this library there is a parse function that effectively splits the URL into its components scheme, netloc, path, parameters, query, and fragment. We choose to keep only the netloc and path of the URLs as we evaluate these to include the most amount of information related to the lead's activity, and they are also the features in the URLs that have the highest degree of completion. These two features from the original URL are then split up into separate columns.

To further reduce the cardinality of the URL-related variables we perform the same threshold reduction on low frequency occurrences, with the same threshold of 0.2%. These low frequency occurrences are replaced with "other". After these two operations on the URL variables, we have managed to reduce the cardinality down to the range of 10 – 30 unique values, while maintaining a high degree of information.

Conversion

Conversion represents the last high cardinality variables in our dataset. These variables are a concatenation of up to five actions that describe the leads customer journey so far, where each action is separated with " | ". To reduce the dimensionality of these features we have chosen to use the separator as a split criteria and include each action as a separate variable. Some of the leads have all five actions in place, while others have fewer or none. To further reduce the dimensionality of the categorical variables we perform a frequency threshold reduction, using the same frequency of 0.2% of the total observations as before.

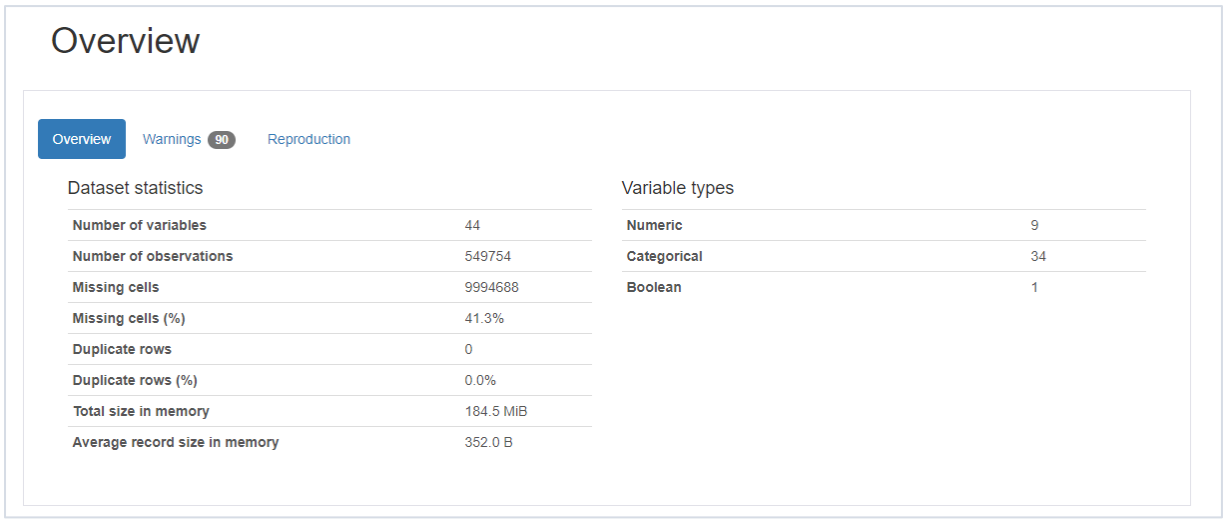
Merging of datasets

The two datasets are merged on the unique leads id to obtain a full dataset that includes all available features on each lead from the two different data sources. All observations

that miss information regarding which cycle that lead belongs to are removed. The reason we do not have cycle information on all leads is related to a change in how UIN stores information on leads, which also results in the lack of enrollment information related to the leads that are missing cycle information. It also helps us partition our dataset into a training and holdout dataset, where we ensure that we do not train the model on future leads, to predict past leads, which would be the opposite of how this process would be operationalized.

Pandas Profiling Report

Appendix G Pandas profiling report summary of metadata dataset



Overview

Overview | Warnings 90 | Reproduction

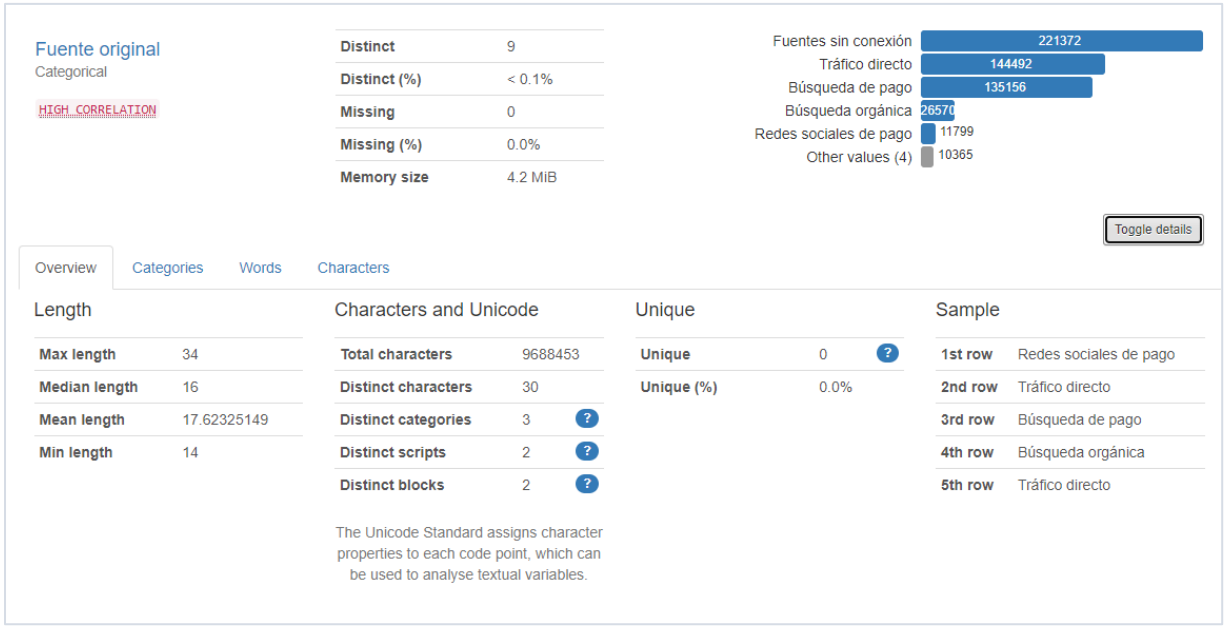
Dataset statistics

Number of variables	44
Number of observations	549754
Missing cells	9994688
Missing cells (%)	41.3%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	184.5 MiB
Average record size in memory	352.0 B

Variable types

Numeric	9
Categorical	34
Boolean	1

Appendix H Pandas profiling report for categorical variable "Fuente Original"



Fuente original

Categorical

HIGH CORRELATION

Distinct	9
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	4.2 MiB

Toggle details

Overview | Categories | Words | Characters

Length

Max length	34
Median length	16
Mean length	17.62325149
Min length	14

Characters and Unicode

Total characters	9688453
Distinct characters	30
Distinct categories	3
Distinct scripts	2
Distinct blocks	2

The Unicode Standard assigns character properties to each code point, which can be used to analyse textual variables.

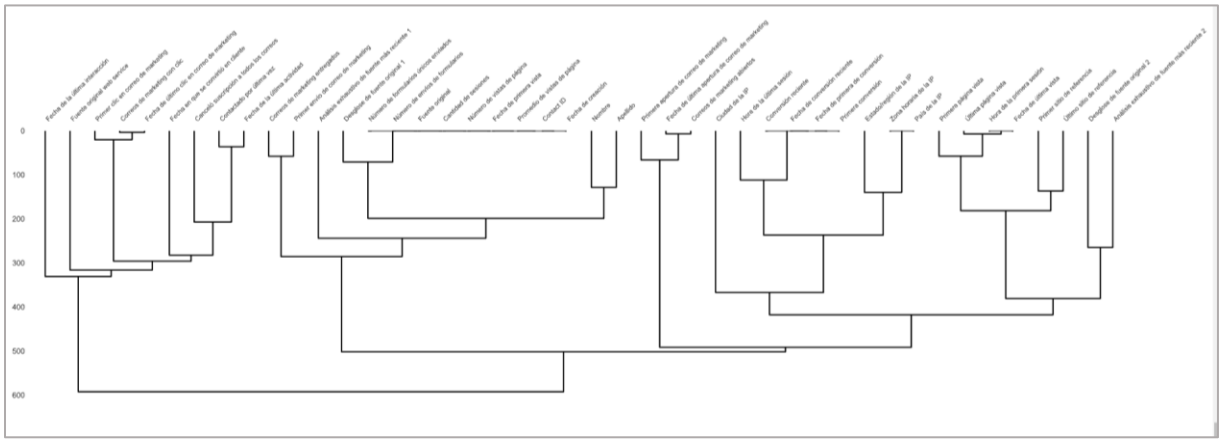
Unique

Unique	0
Unique (%)	0.0%

Sample

1st row	Redes sociales de pago
2nd row	Tráfico directo
3rd row	Búsqueda de pago
4th row	Búsqueda orgánica
5th row	Tráfico directo

Appendix I Pandas Profile Report missing values dendrogram


Fecha de la última interacción
Fuente original web service
Primer clic en correo de marketing
Correos de marketing con clic
Fecha de último clic en correo de marketing
Fecha en que se cancela en correo
Cancela suscripción a todos los correos
Contactado por última vez
Fecha de la última actividad
Correos de marketing entregados
Primer envío de correo de marketing
Análisis estadístico de fuente más reciente
Desglose de fuente más reciente 1
Número de formularios enviados
Número de envíos de formularios
Fuente original
Cantidad de sesiones
Número de visitas de página
Fecha de primera vista
Promedio de visitas de página
Contact ID
Fecha de creación
Nombre
Apellido
Primera apertura de correo de marketing
Fecha de última apertura de correo de marketing
Correos de marketing abiertos
Ciudad de la IP
Hora de la última sesión
Conversión reciente
Fecha de conversión reciente
Fecha de primera en conversión
Primera conversión
Estado/región de la IP
Zona horaria de la IP
País de la IP
Primera página vista
Última página vista
Hora de la primera sesión
Fecha de última sesión
Primer título de referencia
Último título de referencia
Desglose de fuente original 2
Análisis estadístico de fuente más reciente 2
0
100
200
300
400
500
600