A Work Project, presented as part of the requirements for the Award of a Master's degree in Finance from the Nova School of Business and Economics.

SELECTING ACCOUNTS IN A B2B CONTEXT: A DATA-DRIVEN APPROACH

ÉDI MANUEL DA SILVA FERREIRA

Work project carried out under the supervision of:

Qiwei Han
Bruno Silva

04-01-2020

Abstract

The process of selecting accounts in the Business-to-Business context has been mainly reliant on the gut feeling of the sales personnel. With the emergence of advanced analytics and vast volumes of data, this panorama is changing. This project presents an end-to-end pipeline to assess the propensity to buy of an account, before there is any interaction between the parts. It covers the challenges faced throughout the implementation of a data-driven approach in the cloud industry. The results indicate that it is possible to properly rank accounts in order to better allocate the resources of the company, even though a proper baseline comparison is missing.

Keywords: Data-driven decision making; OutSystems; Account Selection; Business-to-Business (B2B); Machine Learning

Acknowledgments: I would like to thank OutSystems for their trust and for giving me the opportunity to develop this project within such good conditions. I would like to extend it to my company's advisor, Bruno Silva, and his team, that made me feel integrated at all times, providing all the needed support. Last but not least, I want to thank my advisor, Qiwei Han, for his outstanding commitment, going well beyond what is expected.

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

1. Introduction

Prior to applying a company's best efforts to acquire a new customer, it is key to define who that customer might be, even before there is an engagement from any of the parts. With the increasing amount of data being collected every day and recent advances in statistical and Machine Learning (ML) methods, a significant number of research and models were applied to the Business-to-Consumer (B2C) segment, using tools such as *lookalike* modeling to define those customers based on their demographics and behavioral characteristics (Dewet and Ou 2019; King 2012). However, when it comes to Business-to-Business (B2B) sales outcome forecasting, the panorama is different, as the literature is still scarce (Rezazadeh 2020). Companies have been using Account-Based Marketing for a while to strengthen their relationships with customers and, therefore, maximizing the lifetime value of each account (Burgess and Munn 2017). In recurring revenue models, this notion becomes even more important as most part of the total revenue is secured after the first transaction is made, making customer retention a critical factor to success (Van Der Kooij 2019). However, to retain an account, it is needed to make this first transaction come true. Therefore, this work project aims to answer three questions: How are the accounts chosen in the first place? How can a company assess an account's propensity to buy without relying solely on its sales personnel gut feeling? What are the main difficulties companies are facing during the process?

This thesis aims to answer the above-stated questions, going through the steps of the sales process within a business context. It focuses mainly on the prospecting stage and goes through the main challenges a data scientist faces when a company tries to implement a data-driven approach to make their final decisions, in order to better allocate their resources. It covers clustering, record linkage, and predictive modeling, as these were found the necessary steps to attain OutSystems' goals and overcome the company hurdles.

This thesis is organized as follows: In Section 2, a review of the literature found relative to the subject is presented. Section 3 introduces the background and data. Section 4 presents the methods used in this work. Section 5 presents the results, discussion of results, limitations of the current work, and potential future directions. Section 6 concludes the project.

2. Literature Review

2.1. Artificial Intelligence in the Business-to-Business Context

The need to introduce Artificial Intelligence (AI) in the sales process is growing day by day as the new leaders need to adapt to the new trends and get on board with the benefits inherent to them (Colter *et al* 2018). Business-to-Business (B2B) companies have seen this process as a more tortuous one, as the customer preferences rapidly change and there are several decision-makers throughout a long and difficult sales process (Ingram 2004).

Firms that are quickly adapting to this new trend, using what Colter *et al* (2018) called the "science of B2B sales", are already witnessing the results: higher revenue, profitability, and shareholder value. It uses a combination of engagement with the customers, advanced analytics, and talent. Instead of relying only on the gut instinct of the sales leaders of the company, as it happened in the past, decisions can now be based upon data collected from several sources. This way, it is possible to tackle specific problems, such as to decide which customers the salespeople should go after and when, in an efficient way, building an augmented profile of each customer (Colter *et al* 2018).

Even though this process was already studied a lot in the Business-to-Consumer (B2C) segment, where companies use AI engines to generate value by providing personalized recommendations

(Singh *et al* 2018) and use what is called the "*lookalike* advertising" to reach out to new customers (Dewet and Ou 2019), the role of AI in the B2B sales is still somewhat underexplored (Singh *et al* 2018; Syam and Sharma 2018). Paschen, Wilson and Ferreira (2020) saw this gap and tried to fill it with a theoretical approach using the seven steps sales process introduced by Dubinsky (1980), going through all steps of the sales funnel and explaining how AI can add value to each one of them.

2.2. Sales Funnel

Dubinski's framework covers (1) prospecting, (2) preapproach, (3) approach, (4) presentation, (5) handling objections, (6) close, and (7) follow-up (Dubinsky 1980). Even though it is a traditional framework and does not cover in much detail the stages after the opportunity is closed, it was considered applicable to most B2B sales situations (Paschen, Wilson and Ferreira 2020; Sheth and Sharma 2008; Syam and Sharma 2018).

According to Van Der Kooij (2019), it is stated that "traditional B2B marketing and sales frameworks (...) do not achieve the desired results in SaaS businesses" and this sentence not only applies to Software as a Service (SaaS) but also to other recurring revenue models, such as Platform as a Service (PaaS), which is the case of OutSystems. It is recommended that sales should be driven by impact throughout the sales process, ensuring that the source of revenue is not lost during the process, i.e. before a profit is made. Impact can be perceived as either emotional or rational. Rational impact benefits the company first, as it is usually related with monetary values, while the emotional impact often affects the individual first, as it improves, for example, the user experience. This methodology is depicted on the bow-tie sales model shown in Figure 1.

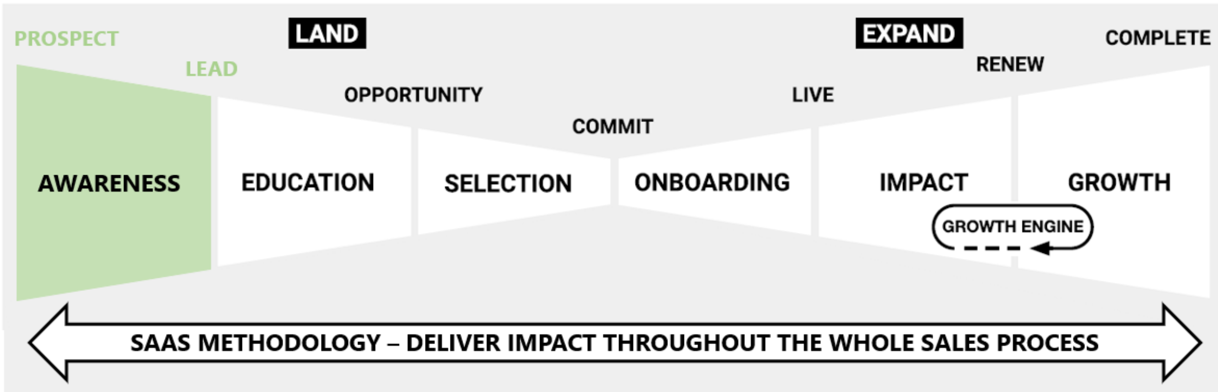


Figure 1. SaaS Growth Methodology (in green the focus of this thesis). Adapted from Van Der Kooij (2019)

Since the goal of this project is to better allocate the efforts made to acquire a new customer, the analysis will be focused on the Land part of the model, more specifically in the Prospect stage, where the impact perceived should be approximately assessed by the demographic and technographic characteristics of the companies. This stage is common to both frameworks and will, therefore, be used interchangeably.

Prospecting consists of finding potential customers out of the available pool of companies. In theory, the pool can contain every company that is not a customer yet (D'Haen and Van Den Poel 2013). The prospecting task is similar to the segmentation task in marketing, as sales and marketing go hand in hand with each other (Syam and Sharma 2018). Prospect lists are built based on third party lists and directories, online sources, networking, and promotional activities (Buttle and Maklan 2015). After the list is created, companies try to narrow it down to the ones that are more likely to buy the product/service. This is called lead qualification. Due to the substantial human resources needed, errors in qualifying leads result in wasted assets, a suboptimal structure for applying marketing and promotion efforts and, hence, a loss in sales revenue (Monat 2011; Järvinen and Taiminen 2016). This is due to the difficulty of firms to find the characteristics that actually classify a customer as "high-quality" and, even after they do, they usually can only acquire that information after the first contact is established (Paschen, Wilson and Ferreira 2020).

As depicted in Figure 2, the objective is to convert more leads into paying customers. By improving the quality of the leads generated, the end of the funnel can be broadened, increasing the win rate of opportunities given by OutSystems. The win rate represents the number of opportunities won over the total number of opportunities given. The ultimate goal is to increase this conversion rate to 100% without reducing the amount of prospects that are defined as qualified leads.

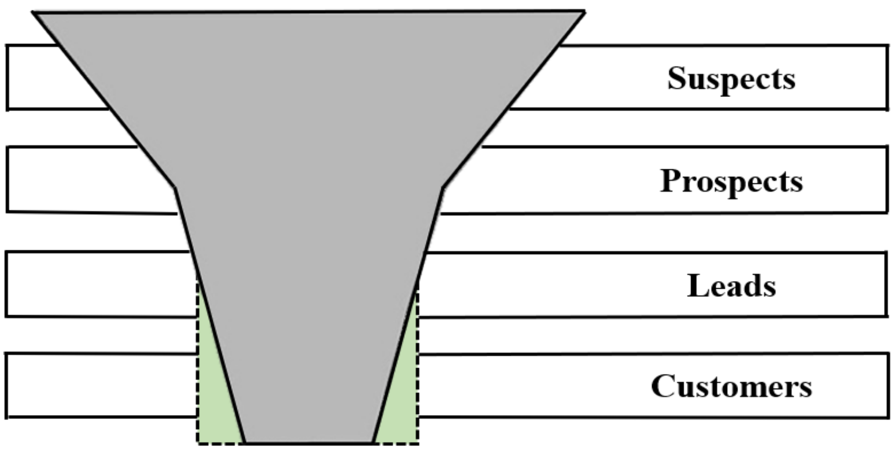


Figure 2. Old (in gray) versus Optimal Suggested Sales Funnel (in green). Adapted from D'Haen and Van Den Poel 2013

2.3. Availability of Data and Machine Learning

According to Paschen, Wilson and Ferreira (2020), the emergence of AI gave companies the ability to stop relying so much on their sales force and support teams to undertake the above-stated activities. AI tools are able to extract information from both structured and unstructured formats, giving firms plenty of information to segment and target their customers (Paschen, Wilson and Ferreira 2020). In conjunction with the power to analyze large amounts of data, AI is able to revolutionize the first stage of the sales process with progressively better results and without the need for human interaction to readjust the rules (Syam and Sharma 2018). Furthermore, machine learning (ML) procedures are used to effectively predict outcomes based on past experience, allowing for continual improvement (Syam and Sharma 2018).

Models are already being used to predict the probability of converting a prospect into an actual consumer, as it is the case of Dell, that analyzes consumer behavior and patterns to predict if they will eventually purchase a product (King 2012). Similarly, Rezazadeh (2020) proposed an end-to-end workflow to forecast the outcome of sales of a consultancy firm, using an ensemble of LightGBM and XGBoost models. Since salespeople had to fill in their estimate for the probability of success for each project, it was possible to compare the user-entered probabilities with the results obtained from ML. The presented model reported an accuracy 18 percentage points higher than the user-entered probabilities during an experiment that lasted for three months, with an accuracy of 85%. These kinds of achievements are starting to shift the role that sales professionals have in the early stages of the sales process (Paschen, Wilson and Ferreira 2020).

2.4. The new role of Sales Professionals in the Prospecting Stage

Sales professionals instead of doing the whole process on their own, can now leave the first “data review” to the machines and focus on delivering their value at the next stage: the final decision regarding the qualification. Human judgment is still necessary to interpret and filter the AI-generated information (Paschen, Wilson and Ferreira 2020), as their intuition and experience are able to detect inconsistencies both in the models’ output and on the data quality. With the information they have available, humans are capable of defining the appropriate course of action, outperforming AI with their experience, background, and skills. Hence, the use of AI might improve processes, but should not be used as the last source of information, as humans are still important in the decision-making process (Paschen, Wilson and Ferreira 2020).

2.5. Management Challenges

With the introduction of changes, managers should expect challenges. The transition into an AI-enabled framework makes employees afraid of losing their jobs, making them resistant to change. In this fashion, leaders should embrace practices that support change management, such as creating a coalition and developing and communicating their vision in order to empower their employees (Seijts and Gandz 2018). Establishing a framework where employees feel they still have the upper hand in regard to decision making, can smooth the whole process of change (Paschen, Wilson and Ferreira 2020).

3. Background and Data

3.1. OutSystems

OutSystems is a software company founded in 2001 in Lisbon, Portugal. It provides a low-code platform that helps developers deliver applications quickly and efficiently, by providing a simple and visual interface (OutSystems, n.d.). OutSystems was considered a leader in Gartner's Magic Quadrant for Enterprise Low-Code Application Platforms 2020. Its efforts are focused on enterprise application development, being an already established brand in Europe and North America and gaining traction in Asia-Pacific (APAC). Their customers are mainly large enterprises across service, product, and public sector organizations (Gartner 2020).

OutSystems delivers a Platform as a Service (PaaS), which is "a complete development and deployment environment in the cloud" where you pay-as-you-go (recurring revenue) as per Microsoft (n.d.). PaaS is designed in order to reduce the burden of managing software licenses, supporting infrastructure and development tools (Microsoft, n.d.).

3.2. Data

All the data used throughout the project was provided by OutSystems. Part of it was acquired from an external source. The third-party collects companies' key information, such as the revenue and the number of employees, as well as specific hardware and software technologies used by those companies, uncovering their technographic characteristics.

4. Methodology

This section is divided into 3 subsections: Clustering, Record Linkage, and Propensity to Buy. Each section tackles a different problem and all together form a pipeline that aims to get OutSystems closer to its goal: a better allocation of their resources. Figure 3 presents a summary of the Methodology section.

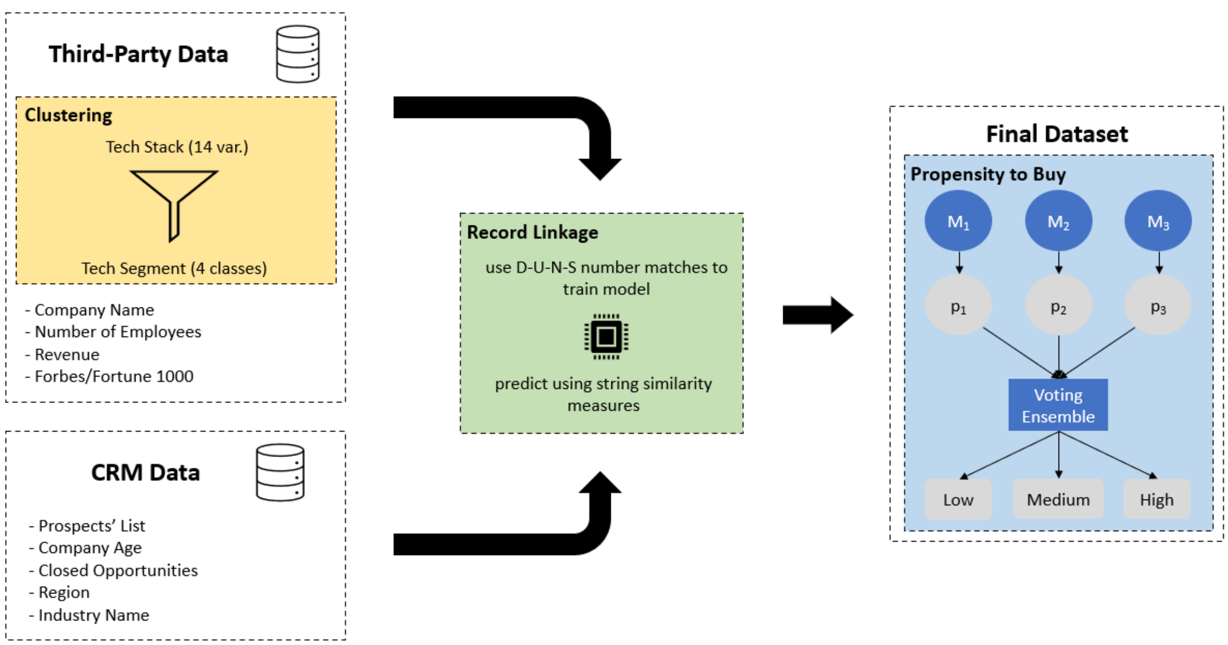


Figure 3. Summary of the methodology section. Each subsection is represented by a different color

4.1. Tech Segmentation

In order to perform the segmentation based on the technographic characteristics of each company, the third-party database was used. Since the data provided did not meet the current needs of the business, domain expertise from within OutSystems was collected and a final list of variables was computed. Most technologies present in the data were binarized, being assigned a 1 if the company under analysis used any of the technologies belonging to that category and a 0 otherwise. The final list consisted of 14 different categories: each category associated with a set of technologies.

Melitski, Gavin and Gavin (2010) found a relationship between the individuals' perception of the company culture and their willingness to adopt new technologies. Since culture and technological adoption are not observable factors, a latent class model was used to estimate them. Since latent class analysis, unlike factor analysis, needs an educated estimation of the number of classes (Bartholomew *et al* 2008), 4 classes were predicted: *Strollers* (no technological adoption), *Walkers* (slow adoption), *Runners* (fast adoption), and *Sprinters* (state-of-the-art adoption). The technological categories were then fed into a Latent Class Analysis clustering algorithm (Linzer and Lewis 2014). The distribution of each class is shown in Appendix A1.

4.2. Record Linkage

OutSystems already had in place a list of target accounts built on the company's Customer Relationship Management system (CRM). In order to be able to use the recently acquired data with the already existing ones, the need to merge the databases and match the companies present in both emerged. The package chosen to tackle the problem found in this section was the *Python Record Linkage Toolkit*, maintained by De Bruin (2019).

After looking at the data, common variables were found. However, the only variables able to give exact matches were the ones containing the company names and D-U-N-S numbers. The

D-U-N-S number, which stands for Data Universal Numbering System, is a unique identifier for businesses, maintained by Dun & Bradstreet, and is used by businesses to tackle this kind of issue (Dun & Bradstreet, n.d.). As so, it worked as a good starting point for the linkage problem. Variables containing information about the country and the state were a good source of complementary information to make a final decision, even though the state only had information regarding the United States, Canada, and Australia. The problem is that in the CRM dataset, there are 6,916 missing D-U-N-S numbers and there is no way to certainly know how many companies are actually represented in both datasets. Just by matching on the exact D-U-N-S number, 26,152 records were linked. Yet, by looking at the data, there were reasons to believe that this number did not represent the actual intersection count. To solve this problem, a model that analyses similarity was used. Since some records were already matched through the D-U-N-S, the problem was transformed into a Supervised Machine Learning problem.

Given the high dimensionality of both datasets, a similarity matrix containing full indexing between both would be computationally unmanageable (63k * 115k), given the constraints faced by the use of a local machine. To overcome this challenge, a Sorted Neighborhood indexing on the Company Name column was used. The Sorted Neighborhood merges both columns, sorts them alphabetically and then, for each record, it creates possible matches given a fixed window. For a window of 3, it checks for records of the other dataset one place to the right and one place to the left, being possible to find 0, 1, or 2 matches, depending on the sorting results. For this project, after several trials, a window of 19 was defined as a good balance between the number of possible matches and actual matches found. Furthermore, to increase the coverage of the indexing, an exact block match between the first word of the company name of one dataset with the second of the other and vice versa was also used, in order to avoid missing companies with their names in a different order. The indexing resulted in a total number of 1,982,883 possible pairs, with approximately 95% of the total number of D-U-N-S matches being caught by the indexing.

Before computing the similarity matrix, some preprocessing on the company names was done, removing terms that indicate the organization type (e.g. “Ltd.”), information inside brackets, and punctuation, and standardizing the letter case. To do so, the *cleanco* package, maintained by Savolainen (2020), was used. The columns used in the matrix were then computed and represented the exact match of country, state, and D-U-N-S (binary variables) and the string similarity between company names. As previous papers suggest, the performance of each string similarity algorithm is task-dependent (Santos, Murrieta-Flores, and Martins 2018). As so, several trials were run with different measures, using cosine similarity, qgram, longest common subsequence, Damerau-Levenshtein and Jaro-Winkler as candidates. Given the high complexity of computation inherent to most measures, the last two were the ones selected given their performance, both in terms of time and results. Adding more measures was exponentially increasing the computation time while marginally improving the results. The selected measures range between 0 and 1, being 1 an exact match and 0 the opposite. While the Damerau-Levenshtein measures the minimum number of insertions, deletions, substitutions, and transpositions needed to turn one string into the other (Damerau 1964; Levenshtein 1966), the Jaro-Winkler measures the number of matching characters that are not too far from each other, depending on the length of the string, and gives a higher weight for the matching characters found at the beginning of the string (Jaro 1989; Winkler 1990).

To train the model, a subset of the data was created with all the possible pairs that contained CRM IDs already matched by the D-U-N-S. The resulting table had 798,720 rows, with 26,152 positive matches. All the negative matches present in the data were a result of the indexing previously done.

The subset was split into train (70%), validation (20%) and test (10%). A Logistic Regression was then fit into the training data. After analyzing the results, it was noticeable that the algorithm was wrongly classifying similar, yet different, possible customers as positive matches (e.g. County of San Francisco and County of San Diego). Since the main goal of this task is to get the positive matches as accurate as possible, the chosen metric was precision. Precision represents the number of true positives over the total number of positives predicted by the model. A model with a precision of 1 means that every positive prediction is indeed an actual positive.

To increase precision and avoid cases similar to the example mentioned above, two thresholds were defined. The first was the minimum score accepted for a match to be considered positive and the second was the difference between the top 2 match scores of each CRM account. This way it is possible to avoid matches in which the model cannot decide upon, even though they might have a high score when assessed individually. These thresholds were defined on the validation set, running a grid search to maximize precision. The first threshold was capped at 0.25 and the second at 0.4, yielding a precision score of 99.64% on the validation set and 99.63% on the test set. The usage of the algorithm increased the number of matches from 26,152 to 28,404, representing an 8.61% increase.

4.3. Propensity to Buy

To build the model that measures the propensity to buy of each customer, a dataset that contains the opportunities given to each potential new customer in the past was merged. The opportunities were filtered to only show those that were already closed, either won or lost, and only contained companies operating in countries considered of interest by OutSystems. Even though the third-party data was extracted recently, the date filter was set from the beginning of 2018 until the end of October 2020, since it was also important to keep a representative amount of data available for the model. Out of the 28,404 matched companies, only 3,045 had at least one closed opportunity, given the constraints stated above.

Exploratory Data Analysis was then performed to assess if there were differences in win rate inside each variable and the results are shown in Appendix A2. The final list consists of 6 different variables, 3 categorical and 3 numeric. Depending on the model, each variable was encoded differently, in order to make the most out of each model's characteristics. The encoding can be found in Appendix A3.

The dataset was split into the train (80%) and test (20%), keeping the same proportion of positive class records in both sets. Cross-validation was used throughout the whole process in order to avoid the need to further split the data and, therefore, decrease the sample size used both for training and testing. Out of the total 3,045 closed opportunities, 609 were used for testing.

4.3.1. Models

As the baseline for the machine learning models, a Logistic Regression was used. The remainder of the models used were boosting methods, as they provide tools to handle both categorical data and missing values, while achieving state-of-the-art results both in terms of performance and accuracy, when compared for example with traditional and bagging methods (Banfield *et al* 2007). The three models used were XGBoost, LightGBM, and CatBoost.

Tree boosting has shown state-of-the-art results in several classification benchmarks (Li 2010). The models start with a weak learner and new weak learners are added and trained with respect to the error of the weak learners fitted so far (Natekin and Knoll 2013). By fitting new learners, the model provides more accurate predictions of the response variable. Tree boosting is highly flexible given its ability to implement a rich variety of loss functions, depending on the problem at hand (Natekin and Knoll 2013).

Chen and Guestrin (2016) presented the XGBoost as a scalable machine learning system for tree boosting, widely used in competitions given its flexibility and results. Ke *et al.* (2017) came up with LightGBM since the efficiency and scalability of XGBoost were still seen as unsatisfactory. The main difference between both is that while XGBoost grows horizontally, the LightGBM grows vertically (leaf-wise), making it more effective at handling datasets with a higher dimensionality (Rezazadeh 2020). CatBoost was created in order to address target leakage, with the introduction of an innovative algorithm to process categorical features, outperforming XGBoost and LightGBM on diverse machine learning tasks (Prokhorenkova *et al* 2018).

4.3.2. Hyperparameter Fine-Tuning

Similar to the model choice, a crucial part of any ML project is the fine-tuning of the models' hyperparameters. Hyperparameter settings often establish the difference between just decent and state-of-the-art results (Hutter, Hoos and Leyton-Brown 2014). Unlike model parameters that the model knows how to tweak and are learned from data, hyperparameters must be set before the training phase as they are high-level parameters that affect the way that same data is learned (Zheng 2015). However, for models that have a large number of hyperparameters, running a traditional optimization, such as grid search, would be inefficient (Hutter, Hoos and Leyton-Brown 2014). As so, the chosen method to fine-tune the hyperparameters was the Bayesian search, using the *scikit-optimize* package (Head *et al* 2020). Bayesian optimization tries to find the optimum of a complex function in the least number of iterations, using an acquisition function to define which sample should be tried next. The acquisition function consists of a "cheaper" function that is easier to optimize than the initial function, in this case, the black-box function given by the models (Nguyen 2019). The package offers Bayesian search combined with cross validation, giving a tool to deal with overfitting.

4.3.3. Metrics

The metric used for the optimization of the model was the Area Under the Receiver Operating Characteristic Curve (ROC AUC). Even though the class distribution of the dataset used is not highly imbalanced (approximately 1:6 ratio), this ratio is likely to change in the future, with the eventual saturation of the cloud industry. ROC curves have the advantage that they are insensitive to changes in class distribution because if the proportion of positive to negative instances changes, the ROC curve is not affected (Fawcett 2006). Other performance measures, such as the F-score, precision, and accuracy, are intrinsically sensitive to class skews (Fawcett 2006). Furthermore, accuracy implicitly assumes that the distribution is constant and relatively balanced while giving the same weight for different types of costs - a false positive costs the same as a false negative error - which is rarely the case in real life problems (Provost and Fawcett 1997). The AUC reduces the ROC performance to a single scalar value, is not dependent on a decision threshold, does not change with class skews, and gives a sense of how well separated the negative and positive classes are, penalizing models that classify as random or allocate every observation to the same class (Bradley 1997).

4.3.4. Probability Calibration

Since the final goal of the model was to be combined with different sources of information to come up with a final decision in regard to which customers should OutSystems go after, it is important to have probabilities estimates that actually represent what happens in reality (Zadrozny and Elkan 2002). Unlike models like the Logistic Regression and bagged decision trees that already produce such probabilities, boosted trees need to be calibrated. After calibration, they produce results better than the ones stated above (Niculescu-Mizil and Caruana 2005). As so, after searching for the best hyperparameters, the predicted probabilities of the best estimator of each model were calibrated in order to predict well-calibrated probabilities. Instead of using the Isotonic Regression, presented by Zadrozny and Elkan (2002), which can correct any monotonic distortion but is prone to overfitting when data is scarce, the Platt Scaling was used (Platt 1999). Even though cross-validation was applied, empirical evidence shows that at least 1,000 records are needed in the calibration set for the Isotonic method yield as good or better performance than the Platt Scaling (Niculescu-Mizil and Caruana 2005). That amount of data was not available and hence the decision.

4.3.5. Explainability

Even though explainability was not a requirement for the deployment of the model, as Lundberg and Lee (2017) stated in their paper, interpretation "engenders appropriate user trust, provides insight into how a model may be improved, and supports understanding of the process being modeled", playing a vital role in the adoption of the new computer-enabled inputs in the decision process.

The feature importance was assessed using the SHAP (SHapley Additive exPlanation) values. SHAP was introduced by Lundberg and Lee (2017) in an attempt to explain the predictions of any machine learning model. It uses a game-theoretic approach to explain the output, computing what makes an individual prediction deviate from the mean prediction value, assuring consistency, local accuracy, and uniqueness. Lundberg, Erion and Lee (2018) also presented the Tree SHAP, an extension of the SHAP for tree ensembles methods, which retains the previous properties but is computed at a higher speed. In Figure 4, the *summary plot* for the most important features is presented. Each dot represents a single observation and the features are shown by order of importance. The full figure can be found in Appendix A4.1. As an illustrative example, it is possible to see that the higher the tech segment class, the lower is the chance of a company to become a customer, according to the model explainability, which means that OutSystems has a higher chance of acquiring a customer classified as *Stroller* (low technological level) than for example a company classified as *Sprinter*.

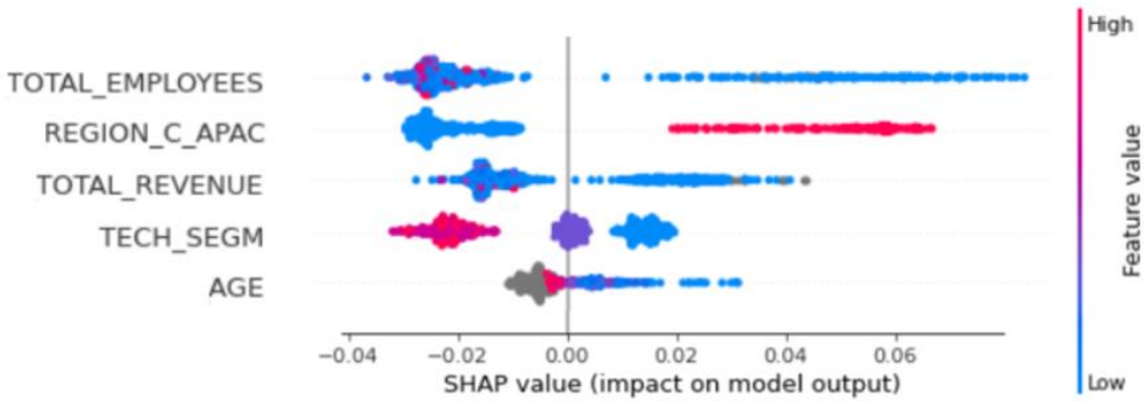


Figure 4. Top 5 Feature Importance (Tree SHAP)

4.3.6. Decision Boundaries

Since the company did not want to provide raw probabilities estimates to their sales force, decision boundaries were defined. Instead of applying just one threshold to classify each prospect as a possible win or lose, two thresholds were defined. Hence, the model produced three different classes: Low, Medium and High propensity to buy. Two different approaches were used to define them. The first defined the thresholds on the whole probability distribution of the training set, using 3-quantiles. The second used a *post hoc* analysis: by looking at the importance of each variable in the model output, presented above in Figure 4, the Tech Segment and the Region were the chosen variables to further split the distributions. Two different thresholds were assigned for each of the 12 combinations, using again the 3-quantiles of each distribution. The reason behind the usage of quantiles to define the thresholds was to ensure that a similar number of accounts fell within each class, considering the distribution of the results and not some arbitrary numbers.

4.3.7. Voting Ensemble

In order to balance out the individual classification errors of the various classification models, a voting ensemble was used as the final model to check the win rates across classes. Since taking the mode of the class labels (hard voting), with the same number of classifiers and labels, could result in ties, the average of the predicted probabilities was used (soft voting). The method used to define the class labels' thresholds was the same.

4.3.8. Baseline Comparison

To compare the performance of the model against OutSystems' old method of choosing accounts, the named accounts of 2019 and 2020 were used. Named accounts are those chosen by the sales force to pursue during a certain year. Even though these accounts are not selected solely on their propensity to buy and include factors that are not considered by the model, it is important for the company to understand if the model output would help in those cases. Since the analysis was conducted only on the test set, it introduces the possibility of win rates not reflecting the reality of that year, as the timeframe of the opportunities was not considered when splitting the data into training and test.

5. <u>Results, Discussion, Limitations and Future Work</u>

5.1. Results

The results obtained before the decision boundaries were applied can be found in Table 1. CatBoost had the highest AUC in the training, cross-validation, and test set, with a score of 0.723, 0.664 and 0.666, respectively. LightGBM was second, with an AUC score of 0.650 in the test set. XGBoost had the worst performance compared with the rest of the boosting methods. The Voting Ensemble showed an AUC of 0.653 on the test set, which means that the model has a 65.3% chance of ranking a random positive instance higher than a random negative. A score of 0.5 would mean that the algorithm achieves the same results as ranking by chance and a score of 1 would mean that the algorithm is able to perfectly rank the observations. All models performed better in terms of AUC than the baseline (Logistic Regression) by at least 3.1 percentage points. Even though the Voting Ensemble was not the best performer, it is expected to yield better results in the future, as it gives the possibility of adding more models, giving flexibility to further adjust the end-to-end pipeline.

Model	AUC Cross-Validation	AUC Training Set	AUC Test Set
Logistic Regression		0.655	0.598
XGBoost	0.651 ± 0.029	0.686	0.629
LightGBM	0.658 ± 0.024	0.687	0.650
CatBoost	0.664 ± 0.026	0.723	0.666
Voting Ensemble		0.709	0.653

Table 1. Classification results

To better understand the capability of the model to rank observations, the win rates by class predicted by the Voting Ensemble for the test set are shown in Table 2. The thresholds and probability distributions can be found in Appendix A6. By looking at Table 2 it is possible to see that the class labeled High is the one with the highest win rate (27.14%). Regarding the other two classes, the model does not seem to do a good job at differentiating them. Using the overall thresholds, it is even possible to see that the win rate of the Low segment (11.17%) is higher than the win rate of the Medium (10.42%), even though by a low margin. Using the thresholds by Region and Tech this situation is corrected: Medium has a win rate of 13.79% while Low has 10.53%. An important aspect is that the number of opportunities won (in absolute terms) follows the order of the classes, leaving the Low segment with only 20% of the total amount of won opportunities in the test set (100).

Class	Overall Thresholds			Thresholds by Region and Tech		
	Win rate	**# Opps**	**# Wins**	**Win rate**	**# Opps**	**# Wins**
High	27.14%	210	57	24.07%	216	52
Medium	10.42%	211	22	13.79%	203	28
Low	11.17%	188	21	10.53%	190	20

Table 2. Win rate by class (Test Set)

Regarding the performance of the model on the named accounts of past years, the results are shown in Table 3. The table shows how the model with varying thresholds did in classifying the accounts chosen by the sales force (test set). Regarding 2019, two thirds of the accounts won were in the High segment, with only 2 won accounts out of 18 total opportunities classified as Low. Here it is possible to see again some lack of distinction power between the Low and Medium class. In 2020, the model seems to have performed well: the High segment had the highest win rate (7.69%) and the Low had the lowest (4.17%).

Class	Named Accounts in 2019			Named Accounts in 2020		
	Win rate	**# Opps**	**# Wins**	**Win rate**	**# Opps**	**# Wins**
High	33.33%	24	8	7.69%	78	6
Medium	9.52%	21	2	6.67%	60	4
Low	11.11%	18	2	4.17%	72	3

Table 3. Win rate by class (Test Set)

The class distribution for the prospects that have not yet been given an opportunity is shown in Table 4. The High segment has the most accounts, with 39.43% of the records falling in that section.

Class	# of Accounts	% of Total
High	9,961	39.43%
Medium	9,819	38.88%
Low	5,477	21.69%

Table 4. Classification of accounts without closed opportunities

5.2. Discussion

By looking at the AUC results and the ROC curves shown in Appendix A5, it is possible to understand that the main challenge at solving this problem is not on the model selection but on the overall data quality and quantity, as every model performs similarly. With a limited sample size, features generated after several steps prone to errors (e.g. automatic data extraction and record linkage) and external factors affecting customers' decisions, the output of the model is somewhat limited. This problem is reflected on how the model predicted the win rate for each class. For lower probabilities, the model is not able to differentiate prospects, likely because the information it has for customers similar to those is very limited. The predicted probabilities of the model, after calibration, do not go much over 50%, reinforcing that the information provided is not enough for the model to do a confident prediction.

In comparison with Rezazadeh (2020) final model, the results are underwhelming, with a lower AUC by approximately 17 percentage points. However, this difference can be in part explained by the differences between the two projects. Rezazadeh (2020) used information after the first contact was already established, using variables such as the project duration, the contract value and the sales engagement manager, something that OutSystems is not able to infer beforehand. The amount of data used was also about 8 times larger than the data available at OutSystems. All in all, although similar, the challenges faced by both projects are not the same, which makes a proper comparison hard to do.

Unfortunately, unlike the above project that had user-entered probabilities, OutSystems did not have a solid baseline to compare with. The analysis presented regarding the named accounts was done as an approximation of the current practices at OutSystems, in an attempt to check the model effectiveness. However, since only the test set was used to not bias the results towards what the model was trained on, the win rates presented for each year do not reflect the reality at the company. What we can extrapolate from that analysis, with some degree of uncertainty given the small sample size, is that the model is properly ranking the accounts selected.

Since OutSystems' goal is to aggregate the output of the model with other sources of information, such as behavioral data and the number of similar current customers, this extra information will fill part of the gaps from the model, making the final ranking of the accounts more reliable. Then, salespeople can have a look at the final output and make a final decision. This is the stage where they can show they are still an important part of the decision-making process, as Paschen, Wilson and Ferreira (2020) suggested. The advantage of this approach is that now they will be able to make a more informed decision, based on actual business results and not solely on their gut feeling. Given that this process does not exclude them from the prospecting stage, the management challenges should be minimal, as the sales personnel can slowly adapt to this new era and see the results by themselves.

Overall, the model seems to give a good insight into the salespeople of the propensity to buy of each account, even though there is still a lot of room for improvement. That room was shown by the AUC score, as it did not go over 67% in any of the models used.

5.3. Limitations

Since part of the data was acquired from a third party, it was hard to ensure data quality. Features such as the number of employees and revenue are often undisclosed (private companies) and the forecasts may not represent the reality lived in the company. Without data quality the results of the models are biased, and the conclusions taken regarding feature importance do not reflect their actual impact in the real world.

Since OutSystems works in a B2B environment and sells mainly to large enterprises, the amount of data regarding their sales is scarce. Furthermore, the data used regarding the revenue, the number of employees and the tech stack, was specific to one moment in time and did not correspond to the exact moment an opportunity was deemed as closed.

Due to time constraints, the results of the model have not been tested in the sales process, aside from the recurrent machine learning practice of testing on a previous unseen dataset and comparison with the business practice of the past (named accounts).

5.4. Future Work

In the upcoming months, the model output will serve as an indicator of the propensity to buy and will help ranking the accounts, but the total control over the decisions made is given to the sales personnel, even if they go against the model predictions. A good practice to check on the model effectiveness would be to split the sales force into two separate groups. The first would apply the current practice and the second would choose solely accounts that the model predicted as High (focus group). By the end of the year the performance of both groups would be compared, and the actual value and accuracy of the model could be properly assessed.

To further enhance engagement from the side of the sales personnel, an implementation of the SHAP algorithm on the CRM account page of each possible customer can be developed. It would show a *force plot* for that specific account, depicting how each characteristic influenced the final decision of the model. An example of a *force plot* can be found in Appendix A4.4. This step would require further analysis to assess if the explainability would disseminate acceptance or increase resistance, as employees might start spending more time challenging the model than actually using its output.

6. <u>Conclusion</u>

Companies have historically based their account selection mechanism on the gut feeling of their sales team. With the emergence of data and business analytics, this trend has changed and companies are now realizing the power of AI, namely of ML, to forecast sales' outcomes. However, with new opportunities come new challenges. Those include the need for data quality and quantity (the more, the better) and the ability to link records from different sources and properly manage change inside the company. In this thesis, an approach to record linkage was presented with an increase of 8.61% on the number of matched records. It was also presented an ensemble model of tree boosting methods, with an AUC score of 0.653 on the test set, to help sales professionals assess the propensity to buy for accounts not yet addressed by OutSystems. After the definition of the decision boundaries, the model seems to do a reasonable job at ranking the accounts. Even though the model suffers from a lack of data quality and quantity, this situation is likely to change as OutSystems keeps growing and expanding their customer base. Since the model output is to be used jointly with other information and the final choice is dependent on human judgment, the actual effectiveness of the model and management challenges are hard to assess at this stage. Nevertheless, as humans still have the upper hand, the management hurdles are expected to be minimal.

References

Banfield, Robert E., Lawrence O. Hall, Kevin W. Bowyer, and W. P. Kegelmeyer. 2007. "A Comparison of Decision Tree Ensemble Creation Techniques." IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1): 173–80. https://doi.org/10.1109/TPAMI.2007.250609.

Bartholomew, David J, Fiona Steele, Irini Moustaki, and Jane I Galbraith. 2008. Analysis of Multivariate Social Science Data.

Bradley, Andrew P. 1997. "The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms." Pattern Recognition 30 (7): 1145–59. https://doi.org/10.1016/S0031-3203(96)00142-2.

Burgess, Bev, and Dave Munn. 2017. A Practitioner's Guide to Account Based Marketing: Accelerating Growth in Strategic Accounts. New York: Kogan Page.

Buttle, Francis, and Stan Maklan. 2015. Customer Relationship Management, Concept and Technologies, Third Edition.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–94. New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785.

Colter, Tim, Mingyu Guan, Mitra Mahdavian, Sohail Razzaq, and Jeremy D Schneider. 2018. "What the Future Science of B2B Sales Growth Looks Like." McKinsey & Company. https://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/what-the-future-science-of-b2b-sales-growth-looks-like#.

Damerau, Fred J. 1964. "A Technique for Computer Detection and Correction of Spelling Errors." Communications of the ACM 7 (3): 171–76. https://doi.org/10.1145/363958.363994.

De Bruin, J. 2019. "Python Record Linkage Toolkit: A Toolkit for Record Linkage and Duplicate Detection in Python." Zenodo. https://doi.org/10.5281/zenodo.3559043.

Dewet, Stephanie, and Jiafan Ou. 2019. "Finding Users Who Act Alike: Transfer Learning for Expanding Advertiser Audiences." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2251–59. https://doi.org/10.1145/3292500.3330714.

D’Haen, Jeroen, and Dirk Van Den Poel. 2013. “Model-Supported Business-to-Business Prospect Prediction Based on an Iterative Customer Acquisition Framework.” Industrial Marketing Management 42 (4): 544–51. https://doi.org/10.1016/j.indmarman.2013.03.006.

Dubinsky, Alan J. 1980. “Analytic Study the Personal Selling” 1 (1): 26–33.

Dun & Bradstreet. n.d. “D-U-N-S Number.” https://www.dnb.com/duns-number.html.

Fawcett, Tom. 2006. “An Introduction to ROC Analysis.” Pattern Recognition Letters 27 (8): 861–74. https://doi.org/10.1016/j.patrec.2005.10.010.

Gartner. 2020. “Magic Quadrant for Enterprise Low-Code Application Platforms.” https://www.gartner.com/doc/reprints?id=1-24BC744E&ct=201005&st=sb.

Head, Tim, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe, and Iaroslav Shcherbatyi. 2020. “Scikit-optimize/scikit-optimize”. Zenodo. doi:10.5281/zenodo.4014775.

Hutter, Frank, Hoiger Hoos, and Kevin Leyton-Brown. 2014. “An Efficient Approach for Assessing Hyperparameter Importance.” 31st International Conference on Machine Learning, ICML 2014 2: 1130–44.

Ingram, Thomas N. 2004. “Future Themes in Sales and Sales Management: Complexity, Collaboration, and Accountability.” Journal of Marketing Theory and Practice 12 (4): 18–28. https://doi.org/10.1080/10696679.2004.11658528.

Jaro, Matthew A. 1989. “Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida.” Journal of the American Statistical Association 84 (406): 414–20. https://doi.org/10.1080/01621459.1989.10478785.

Järvinen, Joel, and Heini Taiminen. 2016. “Harnessing Marketing Automation for B2B Content Marketing.” Industrial Marketing Management 54: 164–75. https://doi.org/10.1016/j.indmarman.2015.07.002.

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie Yan Liu. 2017. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree.” Advances in Neural Information Processing Systems 2017-Decem (Nips): 3147–55.

King, Rachel. 2012. “How Dell Predicts Which Customers Are Most Likely to Buy.” The Wall Street Journal. 2012. https://blogs.wsj.com/cio/2012/12/05/how-dell-predicts-which-customers-are-most-likely-to-buy/.

Levenshtein, Vladimir I. 1966. “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals.” Soviet Physics Doklady 10 (8): 707–10.

Li, Ping. 2010. “Robust Logitboost and Adaptive Base Class (ABC) Logitboost.” Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, UAI 2010, no. 2: 302–11.

Linzer, Drew, and Jeffrey Lewis. 2014. “PoLCA.”

Lundberg, Scott M., Gabriel G. Erion, and Su In Lee. 2018. “Consistent Individualized Feature Attribution for Tree Ensembles.” ArXiv, no. 2.

Lundberg, Scott M., and Su In Lee. 2017. “A Unified Approach to Interpreting Model Predictions.” Advances in Neural Information Processing Systems 2017-Decem (Section 2): 4766–75.

Melitski, James, David Gavin, and Joanne Gavin. 2010. “Technology Adoption and Organizational Culture in Public Organizations.” International Journal of Organization Theory & Behavior 13 (4): 546–68. https://doi.org/10.1108/ijotb-13-04-2010-b005.

Microsoft. n.d. “What Is PaaS?” https://azure.microsoft.com/en-us/overview/what-is-paas/.

Monat, Jamie P. 2011. “Industrial Sales Lead Conversion Modeling.” Marketing Intelligence and Planning 29 (2): 178–94. https://doi.org/10.1108/02634501111117610.

Natekin, Alexey, and Alois Knoll. 2013. “Gradient Boosting Machines, a Tutorial.” Frontiers in Neurorobotics 7 (DEC). https://doi.org/10.3389/fnbot.2013.00021.

Nguyen, Vu. 2019. “Bayesian Optimization for Accelerating Hyper-Parameter Tuning.” Proceedings - IEEE 2nd International Conference on Artificial Intelligence and Knowledge Engineering, AIKE 2019, 302–5. https://doi.org/10.1109/AIKE.2019.00060.

Niculescu-Mizil, Alexandru, and Rich Caruana. 2005. “Predicting Good Probabilities with Supervised Learning.” ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning, no. 1999: 625–32. https://doi.org/10.1145/1102351.1102430.

OutSystems. n.d. “OutSystems Platform.” https://www.outsystems.com/platform/.

Paschen, Jeannette, Matthew Wilson, and João J. Ferreira. 2020. “Collaborative Intelligence: How Human and Artificial Intelligence Create Value along the B2B Sales Funnel.” Business Horizons 63 (3): 403–14. https://doi.org/10.1016/j.bushor.2020.01.003.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” Journal of Machine Learning Research.

Platt, John C. 1999. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.” Microsoft Research.

Pradhan, S. N. 1975. “Aggression and Central Neurotransmitters.” International Review of Neurobiology. https://doi.org/10.1016/S0074-7742(08)60036-7.

Prokhorenkova, Liudmila, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. “Catboost: Unbiased Boosting with Categorical Features.” Advances in Neural Information Processing Systems 2018-Decem (Section 4): 6638–48.

Provost, Foster, and Tom Fawcett. 1997. “Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions.” KDD-97 Proceedings, 43–48. http://www.aaai.org/Papers/KDD/1997/KDD97-007.pdf.

Rezazadeh, Alireza. 2020. “A Generalized Flow for B2B Sales Predictive Modeling: An Azure Machine Learning Approach.” ArXiv, no. Ml: 267–83. https://doi.org/10.3390/forecast2030015.

Santos, Rui, Patricia Murrieta-Flores, and Bruno Martins. 2018. “Learning to Combine Multiple String Similarity Metrics for Effective Toponym Matching.” International Journal of Digital Earth 11 (9): 913–38. https://doi.org/10.1080/17538947.2017.1371253.

Savolainen, Petri. 2020. “Cleanco.” https://github.com/psolin/cleanco.

Seijts, Gerard H., and Jeffrey Gandz. 2018. “Transformational Change and Leader Character.” Business Horizons 61 (2): 239–49. https://doi.org/10.1016/j.bushor.2017.11.005.

Sheth, Jagdish N., and Arun Sharma. 2008. “The Impact of the Product to Service Shift in Industrial Markets and the Evolution of the Sales Organization.” Industrial Marketing Management 37 (3): 260–69. https://doi.org/10.1016/j.indmarman.2007.07.010.

Singh, Jagdip, Karen Flaherty, Ravipreet S. Sohi, Dawn Deeter-Schmelz, Johannes Habel, Kenneth Le Meunier-FitzHugh, Avinash Malshe, Ryan Mullins, and Vincent Onyemah. 2019. “Sales Profession and Professionals in the Age of Digitization and Artificial Intelligence Technologies: Concepts, Priorities, and Questions.” Journal of Personal Selling and Sales Management 39 (1): 2–22. https://doi.org/10.1080/08853134.2018.1557525.

Syam, Niladri, and Arun Sharma. 2018. "Waiting for a Sales Renaissance in the Fourth Industrial Revolution: Machine Learning and Artificial Intelligence in Sales Research and Practice." Industrial Marketing Management 69 (December 2017): 135–46. https://doi.org/10.1016/j.indmarman.2017.12.019.

Van Der Kooij, Ing Jacco J. 2019. "Frameworks Governing B2B Marketing , Sales and Customer Success and Why SaaS Needs Its Own Framework." https://winningbydesign.com/frameworks-that-govern-b2b-marketing-and-sales/.

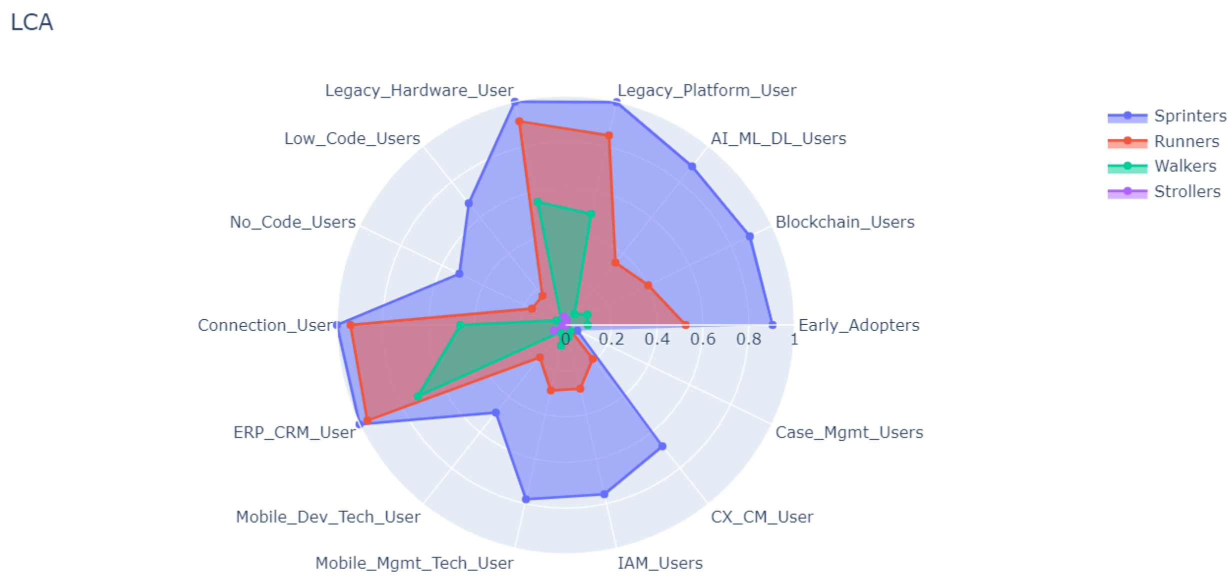
Winkler, William E. 1990. "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage." Proceedings of the Section on Survey Research, American Statistical Association, 354–59. http://files.eric.ed.gov/fulltext/ED325505.pdf.

Zadrozny, Bianca, and Charles Elkan. 2002. "Transforming Classifier Scores into Accurate Multiclass Probability Estimates." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, no. November 2012: 694–99. https://doi.org/10.1145/775047.775151.

Zheng, Alice. 2015. Evaluating Machine Learning Models: A Beginner's Guide to Key Concepts and Pitfalls. O'Reilly.

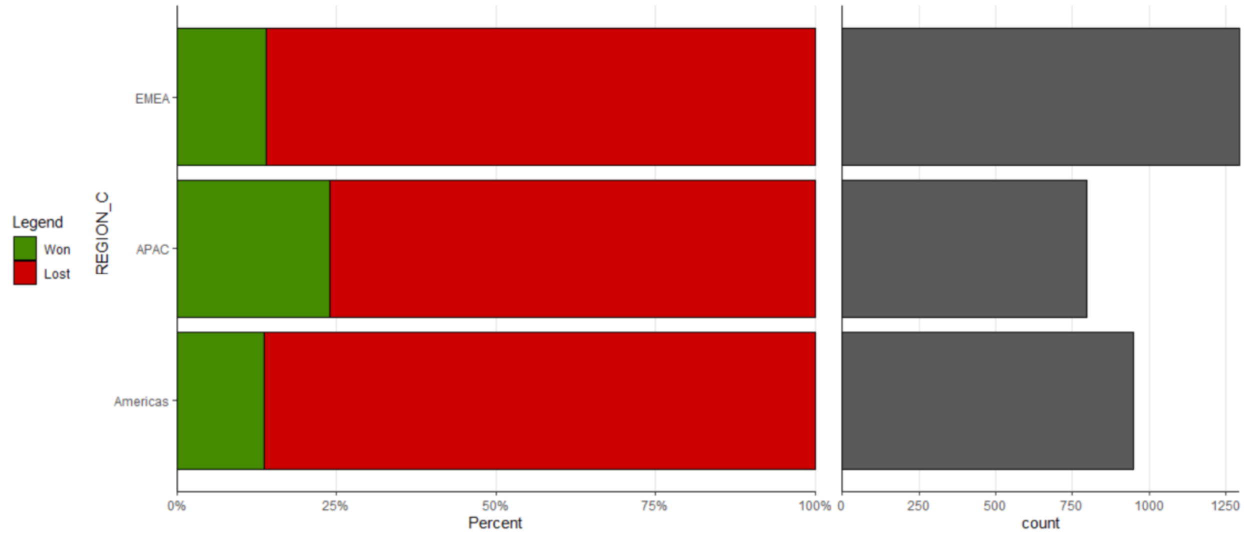
Appendix

A1. Distribution of the Tech Segments, after clustering. The radius represents the percentage of accounts in each segment that use at least one technology of that category. It is possible to see a clear distinction between segments, being the *Sprinters* the ones that adopt the highest number of technologies and *Strollers* that adopt the lowest.

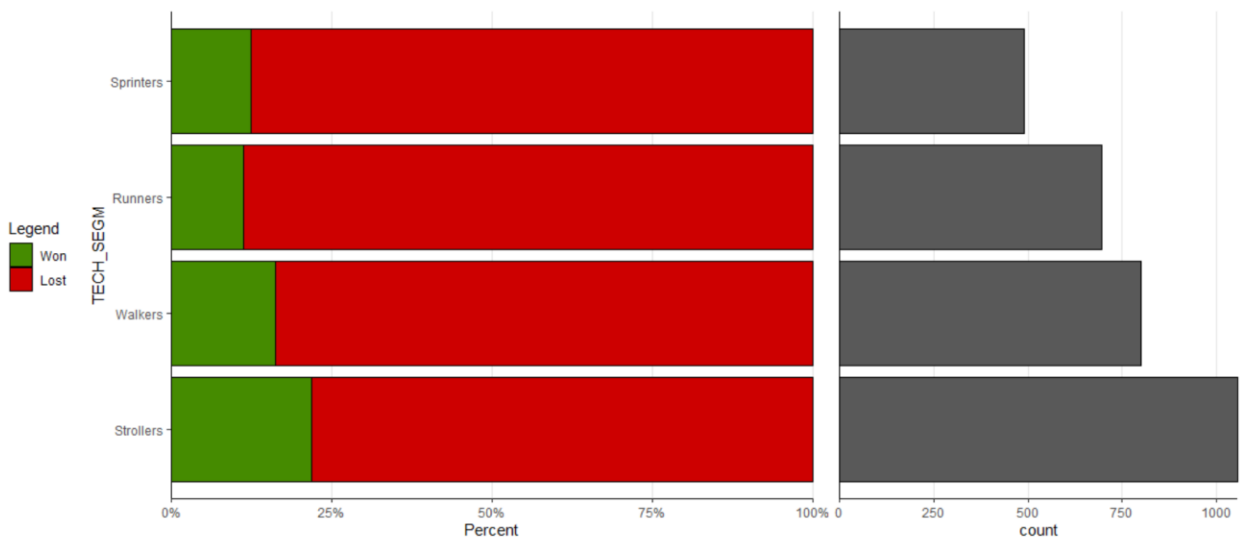


A2. Exploratory Data Analysis (EDA)

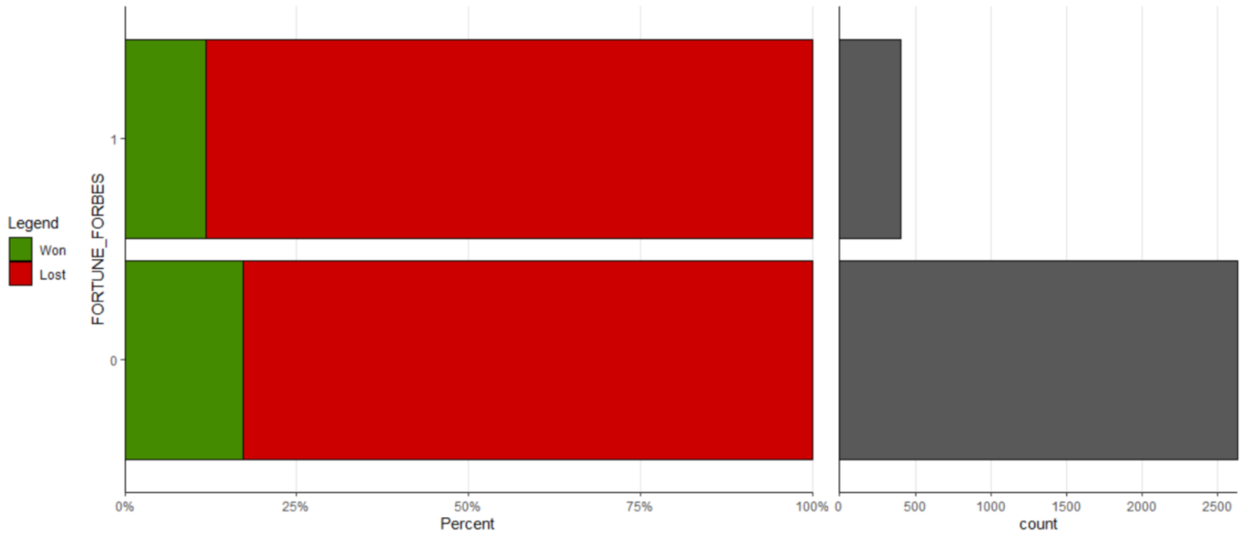
A2.1. Win rate by Region



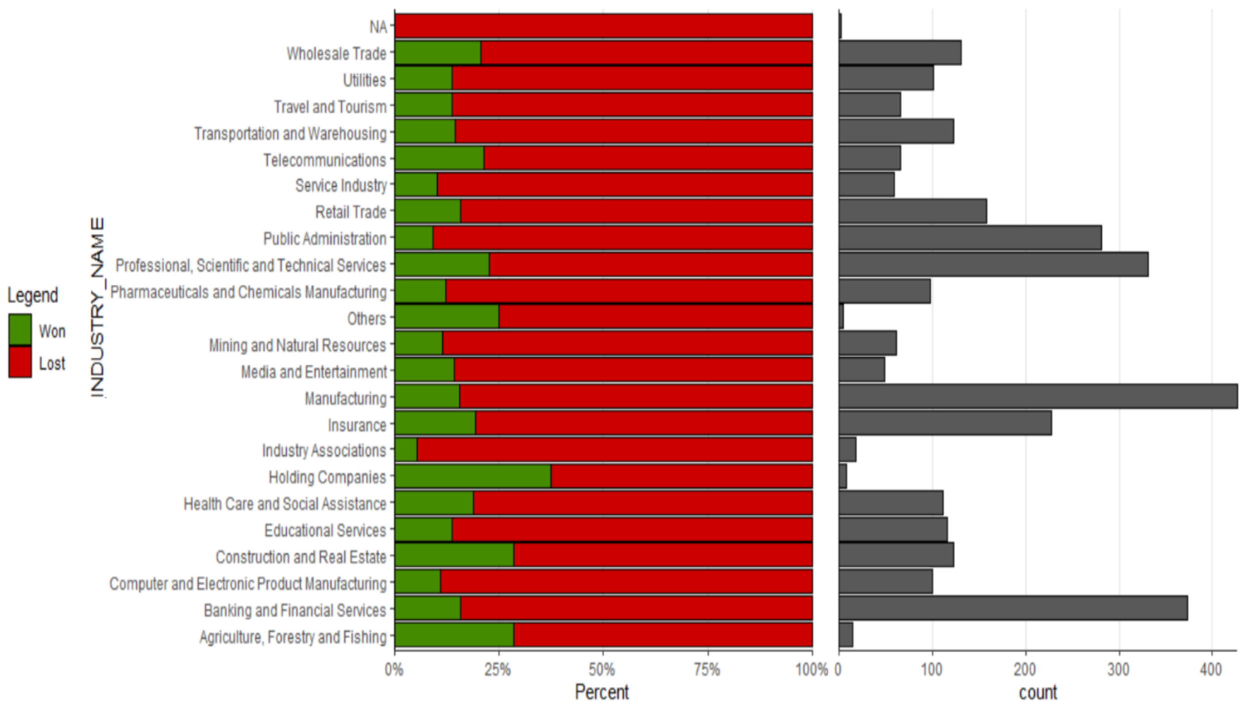
A2.2. Win rate by Tech Segment



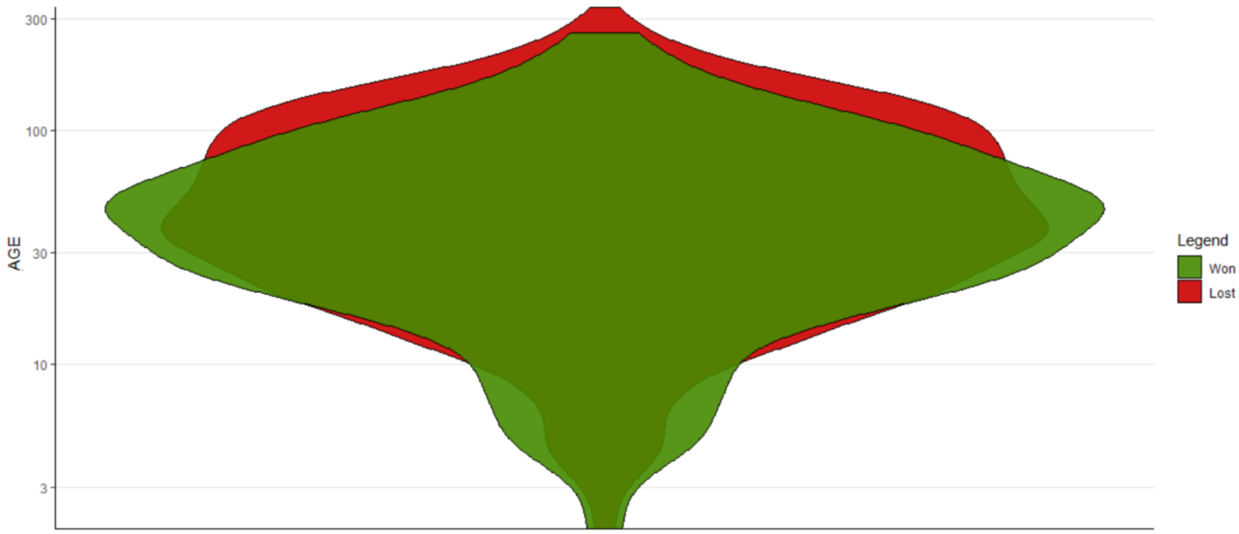
A2.3. Win rate by Fortune/Forbes rankings presence



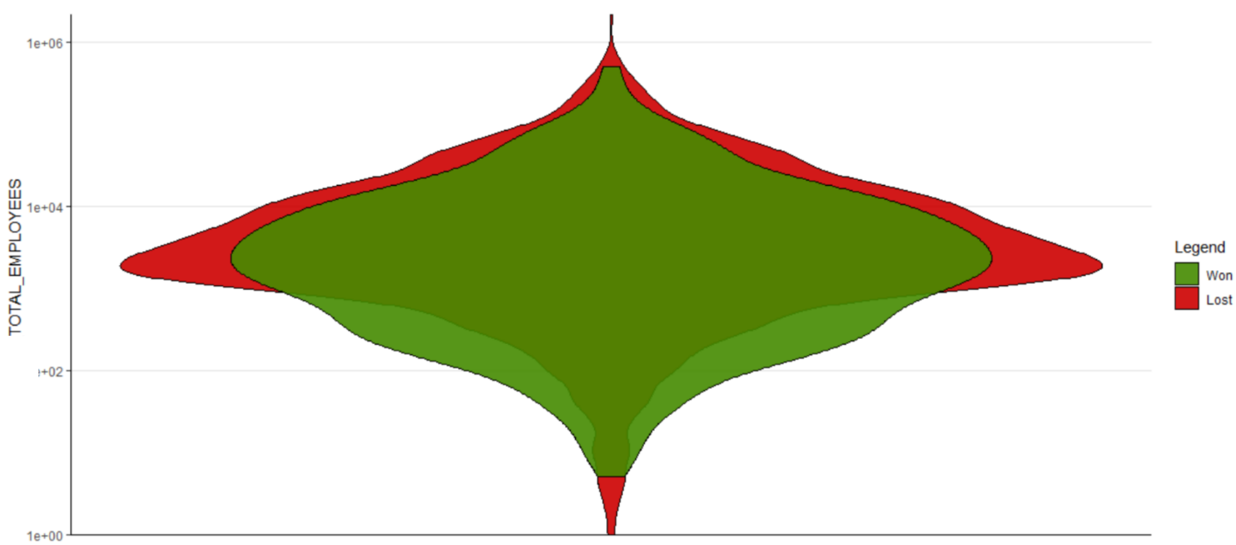
A2.4. Win rate by Industry



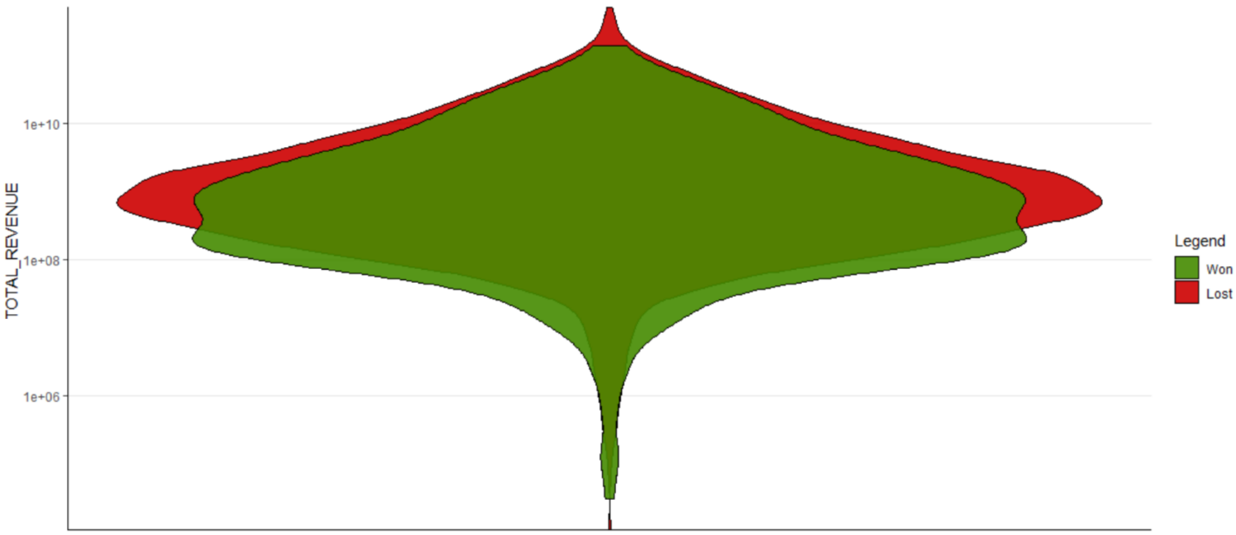
A2.5. Distribution of Company Age by Win rate. Logarithmic scale.



A2.6. Distribution of Number of Employees by Win rate. Logarithmic scale.



A2.7. Distribution of Total Revenue by Win rate. Logarithmic scale.



A3. Encoding

A3.1. LightGBM: Since LightGBM handles both categorical data and missing values, the only encoding done was to set an order to the Tech Segment feature, as it is ordinal (from 0 to 3, lowest to highest pace of adoption)

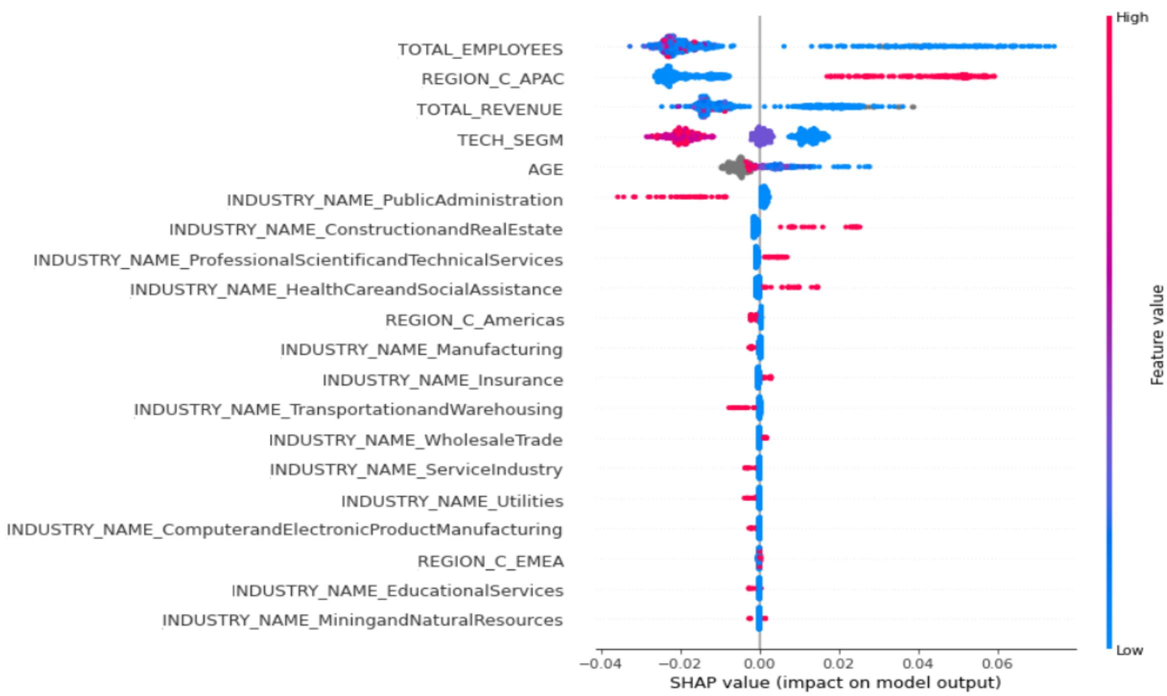
A3.2. CatBoost: CatBoost shares the same encoding as LightGBM, the only difference was that the missing values were imputed with -1, something that is asked in the documentation of the package.

A3.3. XGBoost**:** Since XGBoost is not able to deal with categorical variables by itself, the Region and Industry Names were one hot encoded, creating a different column for each different category. The ordinal encoding of the Tech Segment was again applied.

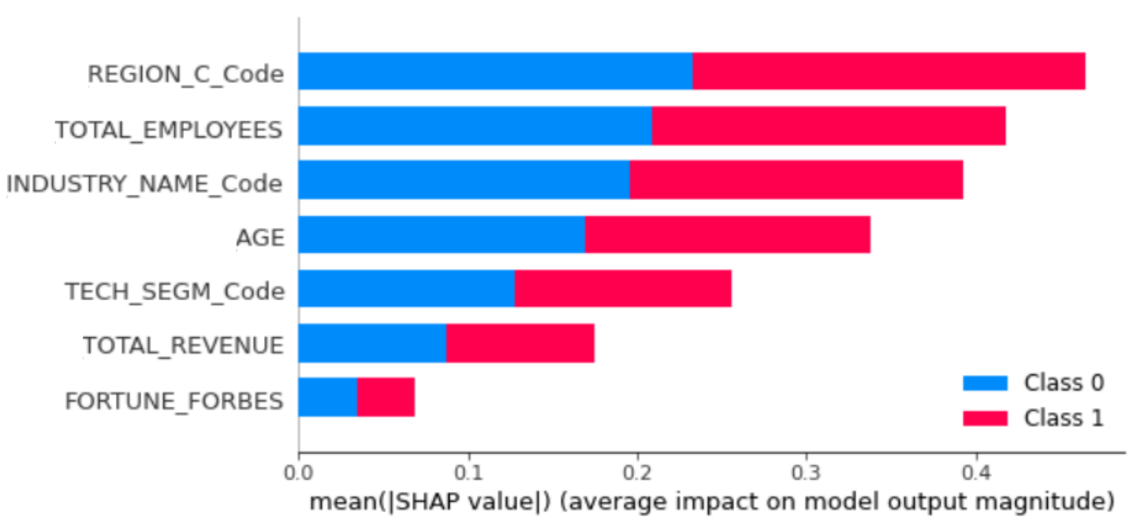
A3.4. Logistic Regression**:** Since the Logistic Regression is not able to handle neither categorical variables nor missing values, the encoding applied was the same as the XGBoost, with the extra step of imputing missing values with -1.

A4. Explainability

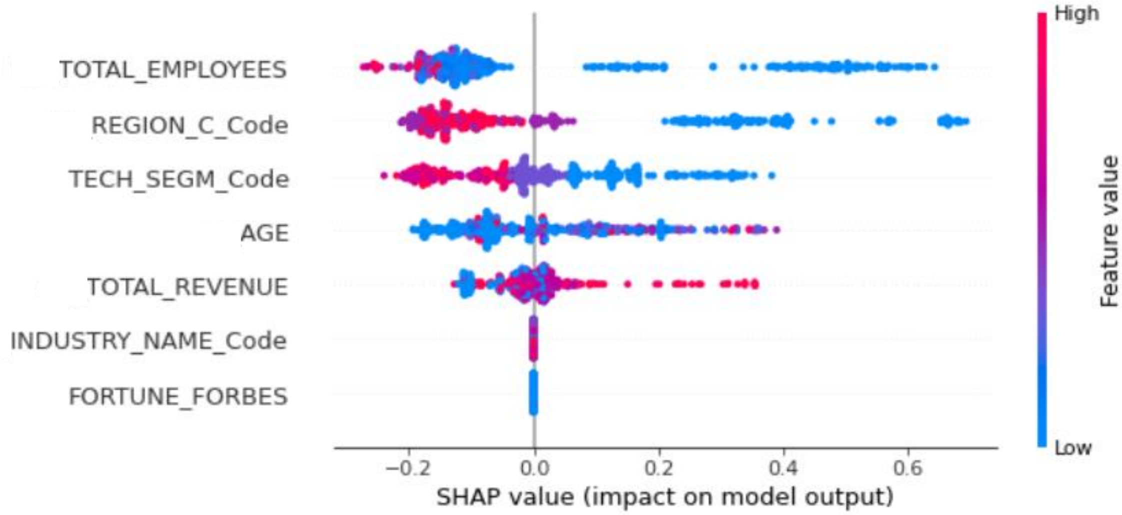
A4.1. *Summary plot* for XGBoost. Since the categorical variables were one hot encoded for the XGBoost, it was the chosen plot to be presented on the body of the project. It gives a better outlook on how each category impacts the final output



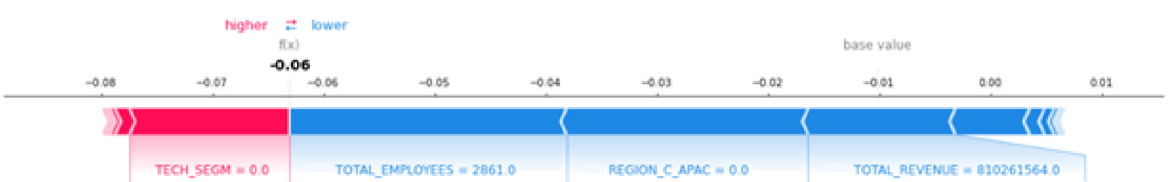
A4.2. *Summary plot* for LightGBM. Only the magnitude of the impact is presented, giving just enough information to assess the importance of the features.



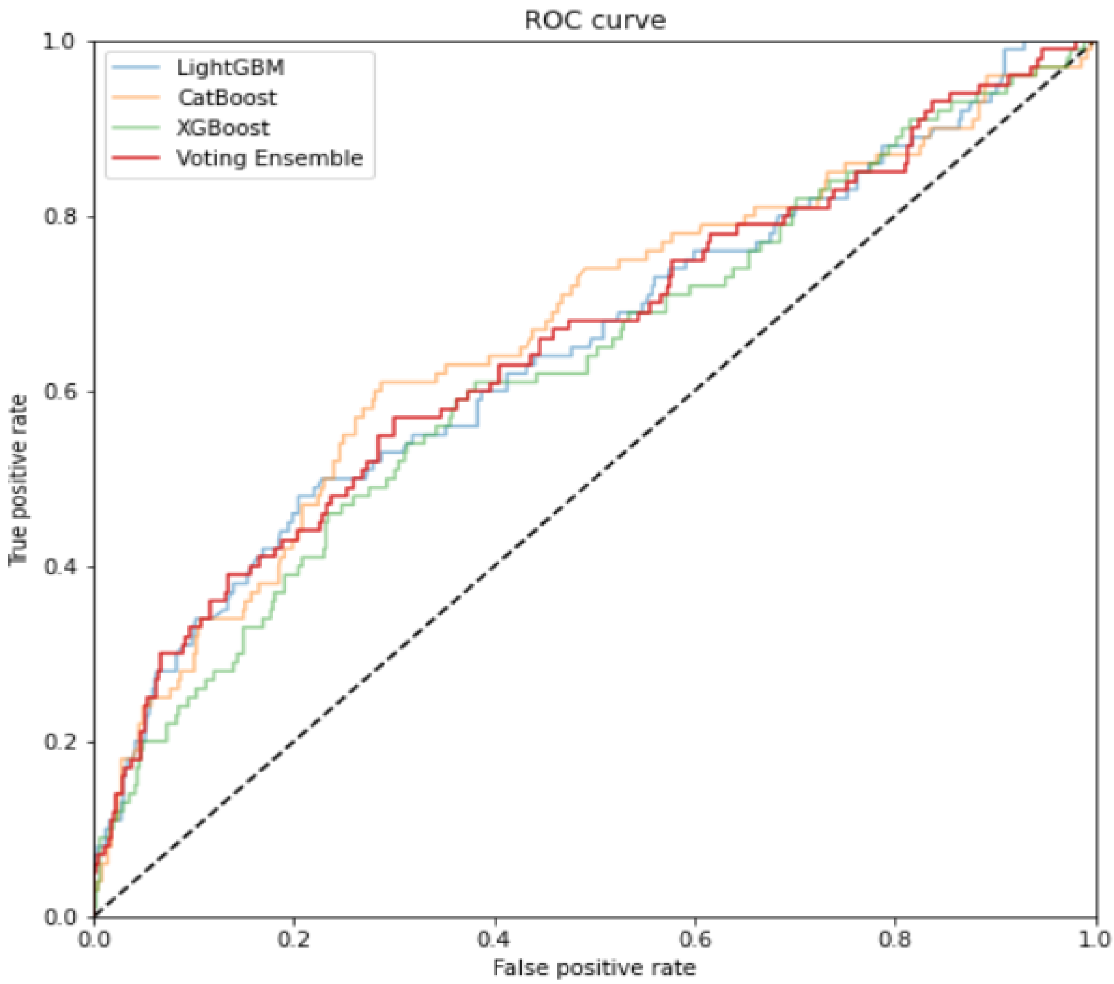
A4.3. *Summary plot* for CatBoost. Even though CatBoost was the best performing model, the representation of the explainability is harder to interpret, as there is no proper distinction between categories.



A4.4. Example of a *Force Plot* for XGBoost, for a random observation in the test set. In red it is possible to see the features that push the model prediction higher, in blue the opposite. The numbers presented on the axis represent the model output before the application of the sigmoid function and the probability calibration.

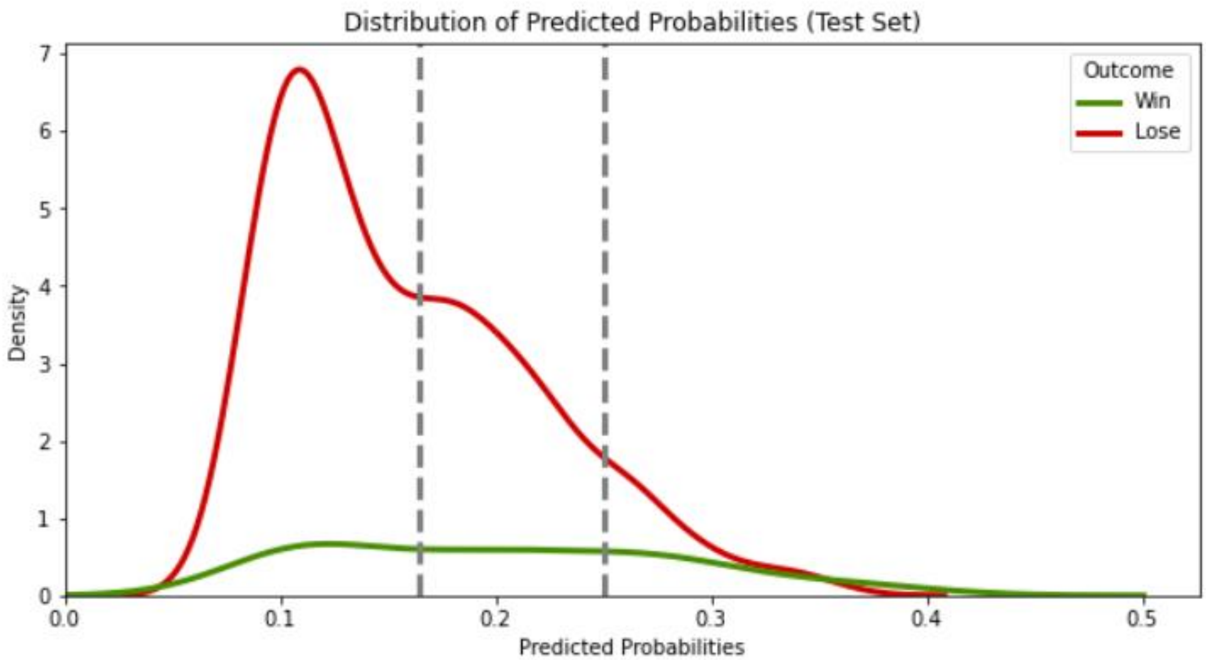


A5. ROC curves



A6. Distribution of predicted probabilities and Thresholds

A6.1. Distribution of predict probabilities and thresholds, defined on the train set and presented on the test set (represented in grey). The thresholds were defined considering the overall distribution of predicted probabilities.



A6.2. Distribution of predict probabilities and thresholds, defined on the train set and presented on the test set (represented in grey). The thresholds were conditionally defined, by Region and Tech Segment, yielding a total of 24 different thresholds.