Categories's Churn: A Machine Learning Approach in Retail

Carlos Sousa[1,3], A. Manuela Gonçalves[1,2(✉)], and Ana da Costa Freitas[3]

[1] Department of Mathematics, University of Minho, Guimarães, Portugal
mneves@math.uminho.pt
[2] Centre of Mathematics, University of Minho, Guimarães, Portugal
[3] Retail Chain, Lisbon, Portugal

Abstract. In the realm of business, the cessation of relations with a company by its customers can yield profound financial repercussions. Such consequences manifest as diminished revenue and profitability, while also bringing risk to the company's reputation. Thus, comprehending the roots of customer churn is of major importance. Equally crucial is the formulation of effective strategies to mitigate churn, thereby enhancing customer satisfaction, retention, and overall profitability. Within the framework of this work, clustering techniques were deployed alongside machine learning methodologies. This combination helped achieve a balance between accuracy and simplified model communication. In addition, data analysis techniques were deployed within the context of churn analysis, where total churn emerged as the optimal solution for the project, even if a case for partial churn could be made. The outcome was the development of a cohort of models capable of identifying the majority of churners within the selected pilot categories. Which empowers the retention department to implement more impactful retention campaigns, ensuring a more effective response to customer churn and bolstering overall business stability.

Keywords: Churn · Machine Learning · Clustering · Classification · Retail · Big Data

1 Introduction

There are a variety of factors that contribute to churn, including poor customer service, a lack luster customer experience, and a failure to provide value to customers. In addition, changing economic conditions, shifting consumer behavior, and evolving industry trends can all contribute to churn. Portuguese consumers have been gradually shifting towards online shopping, driven by convenience and changing consumer preferences. This forced companies to adapt in order to avoid losing customers.

In the Portuguese retail market, churn is a particularly important subject. According to a research made by The Consumer Goods Forum [1] 73% of Portugal retail customers bought in more than one supermarket banner per week,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024 Workshops, LNCS 14815, pp. 319–336, 2024.
https://doi.org/10.1007/978-3-031-65154-0_20

with 21% shopping in more than two. The willingness of Portuguese consumers to change providers underlines the competitive nature of the retail landscape. This is particularly evident in the grocery sector, where consumers are often sensitive to price fluctuations and promotions. While price is a critical factor in consumer choice, it is not the sole determinant of customer loyalty [1]. Retailers need to consider a holistic approach to customer retention, addressing various aspects of the shopping journey.

The aim of this work is to create a set of predictive models capable of accurately predict and explain churn to a category level in a major Portuguese retailer. To do so, state of the art machine learning and data analysis techniques will be applied to the extensive existing customer and transactional databases and the derived results analysed, presented and implemented into production. Category churn will help in the creation of better focused promotional campaigns that current models do not allow for, while also allowing to individually approach categories that are more permeable to competitor influence. These models are also to be integrated into a bigger project aiming to develop an Advanced Diagnostic and Optimization Tool (ADOT), in order to assist higher management with decision making.

1.1 Data Imbalance

When the classes in a dataset are not represented equally, standard machine learning algorithms can have poor performance on under-represented classes. This is because, in such scenarios, the model is trained on mostly one class, leading to a bias towards the majority class and poor generalization to the minority class [2,3].

Resampling techniques, such as oversampling the minority class, undersampling the majority class, or a combination of both, can be used to balance the dataset. This can be done by either duplicating instances of the minority class or removing instances of the majority class. Random Oversampling (ROS) and Random Undersampling (RUS) are two well known sampling techniques, however, because of the random nature of the selection involved in duplicating and discarding the data values, these approaches lack consistency and show varying performance [2]. On top of that oversampling can lead to overfitting and a decrease in the diversity of the dataset, while undersampling can cause a loss of important information even in informed sampling techniques, like EasyEnsemble and BalanceCascade algorithms [3], therefore, a combination of oversampling and undersampling is often used. One such approach is to use SMOTE (Synthetic Minority Over-sampling Technique), capable of generating synthetic samples based on the feature space similarities between existing minority examples, in conjunction with a oversampling technique.

Cost-sensitive learning is another technique that can be used to address data imbalance. This approach incorporates the class imbalance into the learning algorithm by assigning different misclassification costs to different classes. Instead of creating balanced data distributions through different sampling strategies, cost sensitive learning targets the imbalanced learning problem by using different cost

matrices that describe the costs for misclassifying any particular data example. This helps to ensure that the model pays more attention to the minority class during training [3]. Even though these techniques can be effective in addressing data imbalance, they are not a one-size-fits-all solution. Therefore, it is essential to thoroughly evaluate different approaches, before implementing them in a particular application.

1.2 Classification Tools

Classification is a fundamental problem in machine learning, where the goal is to predict the class of a given input based on its features. Churn prediction falls under this type of problem, usually trying to distinguish customers into two classes churner or not churner. In order to predict churn, several methods have been proposed and used during the years, being the most widely established and used ones Logistic Regression [5], Random Forest [6], Support Vector Machines (SVM) [7], Artificial Neural Networks [8], and Naive Bayes [4,9].

Boosting algorithms, that work by combining several base classifiers, sequentially estimated from perturbed versions of the calibration sample, have also been used to improve the classification performance of previously mentioned methods [10]. The biggest trade off between classification methods is that, usually, when performance increases the models tend to become less interpretable. This is mostly fine, as long as predictions are the sole purpose of the model. In more complex business environments, where identifying the roots behind the churn is important the performance/interpretability trade off must be considered more carefully. A business might be willing to trade some of the predictive performance of the model for the ability to better understand how each of the features interacts with the possibility of churn. Fortunately, over the years action has been taken in this regard and developers have built tools, like SHAP (Shapley Additive Explanations) or InterpretML, that look to bring to light more information out of these systems. The InterpretML package also included the first implementation of an Explainable Boosting Machine (EBM), a glassbox model that can be as accurate as many black-box models while remaining completely interpretable [11]. It works by building upon the principles behind Generalized Addictive Models with the use of techniques like boosting and bagging in conjunction with pair wise interactions that help to further boost accuracy.

2 Methodologies

2.1 Establishing Customer Representation and Churn

Customer definitions can vary depending on the context and objectives of the study. In industries where subscriptions are the norm, a customer may be defined as an individual or business entity that has an active subscription or contract with a service provider. On the other hand, in the retail sector, a customer may be defined as an individual who has made a purchase within a specific time

frame or has reached a defined spending amount. While the specific criteria for identifying churn can differ based on the nature of the industry and the available data, by establishing clear definitions for customers and churn, it's possible to ensure consistency in the analyses and results, while minimising the risk of bias in the model. This section used data for the years of 2021 and 2022.

Defining Customer. In a company with the same dimensions as MCS, we may encounter a considerable number of customers who, despite being included in the database, exhibit shopping behaviors that are either sporadic or occur in such low volume that they can have a negative impact on the overall project results. To address this issue, a thorough analysis of the underlying data was conducted, and actions were taken to handle such customers. The first step involved identifying and excluding "*oneshot*" customers, those with only one purchase in a given time frame, in this case one year. Subsequently, the shopping behavior of the remaining customers was analyzed by plotting the number of customers under specific spending thresholds. A curve was then drawn, and its slope was measured. The objective was to identify points just before significant increases in the slope, which indicate potential changes in customer behavior. In this case there's two issues to be taken into consideration, the number of churners that would be removed and the cost of activation. There's also a balance to be struck in determining how much data should be sacrificed to reduce noise. The cut should ideally be made before reaching a point of acceleration to prevent the removal of essential churners from our dataset. Additionally, it's crucial to ensure that unprofitable customers are not inadvertently activated.

Two specific spending thresholds, 30€ and 50€, were identified as being particularly important, Fig. 1. The number of customers under each threshold was taken into consideration to determine how much data could be sacrificed to reduce noise in the dataset. Ultimately, the chosen threshold was 30€, which means only customers with at least two purchases and a minimum total spending of 30€ in the last 12 months were considered in this study. This selection ensured that the analysis focused on customers with more consistent shopping behavior and a higher level of engagement with the company, without risking losing data for customers that, in extreme cases, only buy a very small or even just one category.

Partial vs Total: Defining Customer Churn. To define churn we had to first isolate churners in order to observe their behaviour. To do this, a specific time frame was chosen, and T set as the target month. During this step, any customer who made a purchase on month T but did not make any further purchases between T+1 and T+6 (a six-month period) was considered a churner. This criterion ensures that T is the last month in which the customer made a purchase and confirms that they were not previously classified as a churner before that time, Fig. 2.

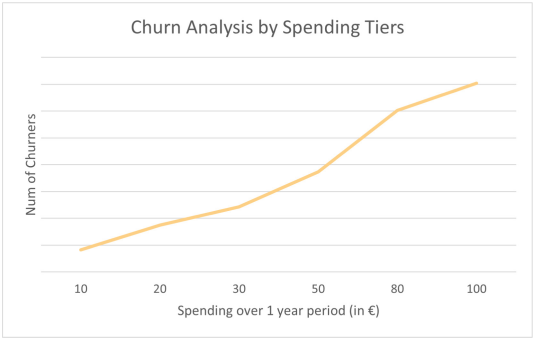


Fig. 1. Number of customers in a specific twelve month spending threshold that did not make a purchase in the following six months

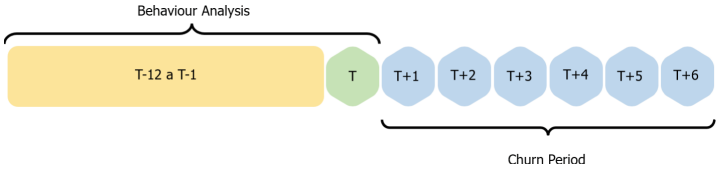


Fig. 2. Diagram of the process used to identify churners for behaviour analysis

Next, the transactional behavior of customers leading up to T (including month T) was analyzed to determine if there were any indications of partial churn prior to the actual churn event. This involved comparing each customer's monthly spending with their average monthly spending. A ratio was then crafted to observe the sales evolution over the year and the results aggregated and compared with non-churners to mitigate any potential bias caused by seasonal variations. This ratio was made by comparing a customer monthly spending to the average spending for the year, as follows

$$\text{Racio} = \frac{\text{Spending month X}}{\text{Average Spending for the Year}} \tag{1}$$

Initially, the findings of this analysis were underwhelming. There was a distinction in behavior between churners and non-churners, as seen in Fig. 3, but this wasn't as big as we had expected from analysing similar works. Due to these unexpected results, a different approach was adopted. The identified churners were categorized into groups based on their value segment, utilizing the industry-standard Recency, Frequency, Monetary (RFM) value system, and the transactional behavior was then reanalyzed using this approach. This are already existing segments of clients, previously created by MCS that group clients according to their overall value. Ranging from groups one to six, with one being the most valuable customers with the value progressively decreasing as the group increases.

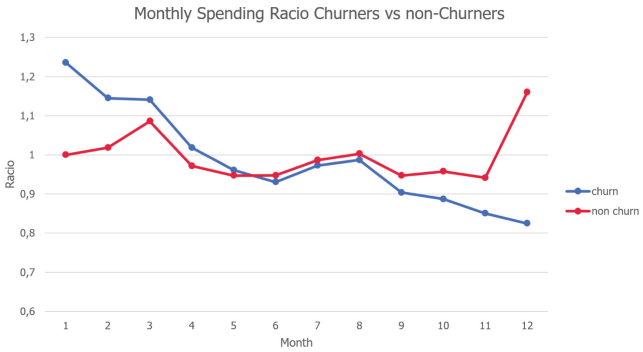


Fig. 3. Comparison of customer monthly spending vs it's average for churners and non-churners, leading up to the target month

With the new approach, a pattern started to emerge. Customers with higher value segments exhibited a more consistent pattern of partial churn, whereas those with lower value segments demonstrated more erratic behavior. This pattern of partial churn was particularly pronounced in the groups 2 and 3 of the RFM segments, loyal clients with medium and small spending, which showed a more pronounced decreased, Fig. 4. Thus, evidence supporting partial churn was emerging. However, upon further analysis, it became evident that the majority of churn cases were in the lower value groups of the customer base, where the partial churn behavior was less pronounced or non-existent. At the time, separating these groups of customers wasn't a feasible option for the company, because the increased complexity would complicate information sharing between different departments, the decision was made to adopt the definition of total churn.

2.2 Clustering Techniques

After reviewing literature for different techniques, we ultimately settled on K-Means and hierarchical clustering. Hierarchical clustering, utilizing the ward method, was employed to visualize the potential clusters. Multiple cuts were then applied to the cluster tree to explore different ways of dividing the clusters that could be created using K-Means. The resulting clusters from different cuts were used to modify the value of 'k' in K-Means, their quality was assessed using the silhouette method, and the outcomes were reviewed by business experts for confirmation and accuracy.

2.3 Performance Metrics and Validation Techniques

The assessment of machine learning model performance in scientific research hinges on the crucial elements of performance metrics and validation techniques. These indispensable tools serve to measure a model's effectiveness on a given dataset and to gauge its ability to generalize to new, unseen data. The selection of



Fig. 4. Comparison of customer monthly spending vs it's yearly average, divided by value segment groups. Top left groups 2 and 3, top right groups 1, 4 and 5 and bottom groups 5 and 6.

appropriate performance metrics and validation techniques is a pivotal decision that warrants careful consideration, tailored to the specific nature of the machine learning task and the characteristics of the available data. A diverse array of performance metrics is at our disposal to assess machine learning models, with the choice contingent on the type of machine learning task under consideration. For this work a mix of metrics was employed, in specific accuracy, precision, recall, the F1 Score, as well as a careful analysis of the confusion matrix and the Area Under the Receiver Operating Characteristic Curve or Area Under the Curve (AUC).

2.4 Models

Explainable Boosting Machine: An Explainable Boosting Machine (EBM) is a machine learning algorithm that combines the principles of boosting and interpretable decision trees to create a predictive model that is easy to understand while maintaining competitive predictive performance. EBMs are based on Generalized Addictive Models (GAMs), but while GAMs were traditionally fit using smooth low order splines, EBMs learn shape functions using boosted decision trees. EBMs are especially interesting because they often match the accuracy of complex blackbox algorithms like XGBoost and random forests, while having a simple optimization procedure and final structure [12].

This restricted machine learning models that have the form:

$$g(E[y]) = \beta_0 + f_1(x_1) + f_2(x_2) + ...f_k(x_k) = \beta_0 + \sum_{j=1}^{k} f_j(x_j)\,, \tag{2}$$

where β_0 is an intercept, each f_j is a univariate function that operates on a single input feature x_j, and g is a link function that adapts the model to different settings like classification and regression.

So, first EBMs learn each feature function f_j using machine learning techniques like gradient boosting or bagging, with the process being restricted in order to train on one feature at a time, in a round robin fashion, using a very low learning rate so that the order of the features does no matter and in order to mitigate co-linearity between features.

This following example shows how the model works each of the features: $y = a_0 + a_1(T_1^1(x_1) + ... + T_r^1(x_1)) + a_2(T_2^2(x_2) + ... + T_r^2(x_2)) + ... + a_k(T_1^k(x_k) + ... + T_r^k(x_k))$. Second this models can also include pairwise interaction terms of the form $g(E[y]) = \beta_0 + \sum f_i(x_i) + \sum f_{i,j}\,(x_i, x_j)$, which helps to further improve accuracy while maintaining intelligibility. In the context of crafting predictions on an individual basis, each function within the framework of Explainable Boosting Machines (EBMs) operates akin to a specialized lookup table corresponding to distinct features. The outcome of each function is a term contribution, representing the impact of the associated feature. By summing these term contributions and subsequently subjecting the aggregate to a designated link function, the ultimate prediction is computed. The key advantage of EBM lies in its modular structure, characterized by additivity. This structural property permits the sorting and visualization of term contributions, thereby facilitating the identification of features that wield significant influence over a particular prediction [12].

Decision Trees: Decision trees have emerged as a foundational and versatile tool for classification and regression tasks. Their roots trace back to the early days of artificial intelligence, making them one of the enduring pillars of data-driven decision-making.

In this work an optimized version of the CART algorithm was used, as implemented in the scikit learn python library and can be described as follows. Consider a specific node, denoted as m, which encapsulates a dataset denominated as Q_m, housing a count of n_m samples. The core decision centers on dividing this data into two distinct subsets: $Q_m^{\text{left}}(\theta)$ and $Q_m^{\text{right}}(\theta)$. Here, θ is instrumental in specifying the conditions for this partition: $x_j \leq t_m$.

The efficacy of a candidate split at node m hinges on evaluating the impurity function denoted as H. This function serves as a gauge of data homogeneity, ensuring the alignment of samples within the same class. The measure of the quality of the candidate split at node m is quantified as

$$G(Q_m, \theta) = \frac{n_m^{\text{left}}}{n_m} H(Q_m^{\text{left}}(\theta)) + \frac{n_m^{\text{right}}}{n_m} H(Q_m^{\text{right}}(\theta))\,. \tag{3}$$

The objective is to identify the optimal candidate split, θ^*, which minimizes the impurity measure.

This process of selecting the best candidate split is reiterated, employing θ^*, and the data is further partitioned into left and right subsets ($Q_m^{\text{left}}(\theta^*)$ and $Q_m^{\text{right}}(\theta^*)$), until certain termination criteria are met. These criteria encompass reaching the maximum allowable depth or encountering nodes with an insufficient number of samples (i.e., $n_m < \min_{\text{samples}}$ or $n_m = 1$).

In scenarios where classification is the goal, involving target values ranging from 0 to $K-1$, the proportion of class observations in node m is quantified as $p_{mk} = \frac{1}{n_m} \sum_{(x,y)\in Q_m} I(y = k)$. Various measures of impurity are considered, with two common options being the Gini Impurity and Log Loss (or Entropy). The Gini Impurity is computed as $H(Q_m) = \sum_k p_{mk}(1 - p_{mk})$. The Log Loss (or Entropy) is represented as $H(Q_m) = -\sum_k p_{mk} \log(p_{mk})$.

Logistic Regression: Logistic regression is a statistical method widely used for modeling the relationship between a binary outcome variable and one or more predictor variables. Unlike linear regression, which aims to predict continuous numeric outcomes, logistic regression is particularly suited for scenarios where the dependent variable is categorical and takes on two possible outcomes, often referred to as "success" or "failure", "1" or "0".

In it's usual form, the simple logistic regression models is given by:

$$E(Y|x) = \pi(x) = \frac{exp(\beta_0+\beta_1 x)}{1+exp(\beta_0+\beta_1 x)} = \frac{1}{1+exp(-(\beta_0+\beta_1 x))}\,, \tag{4}$$

where β_0 is known as the intercept (the vertical intercept of the line $y = \beta_0+\beta_1 x$) and β_1 is the inverse scale parameter, $\beta_1 x = \frac{1}{s}$. These are the y-intercept and slope of the log-odds as a function of x.

When multiple explanatory variables are present, the model takes the form:

$$P(Y = 1|X) = \frac{1}{1 + exp^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p)}}\,. \tag{5}$$

The output of this equation is the estimated probability of the positive class. For modeling purposes, the β parameters that minimize the cost function must be found. The typical cost function is the log-loss function, which for two classes takes is defines as $J(\beta) = -\frac{1}{m}\sum[y_i log(\hat{p_i}) + (1 - y_i) log(1 - \hat{p_i})]$, where y_i is the actual class label and $\hat{p_i}$ is the predicted probability of the positive class for the i^{th} instance.

3 Case Study

3.1 Data Collection and Methods

In this section, the data sources and methodologies employed are described, providing a comprehensive account of the research approach undertaken in the project.

Data. The collection and processing of data from *Cartão Co* database was done using a combination of Structured Query Language (SQL) and Pyspark, a Python API for Apache Spark that enables real-time, large-scale data processing in a distributed environment.

Several datasets were used for different parts of the project. Used transactional and product data from *Co* stores, *Co*, *Co Mo*, *Co Online* and *M Su*, transactional data from partner stores, like *Ga* and *Ib*, and DNA data. This was composed of hourly data, weekly, and monthly data (depending on specific dataset) retrieved for the period between January 2021 and February 2023, comprised of hundreds of millions of transactions and 4.2 million customers. Different parts of the project used different subsets of this data.

As for model training, do to computational effort required to analyze all the data, a representative sample of about 800.000 customers was collected for this purpose.

3.2 Category Clusters and Churn Periods

Data must be communicated across the company different departments, like sales or costumer retention, and so making complex problems simpler to understand helps with communication. With this in mind clustering was proposed as a way to reduce the number of necessary churn periods. By grouping categories we reduce the number of different churn periods mitigating some of the complexity behind the problem.

This section utilized data for the entire year of 2021, and initially the process involved identifying potentially irrelevant categories. Subsequently, we proceeded with the clustering process, which served two purposes: grouping categories with similar behaviors and identifying categories with distinct behaviors and potential outliers.

For this purpose a group of six variables, considered to be capable of representing category behaviour from the available data, were selected. This being Number of Transactions, Number of Customers, Sales, Mean Value Added Tax (VAT), Average Basket and Mean Frequency. All of the variables were grouped by category. Then, before proceeding to the clustering, a heat map was created to observe correlation between variables. This allows to identify variables that could be introducing bias into the models by bringing the same information, or very similar information, into the models and artificially increasing its weight.

After the analysis, highly correlated variables were excluded and a three variable set up was selected, these being Number of Transactions, Average Basket and mean VAT. The number of transactions helps to identify how frequently a customer visits the store, along with providing information on purchasing behaviour. More transactions suggest a higher frequency of visits. Average Basket size helps identify shopping behavior, while VAT serves as a measure of an item's essentially based on the different tax tiers. This setup also offers the advantage of visualization in a three-dimensional space, making it easily understandable by plotting cluster results. For this specific purpose plot was created with the package plotly, that allows for the plot to be freely moved in a digital 3

dimensional space and allows for a easier perception of cluster distribution, with each axis representing one of the three variables.

The clustering techniques were applied to a subset of 59 categories. To determine the optimal value for k in the K-Means algorithm, an elbow plot was created, and the most suitable value was selected. The resulting clusters were evaluated using the silhouette method and further analyzed by an expert. However, upon assessment, it was concluded that the clustering outcome was not satisfactory. As a result, hierarchical clustering (Fig. 5) was employed to establish a new k-value, and the process was repeated. Eventually, a 6-cluster configuration was settled upon, k = 7 was used, but one cluster contained only one category and so was merged into the closest cluster after validation with the respective business department, with the following clusters: Essentials, Side Characters, Pantry Completers, Budget Killers, Delights & Clean, and Beauty & Beyond.

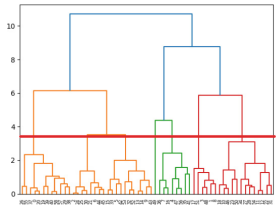


Fig. 5. Resulting dendrogram

Churn Periods. The selection of an optimal churn period is a delicate balancing act, where an excessively short period can lead to inefficient allocation of resources, while an overly extended one may render intervention efforts too late to effectively retain the customer. To ascertain the most appropriate churn periods, a comprehensive analysis of customer behavior spanning the years 2021 and 2022 was undertaken. This analysis involved an examination of the average time intervals between transactions within distinct product or service categories for each individual customer. From this intervals outliers were analyzed using Z-scores and the interquartile range.

3.3 Model Training

The study involved the evaluation of various models to ascertain their efficacy in classifying churners. Three distinct models - Logistic Regression, Decision Trees (DT), and EBM - were tested. To enhance the precision of the analysis, a subset of the training data was addressed, excluding customers with zero sales (0€) in the previous twelve months for the specific category under consideration. This step is crucial in eliminating the potential misclassification of customers who had never engaged in transactions within that category, thus preserving the integrity

of the churner classification process. Otherwise, customers who never bought the category before, would be flagged as churners, but this would in fact be a false positive.

Then the dataset was organized chronologically. Subsequently, it was partitioned into distinct training and validation sets, with the training dataset encompassing the initial 80% of observations and the remaining 20% reserved for validation purposes, around 110 million and 27 million transactions respectively. This partitioning prevents any overlap between the training and validation and avoids training on data occurring after the events in the validation set, thereby helping ensure model's robustness and providing better reliability and validity of the findings presented.

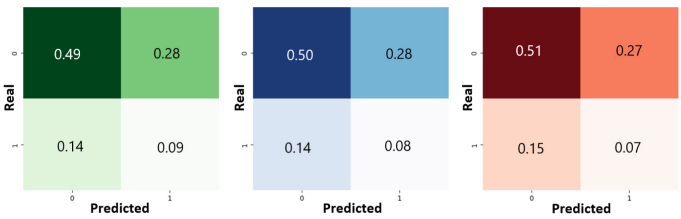


Fig. 6. (From left to right) Confusion matrix for Logistic Regression, Decision Tree and Explainable Boosting Machine

All of the models presented accuracies just above 50% given their particularly poor performance predicting the positive category, with more information available in Fig. 6.

Upon reanalyzing the data a problem was found in one of the features. The mistake was corrected and models were retrained. After some adjustments to hyperparameters the results improved greatly. Despite all of the models performing in line with each other, the EBM is particularly worthy of mention because of it's better ability to correctly predict churners and overall better accuracy. Bellow is shown the comparison of confusion matrix for fruits' models (Fig. 7).

The results remain similar for the other categories with the EBM being more robust and having overall superior performance, closely followed by logistic regression (Fig. 8). For this models, accuracy ranges from around 70% to 78% and AUC scores ranging from 82% to 86%, with very good precision and good recall for the negative class (0). As for the positive class, recall is good, but, except for beers, precision is weak remaining below the 60% mark. This happens because the models have a tendency to over-predict the positive class. This can be problematic, because customers classified as churners will be flagged so the retention department can take action and prevent churn, and that has a cost associated with it. In order to work around this limitation a small code snippet was created in order to allow for a change in the classification threshold. This way, by increasing the threshold we can increase the precision of the positive

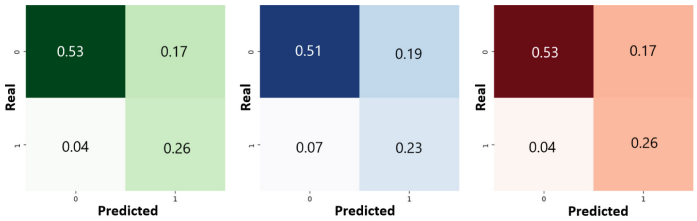


Fig. 7. (From left to right) Confusion matrix for Logistic Regression, Decision Tree and Explainable Boosting Machine, after corrections

class at the cost of recall, and increasing the return on investment. And so, after waiting the results for all the models and categories EBM was the chosen model for all the categories. Even if Logistic Regression could perform in line with EBM, when looking at the AUC metric, EBM is the clear winner.

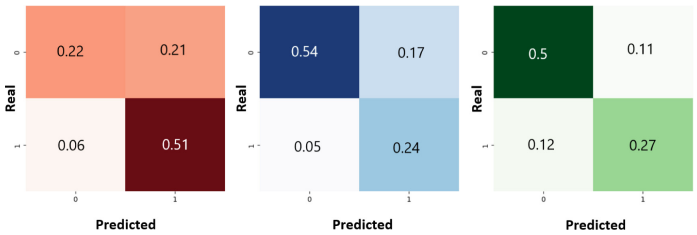


Fig. 8. (From left to right) Confusion matrix for Beers, Cheeses and Clothing Care and Cleaning

Looking into the evaluation metrics previously discussed, the models compare as displayed in Tables 1, 2 and 3, with the 0 and 1 next to recall and precision representing the relevant class.

3.4 Feature Importance

In order to extract feature importance from the model, the interpret package integrates two measures of importance, global and local.

Global importance gives the weight of the fifteen most important features for the model. The first plot is a general score and works by calculating average of the absolute predicted value of each feature for the training dataset. This is, for each feature, it calculates what each training data point would be scored by that feature alone. Then it takes the absolute value of all of these scores, and averages the result, because EBM is an additive model, this method corresponds with which features have the largest impact on predictions in the training set (Fig. 9).

Table 1. Model Comparison Categories

Category	Model	Precision 0	Recall 0	Precision 1	Recall 1	Accuracy	F1 Score
Fruits	EBM	0.93	0.75	0.49	0.80	0.76	0.61
	DT	0.88	0.73	0.42	0.66	0.71	0.51
	Logistic	0.92	0.75	0.48	0.79	0.76	0.60
Cheeses	EBM	0.91	0.77	0.59	0.81	0.78	0.69
	DT	0.70	0.59	0.30	0.41	0.54	0.34
	Logistic	0.70	0.56	0.30	0.44	0.52	0.36
Beers	EBM	0.79	0.52	0.71	0.90	0.73	0.79
	DT	0.58	0.51	0.66	0.73	0.63	0.69
	Logistic	0.75	0.56	0.72	0.86	0.73	0.78
C.C.C.	EBM	0.91	0.57	0.57	0.91	0.70	0.70
	DT	0.80	0.59	0.55	0.78	0.66	0.64
	Logistic	0.90	0.56	0.57	0.90	0.69	0.70

Table 2. EBM Comparison Across Categories

Category	Precision 0	Recall 0	Precision 1	Recall 1	Accuracy	F1 Score	AUC
Fruits	0.93	0.75	0.49	0.80	0.76	0.61	0.86
Cheeses	0.91	0.77	0.59	0.81	0.78	0.69	0.86
Beers	0.79	0.52	0.71	0.90	0.73	0.79	0.82
C.C.C	0.91	0.57	0.57	0.91	0.70	0.70	0.85

Table 3. Model AUC Comparison

Model	Fruits	Cheeses	Beers	C.C.C
EBM	0.86	0.86	0.82	0.85
DT	0.69	0.46	0.69	0.68
Logistic	0.77	0.50	0.71	0.73

By analyzing this features it's possible to derive hypothesis about the behaviour of different categories, specially when the focus is set on the differences between the plots. One very good case is the plot for the Beer category model, Fig. 10. For all of the tested categories the most important feature is gross sales for the specific category, and for all of them but Beers, the second most important feature is *h_ month_ trx_ racio*. For the category Beers the most important feature is *year_ month*. This is very interesting information, because the *year_ month* feature represents the month of the year signifying that the specific month for which we are running the model has a higher importance than for the other three features, hinting to a more seasonal behaviour in this case.

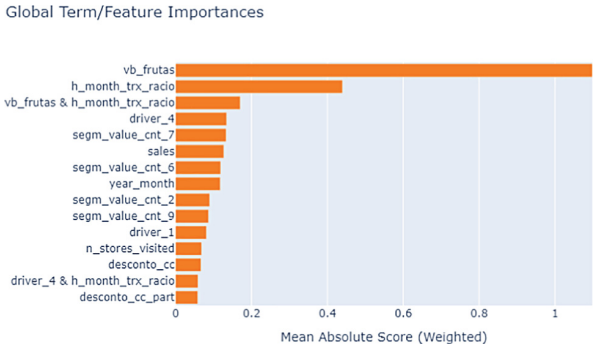


Fig. 9. Summary of the feature importance plot for the model for category Fruits

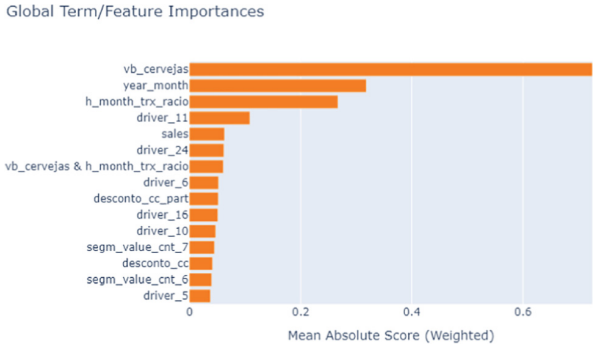


Fig. 10. Summary of the feature importance plot for the model for category Beers

The package also generates individual plots for each of the variables explaining their contribution to the final score. The X axis represents the value taken by the feature and Y-axis represents the log odds, these get then passed through a logit link function to get probability scores at prediction time (Fig. 11).

For the local importance, the package allows the user to see how each of the variables influenced a specific prediction. For each data point the fifteen more important features, for that case, are shown, and for each feature it's corresponding weight. So for each data point the weights would be added and the final value passed through a logit link function to get the probability score. As an example, for the example seen in Fig. 12.

One case where we could see the model reasoning clearly at work is on the wrong predictions. Bellow we see the example of a customer wrongly classified as churner by the model.

In this case (Fig. 13) we can see how a low value for *sales* (6,66€) in *vb_frutas* coupled with a high racio of spending in one month (40% of all expenditure in a month) and the fact that it belongs to the lowest value segment (*segm_value_cnt_9*) pushed the model to classify the customer as a churner with a probability of

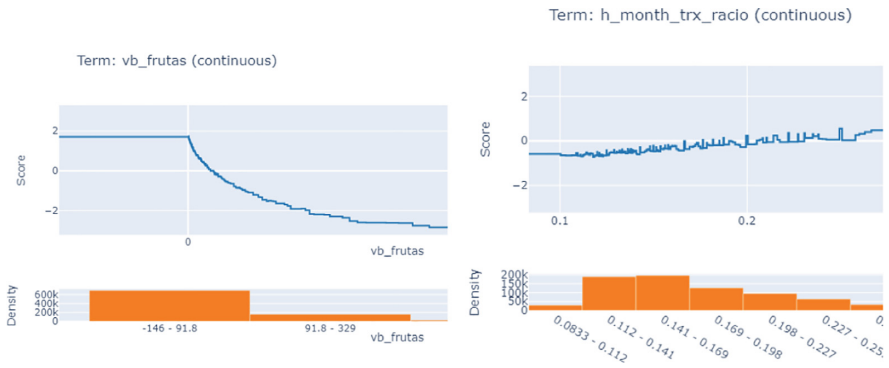


Fig. 11. Snippets of the behaviour for features *vb_frutas* and *h_month_trx_racio*

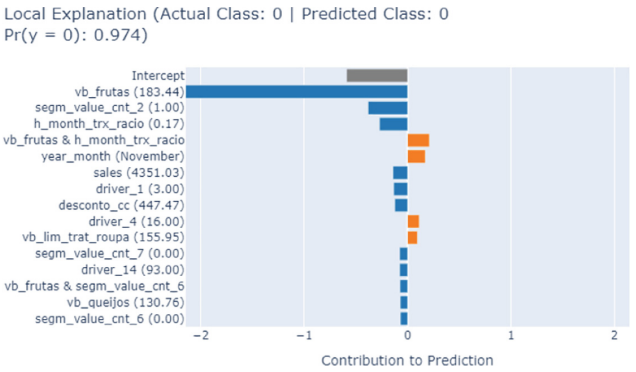


Fig. 12. Local plot for the 3rd observation in the test dataset

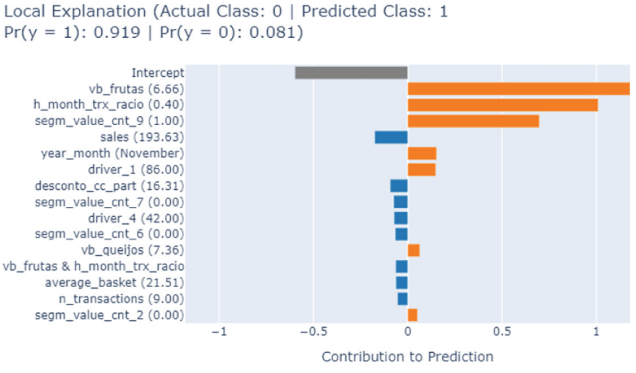


Fig. 13. Local plot for miss classified observation.

91.9%. This kind of errors, even if not optimal, are fairly easy to understand from a decision stand point. Also helps to set the importance of model interpretability for businesses and decision making. This allows the different teams to easily identify factors behind churn and focus on the constructions of micro strategies oriented towards this specific factors for specific customers.

4 Conclusions

As discussed before, the ability to predict churn is of great value for a company. This allows for more efficient retention campaigns and better leverage of customer data. In this work, the proposed approach was to seek accurate predictions for churn while retaining explainability.

In the first phase of the project we set the foundations of the project. Churn and customer were defined by analyzing transactional data for the customers in the retail stores and then clusters of categories were created, going from 196 categories at the start to 6 clusters by the end, in order to decrease the number of churn periods required. Churn periods were calculated and a sample was collected. Secondly, 4 pilot categories were selected and models were trained and its results compared using several metrics. Given it's superior performance over the selected categories EBM was the selected model. All models created had a good accuracy, with values between 70% and 78%, and AUC score, between 82% and 86%, despite some weaknesses in the positive class precision. Even then the models were good in capturing the bulk of the churners.

The low precision in classifying the positive class could be problematic in the future, and so a simple to implement, preemptive work around was created to mitigate this problem, by allowing the classification threshold to be manipulated in order to warrant a more precise allocation of funds, should the need arise. Finally the feature importance was analyzed and it's predictive ability validated by opposing the models decision process to a human decision process with the access to the same data.

5 Future Work

After analysing the variable importance for the models one thing is notorious. The variables the model uses to predict churn are the ones that better describe the behaviour of a bad customer. Low sales, a high amount of total sales in just one month, low value segment, amongst others. This means that the performance of the model in predicting churn for the higher value customers is probably going to be lower, and some light preliminary testing seems to point in that way. This means that in the future a new model should be trained on just this subsection of customers, or if, like in this case, it's the goal to have one model for all of the customers, then the features should be revised. Features that describe the behaviour around competitors, like distance to competitors stores, change in the number of competitors stores in the customer place of residence, can be a good starting point. This approach was attempted during the project, but the lack of suitable data at the time prevented its implementation.

Acknowledgements. A. Manuela Gonçalves was partially financed by Portuguese Funds through FCT (Fundação para a Ciência e a Tecnologia) within the Projects UIDB/00013/2020 (https://doi.org/10.54499/UIDB/00013/2020) and UIDP/00013/2020 (https://doi.org/10.54499/UIDP/00013/2020) of CMAT-UM.

References

1. The Consumer Goods Forum *Portugal: A Retail Snapshot.* The Consumer Goods Forum (2018). https://www.theconsumergoodsforum.com/blog/2018/03/14/portugal-retail-snapshot/. Accessed 18 Feb 2023
2. Idris, A., Rizwan, M., Khan, A.: Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies. Comput. Electr. Eng. **38**(6), 1808–1819 (2012). https://doi.org/10.1016/j.compeleceng.2012.09.001
3. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009). https://doi.org/10.1109/tkde.2008.239
4. Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G., Chatzisavvas, K.C.: A comparison of machine learning techniques for customer churn prediction. Simul. Model. Pract. Theory **55**, 1–9 (2015). https://doi.org/10.1016/j.simpat.2015.03.003
5. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009). https://doi.org/10.1007/978-0-387-84858-7
6. Breiman, L., Schapire, R.: Random forests. Mach. Learn. **45**, 5–32 (2001)
7. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/bf00994018
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. Genet. Program Evolvable Mach. **19**, 305–307 (2017). https://doi.org/10.1007/s10710-017-9314-z
9. Zhang, H.: The optimality of Naive Bayes. In: American Association for Artificial Intelligence. Seventeenth International Florida Artificial Intelligence Research Society Conference (2004)
10. Lemmens, A., Croux, C.: Bagging and boosting classification trees to predict churn. J. Mark. Res. **43**(2), 276–286 (2006). https://doi.org/10.1509/jmkr.43.2.276
11. InterpretML Team: Documentation. interpret.ml (n.d.). https://interpret.ml/docs/getting-started. Accessed 11 Jan 2023
12. Nori, H., Caruana, R., Bu, Z., Shen, J., Kulkarni, J.: Accuracy, Interpretability, and Differential Privacy via Explainable Boosting (2021). http://proceedings.mlr.press/v139/nori21a/nori21a.pdf. Accessed 22 Aug 2023