Explainable Artificial Intelligence for Human-Friendly Explanations to Predictive Analytics on Big Data

by

Joglas do Nascimento Souza

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada

January 2021

Thesis advisor

Dr. Carson K. Leung

Author

Joglas do Nascimento Souza

Explainable Artificial Intelligence
for Human-Friendly Explanations
to Predictive Analytics on Big Data

Abstract

Nowadays, machine learning techniques have become critical for decision-making mechanisms in numerous real-life applications in areas like healthcare, justice, transportation and finance. However, recommendations made by machine learning techniques, as well as their logical reasoning behind these recommendation decisions, are often not easy to be comprehended by humans. This thesis presents an explainable artificial intelligence (XAI) solution that enhances state-of-the-art techniques to produce more understandable and practical explanations to end-users. To evaluate the practicality and usefulness of this XAI solution, a case study was conducted on a big data predictive model built based on real-life customer churn data. Results show that the presented solution successfully provides users with more friendly and useful explanations when compared to related works.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I want to begin by thanking my academic advisor, Dr. Carson K. Leung, who has supported me in all means to achieve the conclusion of my Master's degree successfully. He has always been patient and thoughtful on my learning process in the academic environment.

I also want to thank my advisory and examining committee members, Dr. Pourang P. Irani (Computer Science) and Dr. Liqun Wang (Statistics), for taking the time to read my thesis and for their valuable comments and contributions to make it better.

The whole process of leaving my home country—Brazil—and starting a new life in Canada, and now finishing my Master's degree, would not be possible without the support and love of my wife, Natalia, who agreed to live this dream with me from the very beginning.

I thank my parents and family for their encouragement, love and support in all decisions I made for my life and career. I also thank my old and new friends who were always with me when I needed someone to share my journey's achievements and frustrations. To my lab friends in the Database and Data Mining Lab, I would like to thank them for the discussions and feedback during my research development.

Moreover, I would like to thank DecisionWorks Consulting Inc. for providing the real-life customer churn data used in this thesis. Thanks also to its President, Grant Barkman, for giving valuable insights to make this work happen.

I believe all the people I have just thanked in the sentences above were all gifts from God in my life. And, I feel so fortunate and blessed to have so many people to be thankful for. So, my most important thankfulness is to God that made all this to be possible.

Joglas do Nascimento Souza

AA.Sc.(Data Processing Technology), FATEC São Paulo, Brazil, 2012

The University of Manitoba

January 25, 2021

Chapter 1

Introduction

Nowadays, machine learning techniques have become critical for decision-making mechanisms in numerous real-life applications in areas like healthcare, justice, transportation and finance. However, recommendations made by machine learning techniques, as well as their logical reasoning behind these recommendation decisions, are often not easy to be comprehended by humans. Good quality explanations are needed for humans to trust and collaborate with such intelligent systems [AB18]. The more impact a machine learning decision has on customers or people's lives, the higher is the necessity for its explanation [Mol19]. The need for explanation is not just a demand from end-users but also a regulatory requirement in some countries. For instance, the *general data protection regulation* (*GDPR*) that has been implemented in European countries recently states that customers have the right of explanation for decisions made through automated systems [HHC+19]. Explanations are also a means of detecting model bias as it can reveal when the model is making decisions based on incorrect assumptions. The enhancement of models also relies on how com-

prehensive the model outcome is to propose the appropriate improvements [AB18]. *EXplainable Artificial Intelligence* (*XAI*), the research area that studies how to make models transparent and explainable, is now in the spotlight for keeping the adoption of machine learning growing.

In general, explanations for machine learning models can be broadly classified into two main types:

1. *global explanation*, which aims to give a general explanation considering the whole data population;

2. *local explanation*, which focuses on answering specific questions such as "why a loan was not approved for the customer John?".

The tools available for explanations follow concepts and theories of these two main types. Although there are plenty of tools available for these two types of explanations, most of the output visualization and verbalization provided are not of easy understanding by non-expert users. To address this issue (i.e., to enable non-expert users to understand the output visualization and verbalization), we present an explainable artificial intelligence solution for providing human-friendly explanations to predictive analytics on big data for both expert and non-expert users.

Machine Learning models are commonly classified into two different categories regarding their interpretability:

1. *crystal-clear* models—such as linear regression and decision trees—which are self-explanatory and do not require the application of XAI techniques to explain them.

2. *black-box* models—such as random forest and artificial neural network (ANN)—which can be complex to explain themselves. Consequently, they need XAI techniques to explain the results. The complexity of the black-box models makes them achieve higher accuracy when solving complex problems. Thus, we can say that XAI serves to make black-box models more interpretable, avoiding the trade-off between accuracy and interpretability.

To assure our XAI solution's practicality and usefulness, we conduct a case study on applying the explanations to a random forest customer churn predictive model. Churn is the rate of customers who stopped using a service or product in a given time-frame. The possibility of predicting customer churn can bring a competitive advantage to the business in many different domains such as banking, telecommunications, retail, and education. This kind of strategic knowledge can raise the possibility to prevent and retain potential attrition of customers. Machine learning models have the power to automate the process of identifying those customers, learning from historical data the nuances that differentiate the ones who stopped using a service or product from those who are still loyal. In our particular case study, we used data from customers of a financial institution.

Our solution involves two main components: (a) a back-end component where the machine learning model runs and the explanations' processing occurs, and (b) a front-end component that comprehends the explanation web-interface. Hence, in this thesis work, *our key contributions* include:

- creation of a solution that integrates different techniques to facilitate the use and understanding of machine learning reasoning for non-expert users, and

- enhancements in the way explanations are processed and presented for some of the state-of-the-art techniques.

1.1 Thesis Statement

This MSc thesis aims to design and develop a machine learning explanation solution to integrate and enhance existing methods and present them in an abstraction layer to end-users through a web-interface that they can interact and consume the explanations.

To elaborate, specific questions to be examined and answered in this thesis include the following:

Q1. How to produce more consumable and understandable explanations to end-users?

Q2. Does the simplification of the existing explanation tools help end-users to understand models better?

1.2 Thesis Organization

This thesis is organized in the following way. In Chapter 2, we provide the necessary conceptual background to understand essential aspects of techniques and theories applied in this work. We review related work and how they connect to this work in Chapter 3. The gaps and opportunities for enhancements of the XAI research field are also discussed. Moreover, this research's contribution to filling some of these gaps and participation in the field's evolution.

In Chapter 4, we describe the solution developed for machine learning explanations for predictive analytics. Each one of the components of the solution's architecture proposed is depicted and explained. We divided the components into two main areas: back and front-end. In the back-end, we explain the machine learning model developed to serve as our case study, the explanation engine and the web-framework used to integrate both areas. Details of the techniques applied to the machine learning model for predicting customer churn and how we modified some state-of-the-art techniques in our explanation engine are also described. In the front-end, we showed the web-interface that serves as an abstraction layer to the end-users. We explain and show the functionalities of each of the screens created to give the explanations.

In Chapter 5, we present the methodology for evaluating two pieces of our solution, which are the machine learning model and the web-interface. Classification analysis and results for different experiments are presented for the machine learning model. An objective evaluation based on functionalities and a subjective evaluation through human-subjects are presented for the web-interface.

Finally, in Chapter 6, we end with the research conclusions and discuss future work that is open to extending our study on the explanation of artificial intelligence for predictive analytics.

Chapter 2

Background

In this chapter, we explain some concepts and theories that were used in this thesis work.

2.1 Random Forest

Random Forest [Bre01] is an ensemble algorithm for machine learning classification and regression predictions. It is the evolution of the more straightforward decision tree algorithm. The random forest's ensemble characteristic brings sophistication to this method, as it enhances its prediction power relying on the results of a more significant set of trees. Each tree receives different portions of the dataset, and the final prediction result is the average of the individual's tree results available in the forest [Ere18]. In this thesis, we use the Random Forest because it is a black-box model that requires ways of interpreting the results. Second, it is a well-known and efficient method used to solve many different problems in the literature in terms of

prediction power.

2.2 Sensitivity, Specificity and ROC-AUC

Sensitivity, *specificity* and *Receiver Operating Characteristic Curve* (*ROC-AUC*) are some of the metrics to evaluate the prediction accuracy of machine learning classification models. In the explanations below, it is considered the evaluation of a binary classification, which is the case in this thesis.

Let TP denote true positives, TN denote true negatives, FP denote false positives, and FN denote false negatives. Then:

- *Sensitivity* or *true positive rate* measures the ability of the model to classify the positive classes correctly:

$$sensitivity = \frac{TP}{TP + FN} \tag{2.1}$$

- *Specificity* or *true negative rate* measures the ability of the model to classify the negative classes correctly:

$$specificity = \frac{TN}{FP + TN} \tag{2.2}$$

- *Area Under the Receiver Operating Characteristic Curve* (ROC-AUC) measures the model's ability to differentiate the positive from the negative classes for different thresholds. The threshold is the probability value used for classifying an instance as positive or negative. For instance, the 0.5 or 50% threshold is the default cut-off probability for most cases. When predicted by the model with

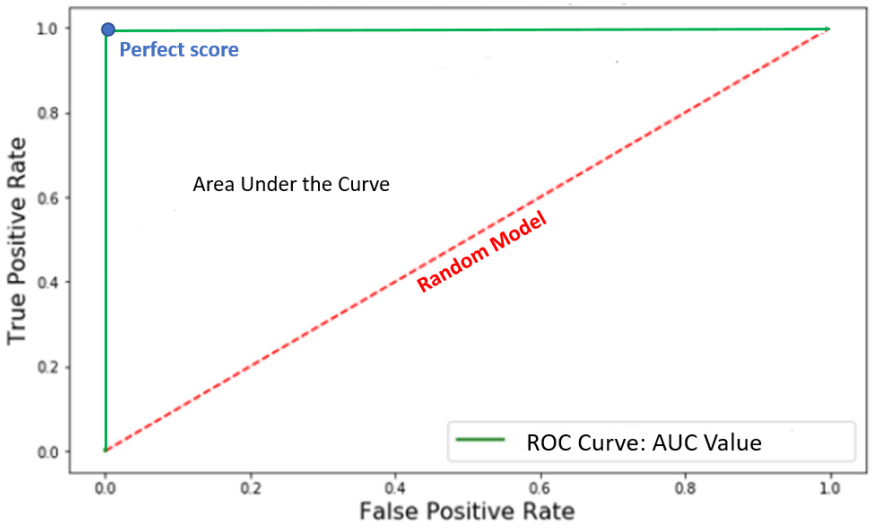


Figure 2.1: The ROC-AUC chart

this probability or greater, an instance is classified as positive and negative otherwise. The ROC curve depicts what happens to the sensitivity and specificity for distinct thresholds. Figure 2.1 shows how the curve is plot. The green line is the ROC curve itself. The red dotted line represents a random model. Points further and to the left of the red dotted line present better results. The perfect score, improbable, happens when the false-positive rate values are 0, and the true positive rate is 1. The area under the curve (AUC) measures the area under the ROC curve; the bigger the area, the better the model.

2.3 Hyperparameter Optimization with 10-Fold Cross-Validation

Machine learning models have a set of hyperparameters that, when optimized, can potentially improve the model's decision boundaries. For instance, in the Random Forest, values such as the number of trees, the tree's maximum depth, the number of samples to split a node, and others can be adjusted. There are mainly two popular ways to find hyperparameters: the *radom search method* and the *grid search method*. The primary difference is that the grid search method would try all the different combinations of hyperparameters in the search space. In contrast, the random method would search for a limited number of randomly selected combinations. Thus, grid search takes longer and is much more computationally expensive. In terms of finding the optimal or close to optimal hyperparameters, Bergstra and Bengio [BB12] have proven that random search can be as good and sometimes superior to grid search when comparing both methods against the configuration of neural and deep belief networks for many different kinds of datasets. Hence, in this research, we chose the random search method to find a hyperparameter set for the predictive model used as a case study.

The search for the hyperparameters has to be applied under the training portion of the dataset. Otherwise, the hyperparameters' learning process would also be based on the testing dataset, creating an over-fitted model that would not have the same performance with new data. In this thesis, we used an algorithm that included the random search method combined with the k-fold cross-validation technique [PVG+11].

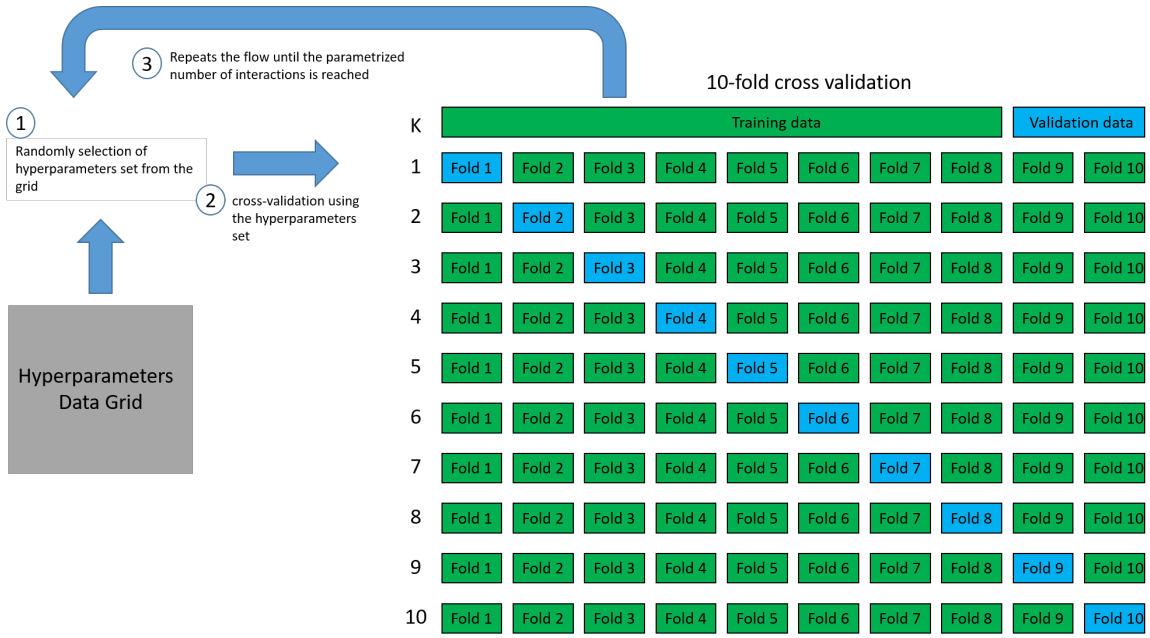


Figure 2.2: Hyperparameter tuning process with 10-fold cross-validation

The k-fold cross-validation is a technique that splits the training dataset in k different folds, and for each of the k runs of training, separate one of the folds to be used as validation. The validation fold is rotated in each run, guaranteeing that the model is validated in different portions of the dataset. Thus, less likely to over-fit.

Figure 2.2 shows the idea of the combined algorithms. The algorithm first selects a random set of hyperparameters from the hyperparameter grid. This grid varies according to the chosen model. Then, for the selected set of hyperparameters, the algorithm cross-validate it using the training dataset. It goes again to the first step until the number of iterations, a configurable parameter of the algorithm, is reached. We chose 100 as the number of iterations and 10 folds for cross-validation. According to a chosen metric, when the algorithm completes the runs, the hyperparameter set that presents the best score is selected. For our model, we used sensitivity as the

metric.

2.4 The Class Imbalance Problem

The "class imbalance" problem has been widely discussed in the literature. It happens in many different domains in which one of the classes in a classification problem is more likely to happen than the other. For machine learning models, this can interfere in the algorithm's learning process as it can be biased towards the class with most instances [PBM09]. Equation (2.3) [LNA17] shows the ratio between the minority and majority classes, the intention is to increase this number for imbalanced datasets:

$$r_\chi = \frac{\chi_{min}}{\chi_{maj}} \tag{2.3}$$

where:

- r_χ is the ratio of the imbalanced dataset,
- χ_{min} is the number of minority class instances,
- χ_{maj} is the number of majority class instances.

There are different approaches to addressing the "class imbalance" problem. The most common are over-sampling, under-sampling and the combination of the two. Over-sampling aims to increase the number of instances for the minority class (χ_{min}). In contrast, under-sampling aims to reduce the number of instances of the majority class (χ_{maj}) [LNA17]. Different techniques can be applied for both methods, varying from randomly creating new copies of the minority and deleting the majority classes'

instances to more sophisticated approaches that rely on algorithms intelligence for creating and deleting instances. The random methods issue is the increase in over-fitting chances when creating instances and under-fitting when losing significant instances [PBM09].

Synthetic Minority Over-Sampling Technique (SMOTE) [CBHK02] is one of the techniques used to increase the number of instances for the minority class. The SMOTE technique generates synthetic data based on the k nearest neighbour instances of a randomly selected minority class instance i. One of the k instances is selected, and the values of the features of i and the selected k are subtracted. The result multiplied by a constant and added to the values of the original features of i is what forms the synthetic instance; Equation (2.4) [PBM09] illustrates this idea:

$$E_{new} = E_i + (E_j - E_i)\delta \tag{2.4}$$

where:

- E_{new} is the synthetic instance,
- E_i is the randomly selected minority class instance,
- E_j is the selected neighbour instance,
- δ is a random constant between 0 and 1.

Adaptive Synthetic (ADASYN) [HYGS08] it is also an oversampling method that generates synthetic instances for the minority class. This technique is an enhanced version of the SMOTE technique discussed previously. The main difference between

ADASYN and SMOTE is that the former weights the minority classes' instances according to their learning difficulty. For example, instances close to decision-boundaries are harder to learn compared to those further away. Thus, ADASYN outweighs the generation of synthetic instances based on original instances that are harder to learn, forcing the machine learning model to learn difficult instances better. In this thesis, we applied both methods to generate synthetic instances and compared the results based on sensitivity and specificity.

Among the methods available for under-sampling, the clustering k-means algorithm is one of the options. Clustering is the process of dividing a set of data into subsets. This division is made based on the similarities and dissimilarities of the data, forming groups accordingly to their intrinsic characteristics [HKP12]. The idea of under-sampling using clustering is to eliminate instances of groups where the cardinalities are higher, keeping the most diversification of data representation possible.

2.5 Shapley Values

Shapley Values [Sha53] is a method based on cooperative game theory that calculates the contribution each player had in the final score of a game. In the context of machine learning, each feature is a player of this game and has a contribution to the final prediction. The contribution of each feature is determined based on its average marginal contribution, calculating how the feature affects the result of the prediction when it is present or not for different coalitions of the remaining features [Mol19].

The average marginal contributions are calculated with the following equation:

$$\phi_i(N, v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\ (|N| - |S| - 1)!}{N!} \left[v(S \cup \{i\}) - v(S)\right] \tag{2.5}$$

where:

- ϕ_i is the average marginal contribution of player i,
- N is the number of players,
- v represents the game,
- S are the sets of different coalitions.

To better illustrate the formula presented, we adapted the example given by Knight [Kni14] that elucidates the Shapley formula and theory using a taxi fare example. Imagine that three friends (A, B and C) share a taxi, and the fare varies according to the distance from the starting point to each home, as illustrated in Figure 2.3. Now, consider that the friends pay their portion of the fare right at the beginning, when they first get in the car. The payment order can vary. For instance, (i) A pays, then B and C, or (ii) B, A and C, representing the different coalitions in the formula above. Table 2.1 depicts what would happen with the fare for each different coalitions. If the friends pay in order of distance, we can observe that A pays \$10, B pays \$10 and C pays \$20 given the total of \$40 that comprehends the amount charged by the taxi driver from the start point to the final destination which is C's home. For the second coalition, we have A, C and B, in which A pays \$10, C pays \$30 and B pays \$0. The average payment for the different coalitions of each friend is what constitutes their marginal contribution. Table 2.2 shows each friend's marginal contribution and the fair amount each one has to pay according to the Shapley Values.

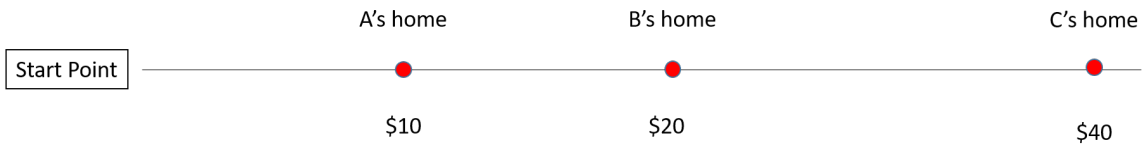


Figure 2.3: Shapley values illustration for a taxi fare.

Table 2.1: Fare share for different coalitions of payment

S	A	B	C
(A,B,C)	$10	$10	$20
(A,C,B)	$10	$0	$30
(B,A,C)	$0	$20	$20
(B,C,A)	$0	$20	$20
(C,A,B)	$0	$0	$40
(C,B,A)	$0	$0	$40

Table 2.2: Average marginal contribution for each friend

	A	B	C
ϕ	$3.33	$8.33	$28.33

The Shapley Values is used in this thesis for global and local explanations. The SHapley Additive exPlanations (SHAP) package [LL17], which we did an adaptation in this work, applies the theory proposed by Shapley Values.

2.6 Summary

This chapter presents an overview of the most relevant techniques and theories that will help understand the remaining chapters of this thesis. We have seen that *Random Forest* is a sophistication of the Decision Tree models, and it has applications for classification and regression predictions. *Sensitivity*, *specificity* and *ROC-AUC* are metrics used to evaluate the accuracy of machine learning classification models. Then, we illustrated the use of the k-fold cross-validation combined with a random search algorithm for hyperparameter optimization. Next, we elucidated the "class imbalance" problem and some alternatives to deal with it. We ended the chapter with the explanation of the theory behind *Shapley Values*, describing the calculation process to define the importance of the features based on their marginal contribution to the model prediction.

Chapter 3

Related Works

In this chapter, we present related works of two macro-areas of the XAI field: Global and Local Explanations. There are different theories, tools, and techniques proposed by different researchers in these two areas. The importance of interpretability and the link between XAI and Human-Computer Interaction (HCI) is also part of this chapter's literature review investigation. Not all the related work described served as the base of this thesis work, but they are worth mentioning to give a general perspective of the XAI research field.

3.1 Importance of Interpretability

It is a consensus among researchers that machine learning explanation is critical and expected by users of many different domains, especially in the more sensitive ones such as healthcare, justice, transportation and finance. The more impact a machine learning decision has on customers or people's lives, the higher is the necessity for its

explanation [Mol19]. Appropriate explanations increase confidence and collaboration between users and machine learning models [YHSA20]. The inclusion of domain users in machine learning explanations is necessary for elaborating explanations that will serve them [ZC18]. Explanations are also a means of detecting model bias as it can reveal when the model is making decisions based on incorrect assumptions. The enhancement of models also relies on how comprehensive the model outcome is to propose the appropriate improvements [AB18].

There are a significant amount of techniques aiming to provide explanations to black-box machine learning models. However, there is still a lack of understandable explanations. Kaur *et al.* [KNJ+20] stated in their research paper that even data scientists could not make an accurate interpretation of some of the explanations given by popular existing methods. These difficulties in understanding the explanations would be even harder for business users who do not have the same technical background. Zhu *et al.* [ZLR+18] stated that existing XAI techniques do not address usability, accessibility, and practical interpretability and are not focused on real users.

There is no consensus among researchers on what interpretability is and how to evaluate it [DK17, Lip18]. Organizations, such as (a) Fairness, Accountability, and Transparency (FAT) and (b) Defense Advanced Research Projects Agency (DARPA), stated that XAI has to provide means for humans to understand, trust and collaborate with intelligent systems [AB18].

As a preview, our solution is applied in a case study of a customer churn predictive model, developed in collaboration with an industrial partner of the financial sector. The importance of interpretability was also a point of discussion during the interac-

tions we had with them. They clearly stated that only machine learning prediction is not enough for most business scenarios in the finance space, and understanding the model's outcomes is crucial for adopting automated systems in their decisions. Our human subjects evaluation in this research will contribute to narrow down the controversial question of what constitutes interpretability for the machine learning field.

3.2 Local Explanation Techniques

Local explanation addresses the interpretability of a specific instance of the dataset. The idea is to understand the reasoning that the model applied to a particular instance. Below are some of the local explanation techniques described in the literature.

3.2.1 Individual Conditional Expectation (ICE)

Individual Conditional Expectation (ICE) plots the interactions for all instances between a selected feature and the dependent variable. It plots individual lines for each one of the instances [Mol19]. It allows checking if there is a common trend among different instances.

3.2.2 Local Interpretable Model-Agnostic Explanations (LIME)

Local Interpretable Model-agnostic Explanations (LIME) [RSG16] is a local explanation technique that uses local surrogate models for its explanations. Surrogate models are more interpretable and simpler models, such as linear models, decision trees, logistic regression and Generalized Additive Models (GAM) [Mol19]. Moreover, a library is made available with different visualization tools for model interpretability using the LIME technique. Recently, researchers [HHC+19, WPB+20, Hoh19] have expanded the number of visualizations tools for local explanations.

3.2.3 Shapley Additive Explanations

Shapey values have applications to global and local interpretation techniques. In this thesis, it is adapted for both scenarios. Moreover, in addition to adapting the Shapey values, we also further improve the Shapey values in enhancing global and local interpretation.

To elaborate, Lundberg and Lee [LL17] proposed in their paper one of the methods used extensively in interpreting machine learning models with the application of Shapley values theory. They proposed SHapley Additive exPlanations (SHAP) as a unified method of identifying feature importance on the predictions, considering the works proposed by other researchers, such as LIME and DeepLIFT [RSG16, SGK17]. With this new unified approach, they compared the new enhanced method in terms of computational efficiency and how intuitively the explanations were to humans. Results showed that the new approximation method proposed in SHAP uses fewer

evaluations to calculate the feature importance, and it has high accuracy. Human subjects' evaluation also showed that SHAP was more intuitive to human understanding than the other methods.

Recently, Kumar *et al.* [KVSF20] argued that Shapley values do not provide human-friendly explanations, which are better satisfied with contrastive explanations. Kaur *et al.* [KNJ+20] evaluated the level of understanding of interpretability tools by data scientists. It revealed that most participants had a wrong interpretation of results and did not use the tools in the way researchers had in mind when they designed them.

In this research, we used the SHAP package that was freely available. As previously explained, their method demonstrated to be superior to other approaches in terms of explanation accuracy and easiness of understanding. However, when Kumar *et al.* [KVSF20] and Kaur *et al.* [KNJ+20] stated that it is still a challenging method to understand by non-expert users, their statement turned to be valid during the interactions we had with end-users on the course of this research. Hence, in this thesis, we improve SHAP to be more human-friendly, and this is further explained in Chapter 4.

3.3 Global Explanation Techniques

Global explanation techniques comprehend the machine learning model's explanation reasoning for most of the patterns learned during the training process. It is useful in cases where explanations regarding the whole data population are needed, such as climate change or economic predictions [YRR19]. There are many differ-

ent techniques in the literature for global explanations, and the most relevant are highlighted in this section.

As a preview, in this thesis research, we use feature importance for global explanations. More details are explained in the remainder of this section.

3.3.1 Feature Importance

The feature importance measures the error increase in the model output when shifts of feature values occur. The difference between the original and new error values indicates the degree of importance each feature has for the model [Mol19]. This method was first introduced in 2001 by Breiman [Bre01]. Fisher *et al.* [FRD19] proposed enhancements on the more classical approach and designed a new method called Model Class Reliance (MCR). They stated that different models could reveal different features important for the same dataset under analysis, a phenomenon known as the Rashomon effect. MCR captures a range of values from a different set of models in order to have an explanation that is not only dependent on one type of model.

The SHAP package previously explained also has techniques available for global explanation of features importance. The same theory for Shapley values applies in this context, with the only difference being that the whole population is considered to calculate the marginal contributions. As MCR, the SHAP for global feature importance is not dependent on the model under analysis, making this technique also immune to the Rashomon effect.

As a preview, in this thesis work, we improve the presentation of the global feature explanation technique in the SHAP package to make the interpretation more

accessible to the end-users. Details will be given in Chapter 4.

3.3.2 Partial Dependence Plot (PDP)

Partial Dependence Plot (PDP) is a visual approach that shows the interaction between one or more features and the model target outcome. This visualization also depicts the level of dependency complexity between the feature and the target [Mol19]. Goldstein *et al.* [GKBP15] stated that the aggregation approach of PDP plots could obfuscate the understanding of the model.

3.3.3 Global Surrogate

Recall from Section 3.2.2 that surrogate models are simpler and interpretable models that can approximate the predictions made by a black-box model. Unlike the use with local explanations, in global explanations, this is used to explain the general model reasoning based on the whole data population. A good example would be to explain a random forest model with a decision tree.

3.4 Contrastive Explanations

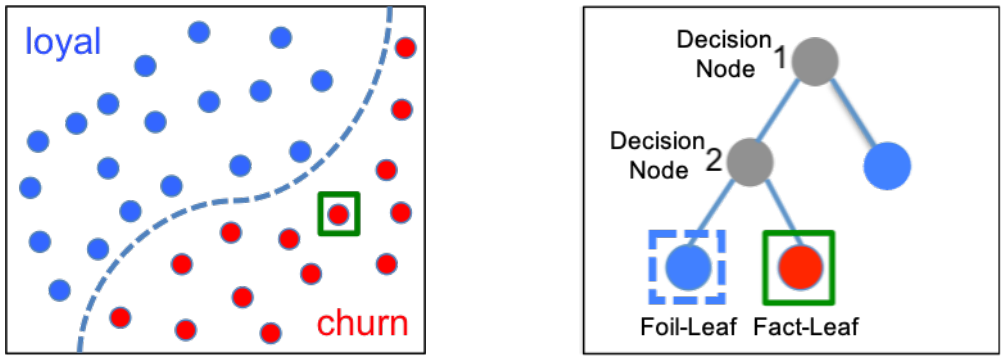
Contrastive or counterfactual explanations refer to techniques that explain the outcome of a specific instance based on what should be done differently for changing the current prediction. It infers the smallest number of changes necessary in the features' values to modify the prediction outcome. This technique is human-friendly as humans naturally tend to use counter-facts to explain facts [Mol19]. Different

from the local explanations described above, the contrastive explanations will not present all the causes that contributed to a prediction but rather explain why the prediction was not the opposite. As an example given by Miller [Mil19] for classifying arthropods, to answer the question "Why the model classified an image as a beetle instead of a spider?" the contrastive technique would be answering by explaining that the arthropod in the image has six legs. In contrast, the traditional local explanation techniques would include all the other characteristics to explain the prediction, such as eyes, strings and wings.

Wachter *et al.* [WMR18] added that for some real-world problems, the counterfactual explanation with the smallest number of changes might not be feasible to turn into action. Moreover, they stated that a higher number of alternative counterfactuals must be available to select the one that better fits reality for these scenarios.

Mothilal *et al.* [MST20] extended the work done by Wachter *et al.* [WMR18] and focused on a method of generating a high number of counterfactual explanations that are feasible and respect real-world constraints.

Dhurandhar *et al.* [DCL$^+$18] proposed an approach called Contrastive Explanations Method (CEM) for contrastive explanations in neural networks that has two main components—namely, *pertinent negatives* (*PN*) and *pertinent positives* (*PP*). PN highlights the features missed in the instance prediction that would change the model outcome, and PP highlights the critical features that contributed to the current outcome. Image and tabular data were inputs for the experiments. Human subjects analyzed explanations for the tabular data outcome, and they evaluated CEM as superior to LIME and Layerwise Relevance Propagation (LRP).



(a) Dataset class boundaries

(b) Decision tree: fact-leaf, foil-leaf and decision nodes

Figure 3.1: Local foil trees method

van der Waa *et al.* [vRv+18] proposed a method entitled *Local Foil Trees* for finding contrastive explanation using decision trees. The approach has two main components: the fact (the true output class) and the foils (the contrastive class). Considering the case study of churning prediction used in this work, imagine that we have the class boundary showed in Figure 3.1a and the data-point highlighted with the green square is the instance we want to generate the explanations. The first step consists of training a decision tree, using the foil class' data-points, in this case, loyal customers. This training method is the one-versus-all approach, in which the decision tree will learn to classify loyal or not-loyal. The data-points closer to the point of interest (fact) have a high weight on the decision tree. With the decision tree trained, the selected data-point (green square) should be input into the trained model and then located in the tree, as shown in Figure 3.1b. Then, the foil-leaf, which has the lowest path from the fact-leaf, is chosen as the foil. In the example in Figure 3.1b, the foil-leaf is represented by the dashed blue square. The last step checks the

differences between the two selected paths' rules and only accounts for the common features between the two, and all the same parent decision nodes are removed. In our example, in Figure 3.1b, node number 2 would be excluded, and node number 1 would be considered to generate the contrastive explanation. The explanation consists of contrasting the values of the fact and foil used on their decision nodes, and the output is in the form "The model predicted A instead of B because feature x is greater than $\langle number \rangle$ and feature y is lower than $\langle number \rangle$."

As a preview, for this thesis work, we enhance and integrate the solution proposed by van der Waa *et al.* [vRv+18] with their Local Foil Trees method. The applied enhancements and why this solution was chosen among the others is further explained in Chapter 4.

3.5 HCI Applications to XAI

The incorporation of Human-Computer Interaction (HCI) techniques into XAI has been discussed in the literature. Wang *et al.* [WYAL19] proposed a theoretical framework that aims to guide model explanations following human reasoning concepts. The framework has four main divisions: human reasoning and the necessity of explanations, how people reason, how XAI generates explanations, and how XAI support reasoning. Miller [Mil19] did extensive research on how the social sciences could contribute to more human-friendly explanations in the XAI field. Abdul *et al.* [AVW+18] conducted research that brings a broad perspective of the current advances in the areas of XAI and HCI. They pointed out that future work between the two areas is promising for the evolution of the model interpretability field and represents

an excellent opportunity for researchers.

As a preview, in this thesis work, we incorporate some of aforementioned findings when designing the user interface. The selection of the contrastive component and how it should be presented in our explanation interface is based on the social research presented by Miller [Mil19], who stated that explanations are contrastive, that people do not ask why something happened but rather why something else did not happen. They described a social experiment in which participants had to wear eye-tracking equipment and observe two colliding balls. For the balls collided situations, participants were asked what they think should have happened to turn it into a different outcome (not collided balls). With the eye-gaze data collected, researchers have shown that participants traced for places that the balls would have gone without colliding, showing that people tend to explain using contrasting causes even in physical experiments. They also stated that people seek a limited number of causes when inquiring for an explanation. To attest to this affirmation, Miller described a second experiment in which explanations were presented for some hypothetical situations. According to their judgment, the participants had to choose the one with the better probability of being accurate and with the best quality. One of these situations was about a person that was experiencing three symptoms: (1) weight gain, (2) fatigue, and (3) nausea. Then, some explanations were given to explain the symptoms. The first one explained the weight gain due to stopping exercising, the fatigue to mononucleosis and nausea to a stomach virus. The second explained that the symptoms were due to a pregnancy. The third one explained the three symptoms as a combination of the causes explained in the first explanation but without attributing a specific

cause to each symptom. Results showed that participants preferred more straightforward and fewer causes explanations, such as the pregnancy one. For this reason, we searched for a contrasting explanation method that produces fewer causes, and we also limited the number of causes (features) used to explain the global and local explanations. Our interface's human subjects evaluation was based on human-grounded metrics proposed by Doshi-Velez and Kim [DK17]. The details of the interface will be addressed in Chapter 4.

3.6 XAI User-Interfaces

Hohman *et al.* [HHC+19; Hoh19] proposed interactive user interfaces (GAMUT & TELEGAM) to explain local and global explanation of classification predictions generated by Generalized Additive Models (GAM). The evaluation of the interface was through human-subjects, recruiting machine learning experts and practitioners. Results showed that the interface improved data professionals' capacity to interpret the model results. Most participants expressed a high interest in having a tool like the one proposed in their daily activities.

Collaris and van Wijk [Cv20] proposed an interface called ExplainExplore to provide explanations based on surrogate models. The interface allows the users to select from a different set of models, the one they want to use as a surrogate. The tool provides local and global explanations and the capability to choose neighbour data points if there is a need to look for a better explanation of a similar instance. Results showed that data scientists understood and identified problems on the models they were unaware of before.

Adams and Hagras [AH20] used a proprietary tool called Temenos to explain the outcome of the predictions of their Fuzzy Logic model. This tool has similar ideas to the interface proposed in our work based on the description and visuals shown in the paper and information from the company website. Temenos also has the purpose of serving non-expert end-users, but it seems to be business-specific with a focus on the banking sector.

As explained in the last section, Wang *et al.* [WYAL19] proposed a theoretical framework to guide model explanations following human reasoning concepts. For testing the theoretical framework effectiveness, they also developed an XAI explanation dashboard applied to a medical diagnosis model. As they did not name their interface, we refer to it as the XAI Diagnostic in this thesis. As they had domain-expert clinicians to test their interface, they had a practical experiment of the proposed theoretical framework. With the user's evaluation, they could see the flaws and improvements needed on the explanations provided.

As a preview, we incorporate some of the aforementioned suggestions into our work, such as access to the raw data and a reduced number of counterfactual explanations for each instance. The latter suggestion follows Miller [Mil19] findings discussed previously.

3.7 Summary

This chapter presented the most relevant work on the XAI field, directly and indirectly, related to this thesis work. We first addressed how researchers identify the interpretation of models as an essential topic for the evolution of machine learning

adoption, and at the same time, the lack of consensus on what characterizes proper interpretation.

We have seen that explainable AI has two main areas of explanation techniques: Global and Local Explanations. Global Explanation techniques aim to give a general explanation based on the whole data population of a dataset. On the other hand, Local Explanation answers questions for specific instances such as "why the model denied a loan for customer A?". Among the different techniques for Global Explanation, we can highlight feature importance, which lists the most relevant features contributing to the model decisions. Hence, in this thesis work, we focus on SHapley Additive exPlanations (SHAP) and Contrastive Explanations, proposing the necessary improvements to them.

We also discussed current works linking Human-Computer Interaction (HCI), social science and XAI. Theoretical and practical works were presented and how we applied those findings in our work is also a topic of discussion.

To end the chapter, we presented similar XAI interfaces to the one we propose in this work. We also point out some functions we incorporated into our work based on these previous works' flaws.

Chapter 4

Our Explainable AI Solution for Predictive Analytics

In this chapter, we present the solution developed to integrate and enhance different XAI techniques abstracted in a web-interface to explain machine learning predictions. The solution's architectural pieces, their integration and building, are also part of this chapter. The machine learning model used as our case study for predicting customer churn is one of these pieces, and the steps involved in building it are explained.

4.1 Overview

Recall from Chapter 3 that some researches have shown that current XAI tools do not provide easy to understand explanations. Another aspect of these tools is that they require programming knowledge to manipulate the libraries that the tech-

niques were implemented. With this in mind, we proposed a solution that integrates different explainable approaches in a web-interface that creates an abstraction layer between the methods and the non-expert users. We also enhanced some aspects of the generation of explanations for the available state-of-the-art techniques.

The system proposed in this work has two main parts: back-end and front-end components. The back-end comprehends the background solution's architectural piece, where the machine learning prediction runs, and the processing and generation of the explanations happens. The front-end component brings the interface that the users interact with for understanding the model reasoning.

4.2 Back-End Component

In the architecture designed to link the pieces of our proposed solution, as shown in Figure 4.1, the back-end component is where resides the data preprocessing and model run, the processing of the explanations and storage of the results, and the web-framework to integrate both ends.

To attest to our proposed solution's explanation capabilities, our **machine learning model**, sub-component Ⓐ in our proposed architecture in Figure 4.1, predicts a financial institution's customer churn as a case study. Churn is the rate of customers who stopped using a service or product in a given time-frame. The possibility of predicting customer churn can bring a competitive advantage to the business in many different domains. This kind of strategic knowledge can raise the possibility to prevent and retain potential attrition of customers. Machine learning models have the power to automate the process of identifying those customers, learning from historical

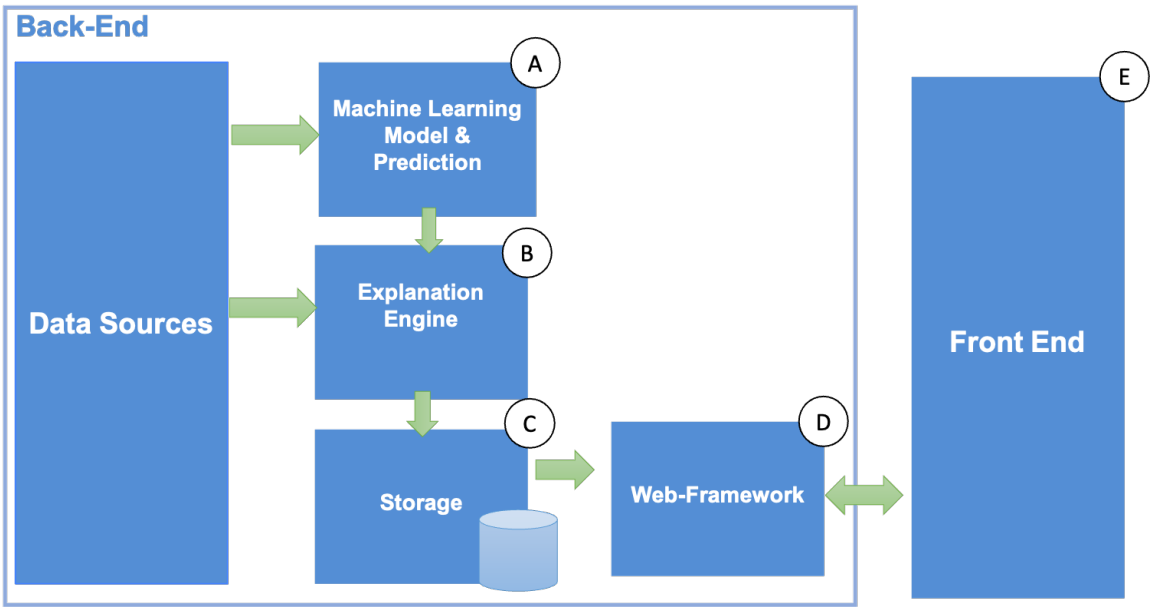


Figure 4.1: The proposed solution architecture.

data the nuances that differentiate the ones who stopped using a service or product from those who are still loyal. We used a Random Forest model for our case study because it is a black-box kind of model and, second, due to its good performance on various solutions. We built a data pipeline to get the data ready for the model predictions, which involved some phases:

- Data exploration,
- Data labeling,
- Data preprocessing,
- Over-sampling and under-sampling

Data exploration is the first step in building the machine learning model proposed in this work, addressing the objectives of getting to know the data, raising hypotheses,

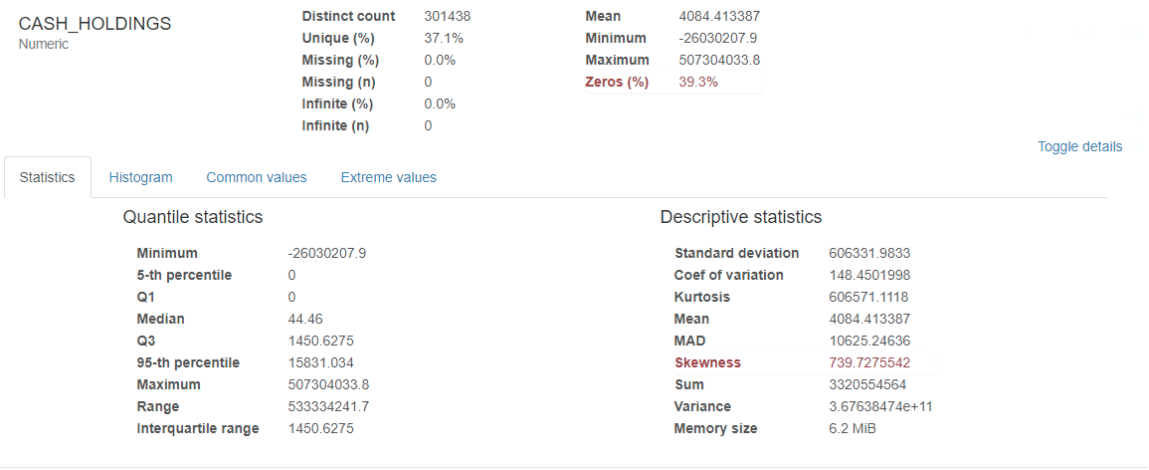


Figure 4.2: Example of data exploration report

and identifying data issues. We generated reports with information such as the number of variables, records, percentage of missing values, variables types, and descriptive statistics (e.g., standard deviation, mean, median, histograms, maximum and minimum values) to leverage the understanding of the dataset quality. Figure 4.2 shows an example of a report containing this information. Visualizations for understanding the distribution of the data and how this is affected between loyal and churned customers are also advantageous for enhancing the understanding of the datasets and elaborating hypotheses worth trying during the model's development.

Data Labelling is the process of labelling datasets, where there is no trivial way of separating the classes that compose them. To determine whether a customer churned or not can vary depending on the services or products provided. For instance, it can vary from when customers close their accounts to when they stop doing transactions even with an open account in the financial sector. In an online educational platform, churn is when the student cancels his subscription. As the definition of churn is broad

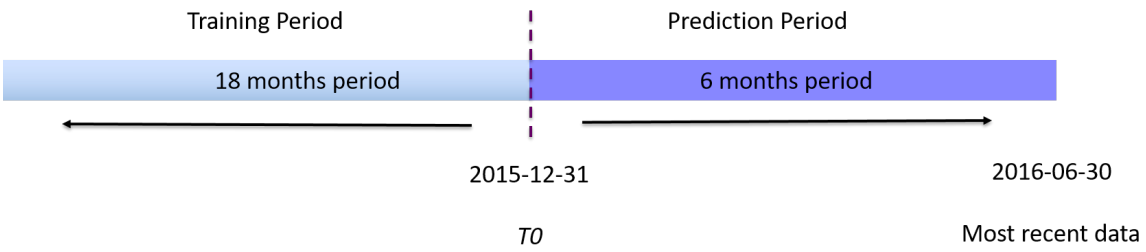


Figure 4.3: Slicing of time-series data for the dataset labelling

and subjected to the business's intrinsic characteristics, there are high chances of the dataset not being labelled with the classes. As a result, it is necessary to establish a definition of churn and label the data accordingly. Since the definition of churn is linked to the business domain, its determination also has to surge from domain specialists. This interchanging of knowledge conveys a characteristic of most of the data-related projects, where technical and domain expertise has to come together for building the solution.

Another common characteristic of churn related cases is that the data has a time-series behaviour. Usually, customers build a history of transactions on time before leading to the churn. This characteristic is also essential when labelling the dataset as determining if a customer churned or not depends on the time-frame under analysis. Figure 4.3 shows an example of how our case study data was sliced to have a consistent prediction period and to train the model to predict into the future. The most recent six-month period, which comprehends the period after $T0$ until our most recent data, was reserved for our testing dataset. When labelling, we also excluded from the testing dataset any customers previously churned in the training period, avoiding the model to cheat. For the training period data, all customers were labelled following

the domain specialists specified definition of churn.

The *data preprocessing* phase is one of the most critical stages in the pipeline of building a machine learning model. In our case study, this phase required a significant amount of time for cleaning, engineering features and aggregating the datasets. For data cleaning, we eliminated any instances and columns that were impossible to recover using imputation techniques. These techniques consist of filling missing data with assumptions based on the remaining data, imputing calculated values, such as medium, median, or regression. For the instances in our dataset with the possibility of recuperation, the appropriate technique was applied. Following a business recommendation, we dropped any customers with total investments below a certain amount from our dataset. They are not exciting customers in terms of behaviour to be analyzed. Figure 4.4 shows the total number of customers among churned and loyal before and after the data cleaning.

To engineer the features, we grouped them into three different categories:

- *demographics*, which comprehends information such as age, tenure, and profession.

- *life events*, which accounts for information of events that happened on the customer's lives, such as a change in income, profession or marital status.

- *portfolio performance*, which are features to measure how well or bad the customers' investments are performing.

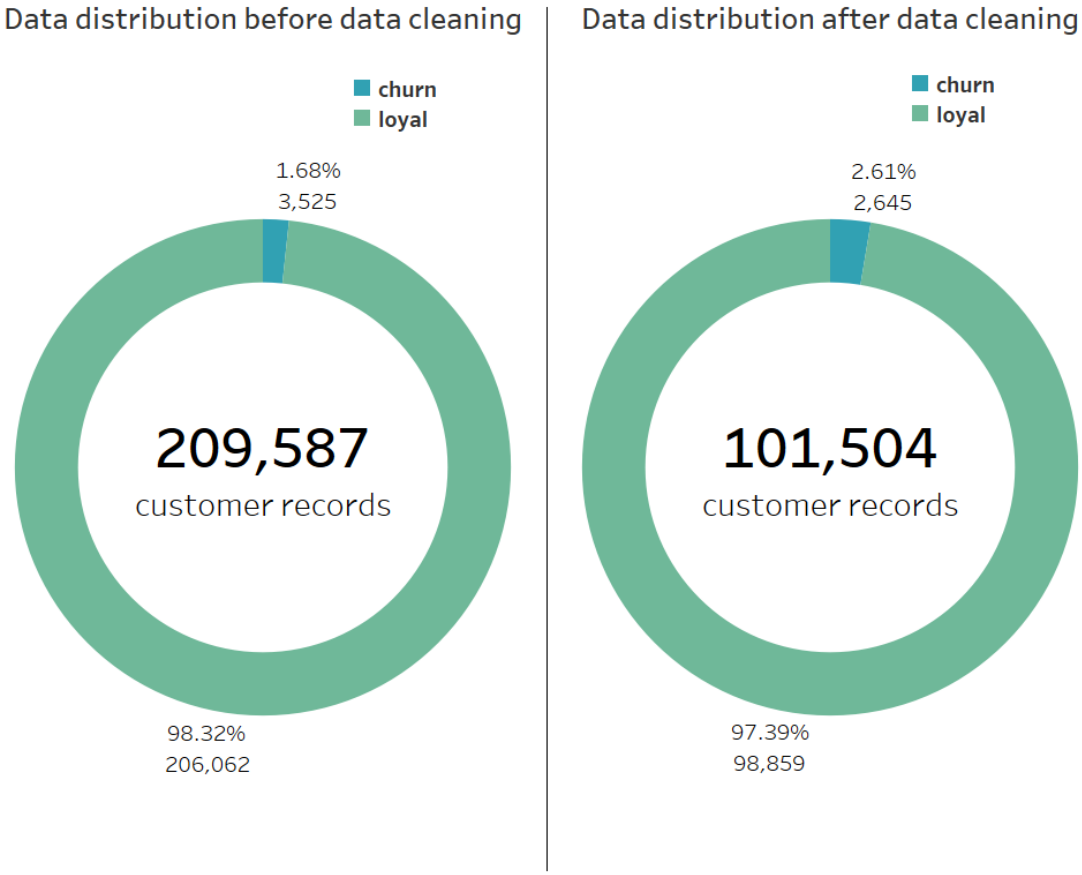


Figure 4.4: Data distribution before and after data cleaning.

Table 4.1: Most important features

Feature	Description	Type	Category
Total Assets	The sum of equity values that belongs to the customer in T0	Numeric	Portfolio Performance

Table 4.1 (continued from the previous page)

Feature	Description	Type	Category
Growth Rate Ratio	Rate of growth in portfolio for the customer in the period of analysis. Given by the formula: $\frac{LPV - IPV}{IPV}$ where: LPV = Last Portfolio Value IPV = Initial Portfolio Value	Numeric	Portfolio Performance
Draw Down Ratio	The ratio between the peak and lowest values of portfolio value for the customer in the period of analysis. Given by the formula: $\frac{PPVP - LPVP}{PPVP}$ where: $PPVP$ = Peak Portfolio Value $LPVP$ = Lowest Portfolio Value	Numeric	Portfolio Performance
CAGR Moving Avg.	Compound Annual Growth: rate of return from an investment from the beginning to its end. Given by the formula: $\left(\frac{LI}{II}\right)^{\frac{1}{n}} - 1$ where: LI = Last Investment II = Initial Investment n = Number of years	Numeric	Portfolio Performance
Annual Income	The customer's annual income	Numeric	Demographics
Tenure	The customer's tenure	Numeric	Demographics
Age	The customer's age	Numeric	Demographics
Gender	The customer's gender	Categorical	Demographics
Marital Status	The customer's marital status	Categorical	Demographics

Table 4.1 (continued from the previous page)

Feature	Description	Type	Category
Profession	The customer's profession	Categorical	Demographics

Table 4.1 shows the most important features selected as inputs to the model and their description, type and category. As will be discussed in the result section, selecting the most important features did not affect the model performance. Moreover, it represents a meaningful improvement in model interpretability.

Over-sampling and under-sampling is the phase where we address the "class imbalance" problem described in Section 2.4. As previously shown in Figure 4.4, the distribution of the classes between churned and loyal customers is highly imbalanced in our case study. In most problems, as it is in our case, the class of interest is usually the one that is rarer to happen. Thus, we applied various techniques for dealing with it: SMOTE and ADSYN for over-sampling and k-means clustering for under-sampling. These techniques are also described in Section 2.4. There is not a theoretical approach to determine the magnitude applied to over-sampling or under-sampling. Therefore, we tried different combinations of ratios and techniques, selecting the better performed for our case study dataset. Before employing the techniques, we had a ratio of approximately 1 to 58 in our training set; in other words, 58 non-churn cases for each churn case. The different configurations of ratios tested and the comparison of the model performance before and after applying the methods will be discussed in Section 5.3.

The phases described previously led the data ready to be input into the Random Forest model. The dataset was split into 70% training and 30% testing. We also applied stratification when splitting the dataset, meaning the distribution of each

class (churn and loyal) was proportional between the two sets. The training set is where we applied the over-sampling and under-sampling techniques earlier explained. We performed a random search 10-fold cross-validation technique, as described in Section 2.3, for the different ratios and methods applied to over and under-sampling. It is essential to mention that we isolated the synthetic generated data points for being used only on the training folds, avoiding overfitting. With this approach, we were able to find the best set of hyperparameters and cross-validated it using the sensitivity as a metric. Once the model was trained, we then measured its performance with the testing set's predictions, further explained in Section 5.5. Finally, this led us to be ready to extract the explanations from the model.

The **explanation engine**, which is sub-component Ⓑ in our proposed solution in Figure 4.1, is where the processing of the explanations take place. The results are stored in a database (sub-component Ⓒ) to be retrieved by the front-end interface. The two primary processing are for the Shapley Values, which explain the prediction for specific instances and the Contrastive Explanations for the model recommendation.

Our **web framework**, which is sub-component Ⓓ in our proposed solution in Figure 4.1, integrates the explanation engine storage results with the front-end interface. The web-framework is also responsible for applying any necessary rules on the data before presenting it to the front-end component. One of the well-known paradigms to decouple each one of these parts of the web-framework is the Model, View and Template (MVT):

- Model: represents the database component.

- View: contains logic and actions performed by the server. This layer interacts with the model, applies any necessary logic to the data and returns the results to the templates.
- Template: contains the interfaces.

4.3 Front-End Component

The front-end, which is component Ⓔ in our proposed architecture in Figure 4.1, contains the web-interface itself, where the users can interact and search for interpretations of specific instances or global explanations. As previously explained, this corresponds to the abstraction layer between the users and the techniques. The front-end component is divided into some screens:

- Home & expected loss,
- Local feature importance,
- Global feature importance, and
- Model recommendation.

4.3.1 Home & Expected Loss

For most business domains, a machine learning model that only classifies instances is not enough. For instance, knowing that a customer will churn or if an employee is going to leave the company is not sufficient for setting a strategic plan of action. There is a need to set strategies and priorities based on the probability of an event

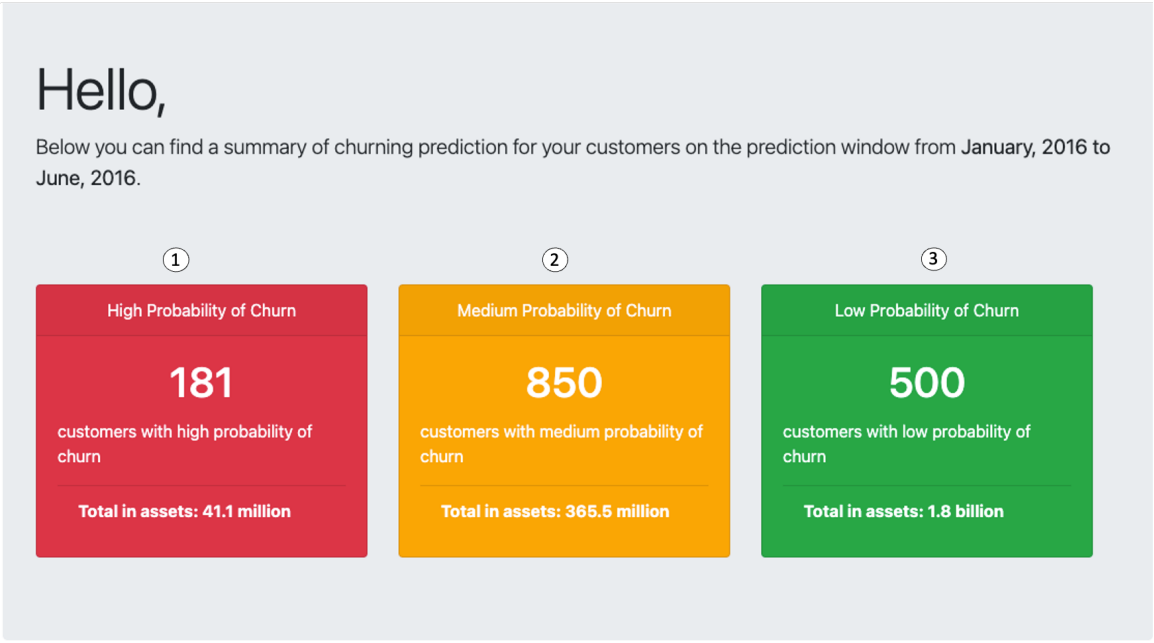
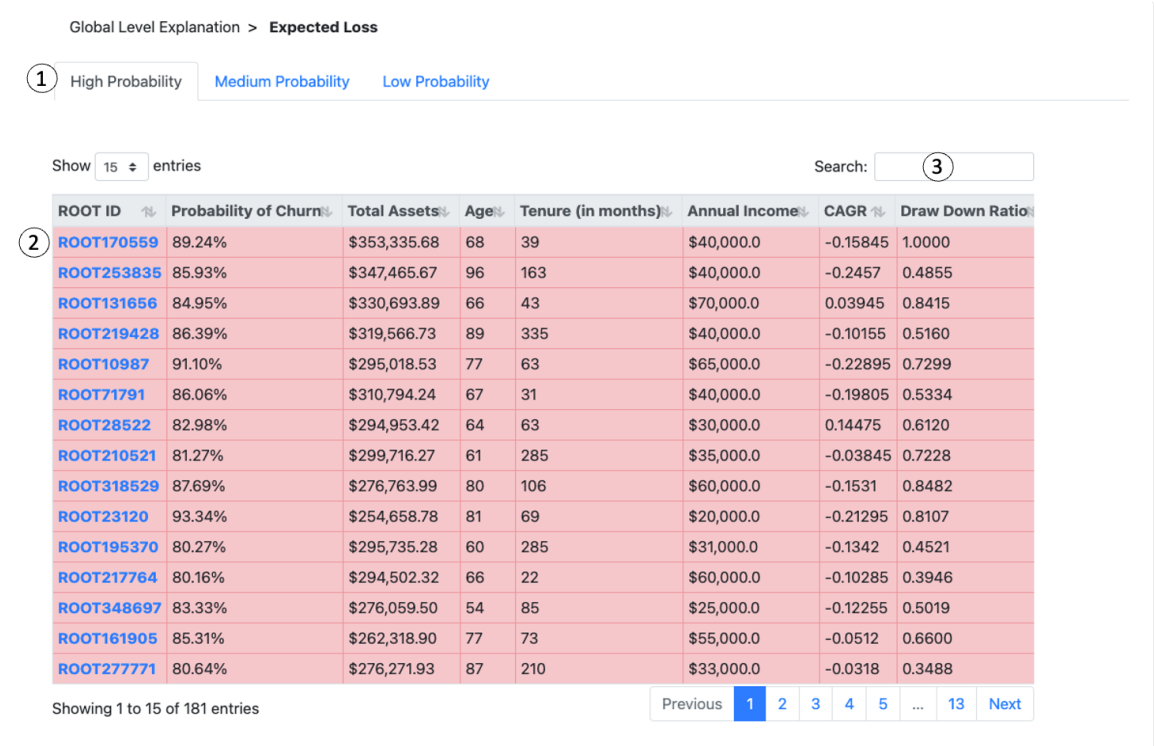


Figure 4.5: The Home screen. Boxes ①, ② & ③ with summaries of customers in each group of risk and total monetary value they represent. These boxes are also clickable leading to the corresponding tab in the Expected Loss screen (as shown in Figure 4.6).

to happen and its monetary impact in case it turns into reality (e.g., list Client A with $20M investment and having 80% churn probability before Client B with $0.2M investment having 90% churn probability). Visual colours to differentiate instances based on the event probability serves to elucidate the appropriate kind of actions.

The "Home" and "Expected Loss" screens allow decision-makers to prioritize their actions according to the expected loss, which results from the product of the probability of churn predicted by the model and the monetary value the customer has for the company. Figures 4.5 and 4.6 show the screens applied to our case study grouping customers according to the risk of churn, which is high, medium or low. For the



ROOT ID	Probability of Churn	Total Assets	Age	Tenure (in months)	Annual Income	CAGR	Draw Down Ratio
ROOT170559	89.24%	$353,335.68	68	39	$40,000.0	-0.15845	1.0000
ROOT253835	85.93%	$347,465.67	96	163	$40,000.0	-0.2457	0.4855
ROOT131656	84.95%	$330,693.89	66	43	$70,000.0	0.03945	0.8415
ROOT219428	86.39%	$319,566.73	89	335	$40,000.0	-0.10155	0.5160
ROOT10987	91.10%	$295,018.53	77	63	$65,000.0	-0.22895	0.7299
ROOT71791	86.06%	$310,794.24	67	31	$40,000.0	-0.19805	0.5334
ROOT28522	82.98%	$294,953.42	64	63	$30,000.0	0.14475	0.6120
ROOT210521	81.27%	$299,716.27	61	285	$35,000.0	-0.03845	0.7228
ROOT318529	87.69%	$276,763.99	80	106	$60,000.0	-0.1531	0.8482
ROOT23120	93.34%	$254,658.78	81	69	$20,000.0	-0.21295	0.8107
ROOT195370	80.27%	$295,735.28	60	285	$31,000.0	-0.1342	0.4521
ROOT217764	80.16%	$294,502.32	66	22	$60,000.0	-0.10285	0.3946
ROOT348697	83.33%	$276,059.50	54	85	$25,000.0	-0.12255	0.5019
ROOT161905	85.31%	$262,318.90	77	73	$55,000.0	-0.0512	0.6600
ROOT277771	80.64%	$276,271.93	87	210	$33,000.0	-0.0318	0.3488

Figure 4.6: The Expected Loss screen. ① Tabs to separate the instances according to the group of risk. ② There are links for each of the instances leading to the corresponding explanation in the Local Explanation screen (as shown in Figure 4.7). ③ The search function enables users to search for specific instances.

example used in this work, this represents 80% or higher, between 50% and 79% and lower than 50%, respectively. These values are parameterized according to specific needs.

4.3.2 Local Feature Importance

The "Local Feature Importance" screen brings for each instance the attributes that contributed positively and negatively to the model prediction outcome. Using

our case study as an example, this is the screen where through visualizations, there are explanations for why the model predicted that a particular customer has high, medium or low probabilities of churning.

In Section 3.5, researchers have shown that people have better comprehension with fewer causes explanations. Thus, we applied a feature selection technique to select the most important features, and they are shown in decrease order of importance in the y-axis of the tornado plot ① in Figure 4.7. The user can also search for specific instances or switch between them in table ② provided. This table facilitates the interaction between the user and the interface, allowing them to access the raw data to understand a customer profile and change between instances easily.

The calculation of the importance of each instance's features in this screen is based on Shapley Values. Recall from Chapter 3 that the SHAP package calculates the importance of the features based on their marginal contribution, using the Shapley Values theory. When presenting the SHAP value results for end-users, we noticed that they had a hard time following the numbers' meaning. It requires understanding the intuition behind the Shapley Values theory. Also, the labelling style and the chart's shape available in the library brings challenges for them to interpret it. Hence, a more straightforward way to comprehend the numbers and the chart was necessary. First, we normalized the marginal contribution results for the features set for each one of the instances i. The normalization scaled the values in a range between -1 and 1, as shown in Equations (4.1) and (4.2):

$$standard_i = \frac{\text{Features List[i]} - \text{Features List.min()}}{\text{Features List.max()} - \text{Features List.min()}} \tag{4.1}$$

$$scale_i = standard_i \times (maxRange - minRange) + minRange \tag{4.2}$$

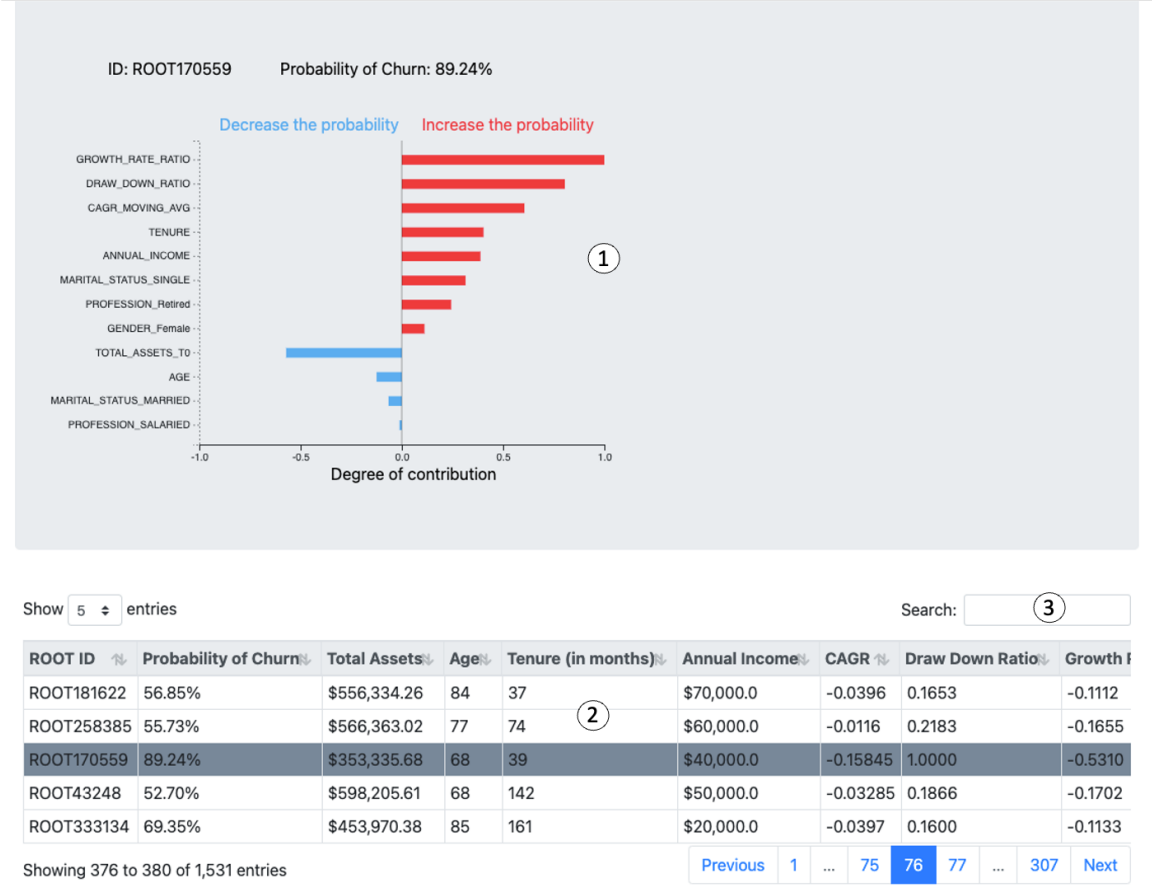
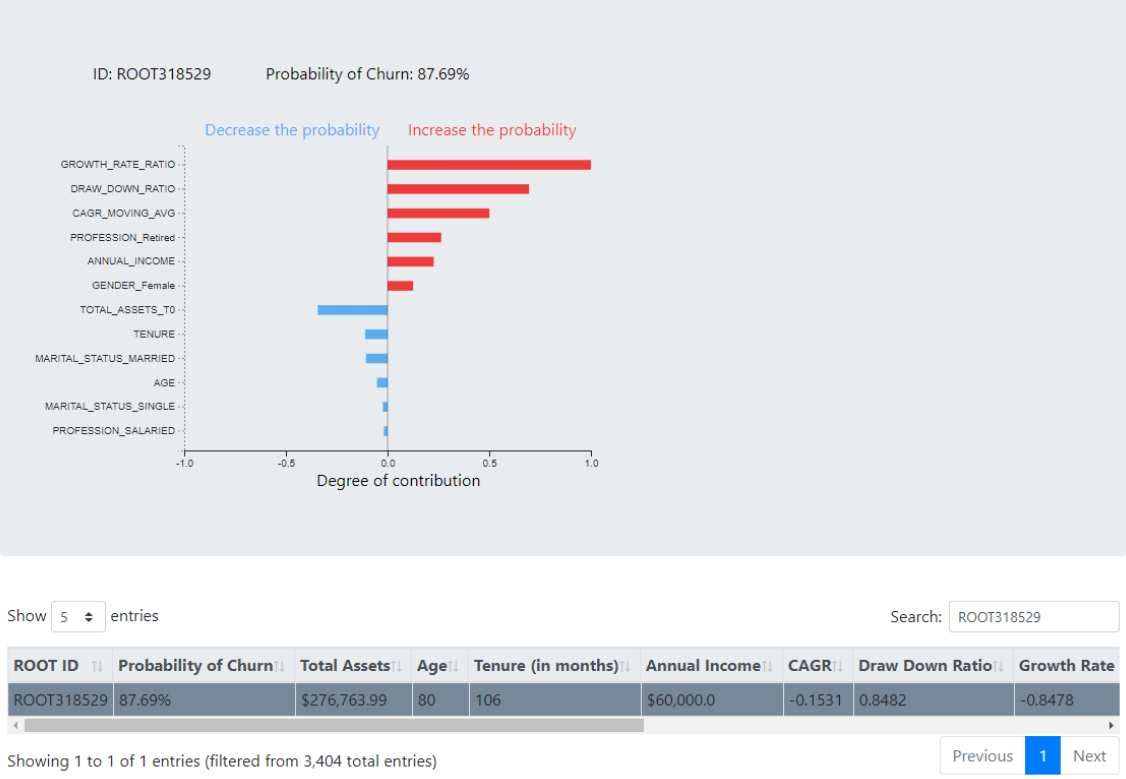


Figure 4.7: The Local Feature Importance screen. ① The tornado plot displays the positive and negative features contributing to the model outcome. ② The table allows users to change the explanation to another instance and also visualize the feature values for the instances. ③ The search function enables users to search for specific instances.

On this scale, -1 means a high negative contribution and 1 a high positive contribution. Although the Shapley Values calculation results are presented in a normalized scale for enhancing interpretability, each feature's degree of importance in a given explanation is preserved. Second, we created a tornado plot with the new range of

(a) The Shapley values chart in SHAP package



(b) The modified Shapley Values chart

Figure 4.8: Comparison between the available Shapley values chart and our modified version

values in the x-axis, showing the negative or positive contributions the features had in the model prediction outcome. Figure 4.8a shows the original chart available in the SHAP package library, and Figure 4.8b shows an example of our modified version.

4.3.3 Global Feature Importance

The "Global Feature Importance" screen gives the user a general panorama regarding each feature's relevance for the model when reasoning the classification decisions. This screen gives quick information on the set of features, from the most to the least important, that the model reasoned its decisions. It can also serve as an agile way to identify biased models.

The calculation of the importance of the features is also based on the Shapley Values theory. The normalization of the values follows the same explanation given in Section 4.3.2, with the only modification being in the range of the values that in this case ranges from 0.0 to 1.0 (0 meaning no contribution, and 1 a high contribution). Figure 4.9 shows details of this screen.

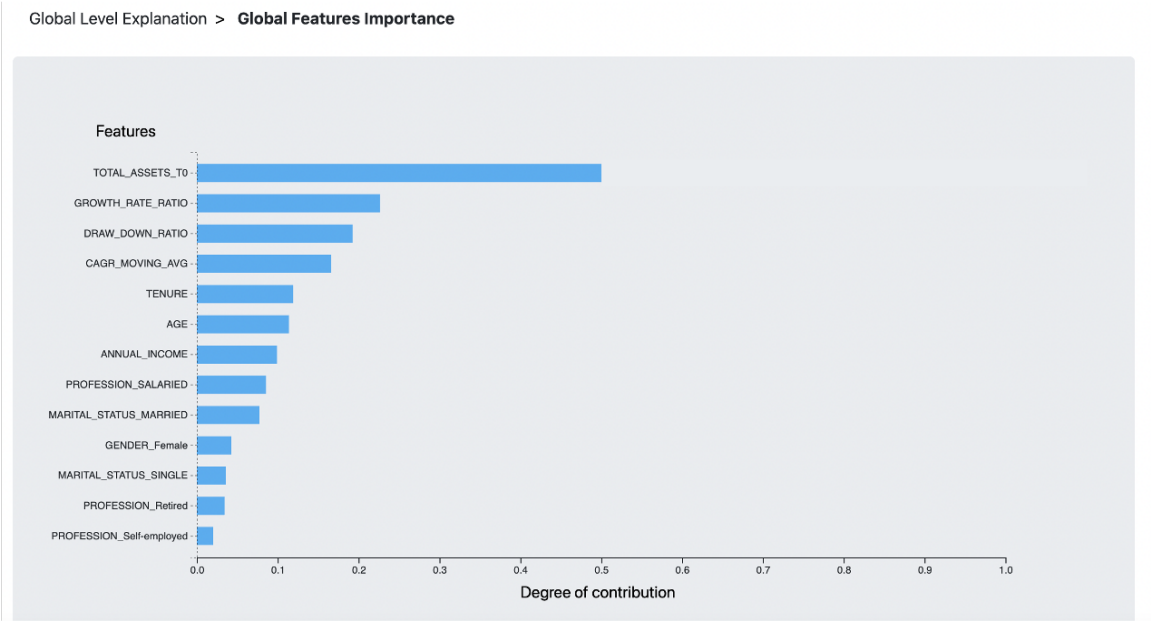


Figure 4.9: Global feature importance screen

4.3.4 Model Recommendation

In the "Model Recommendation" screen, we did not use visualization but verbalization to explain the instances. The information given on this screen can serve two different purposes. First, it can be used as an alternative explanation other than the one provided by the "Local Feature Importance" screen. Second, as the screen's name states, it can serve as model recommendations to the end-users. As the output explanation is given with the necessary changes to modify a prediction outcome, users could use this, when possible, as a recommendation on what to do to prevent a customer from leaving the company, for instance. The example showed in Figure 4.10, where it says that one of the factors the model predicted the customer as churn is due to the "growth rate ratio" lower than 1.37. This ratio measures the growth in the customer's investments. Such an explanation could trigger the financial advisor to give the customer a call to check if he was satisfied with his investments' growth level or if he wanted to make changes in his portfolio. This kind of proactive action has the potential to prevent a customer from leaving the company.

The recommendations given in this screen is based on contrastive explanation using the *Local Foil Trees* technique discussed in Chapter 3. Among the different techniques proposed in Chapter 3, we chose to enhance and integrate the Local Foil Trees technique because it is model-agnostic, which means we can apply the technique for a diversity of models. Second, there is no synthetic data points generation, as it uses the contrastive class's actual data points. Lastly, the form of explanations is easy to understand.

In our proposed solution, we improved some aspects of the Local Foil Trees' ex-

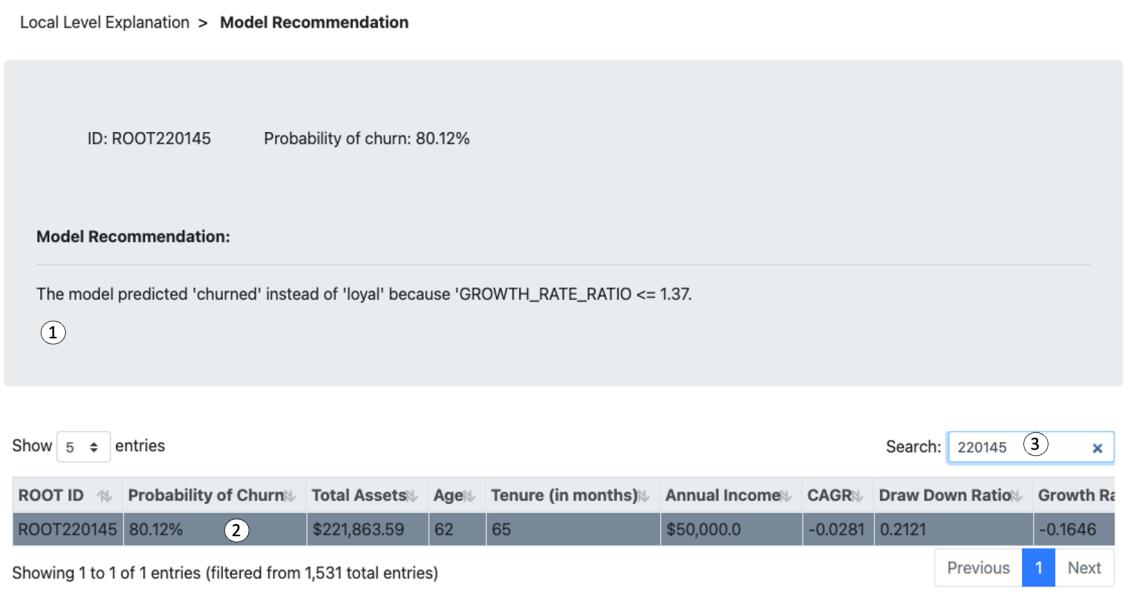


Figure 4.10: Model Recommendation screen. ① The verbalization explanation contrasting the fact and the foil. ② The table allows users to change the explanation to another instance and also visualize the feature values for the instances. ③ The search function enables users to search for specific instances.

planation. We formatted the numerical features' values and adapted the output explanation for categorical variables that were previously also based on numbers. The explanations for categorical features are now in the format "The model predicted A instead of B because customer ⟨is or is not⟩ categorical feature value."

4.4 Summary

In this chapter, we presented the machine learning model that served as a case study for the solution developed in this work. The data pipeline phases and the techniques involved in the model development were described. The pipeline was

divided into four phases: data exploration, data labelling, data preprocessing and over and under-sampling techniques. In the data exploration phase, we presented the methods used to explore the datasets using descriptive statistics and visualizations. Next, we discussed the process of labelling the data into the classes churned and loyal according to a business definition. In the data preprocessing, we described the data cleaning and feature engineering processes. Finally, we explained how we addressed the "class imbalance" problem with the over and under-sampling techniques.

The web-interface created as an abstraction layer between the non-expert users and the explainable artificial intelligence techniques was also presented. We detailed the solution architecture designed to link the back-end and front-end components. The enhancements applied to the state-of-the-art explanation methods are also part of the contents of this chapter. We showed the screens created for each explanation and how that was applied to our machine learning case study predictive model for customer churn.

Chapter 5

Evaluation

In the previous chapter, the two primary components (back and front-end) of our solution and their pieces were presented. To attest to our solution's effectiveness, we need first to evaluate our proposed architecture in terms of implementation feasibility. Second, evaluate the machine learning model that served as our case study and certify that the generation of the explanations is being produced for a reasonable model—lastly, the evaluation of the explanations presented in the web-interface, which involves objective and subjective approaches.

5.1 Overview

To evaluate the machine learning model's predictive capacity for customer churn with Random Forest, we conducted some experiments. The dataset was split into two sets, T_{train} and T_{test}. T_{train} has 70% of the data, and it is used in the training phase, where the 10-fold cross-validation was applied. T_{test} has the remaining 30% of the

data, and it is used to test the model against unseen data and test its performance to new data emulating what will happen in reality.

5.2 Evaluation Setup

To evaluate the proposed architecture's effectiveness and feasibility described in Figure 4.1, we applied well-known technologies for each of the proposed sub-components and implemented a functional solution. Figure 5.1 depicts the architecture with each of the technologies used to implement our proposed architecture.

Machine Learning Model & Prediction Ⓐ. To implement our machine learning model and prediction, we used Python as the programming language, the Pandas package for the data exploration, preprocessing and labelling. The Scikit-Learn package was used for the machine learning capabilities.

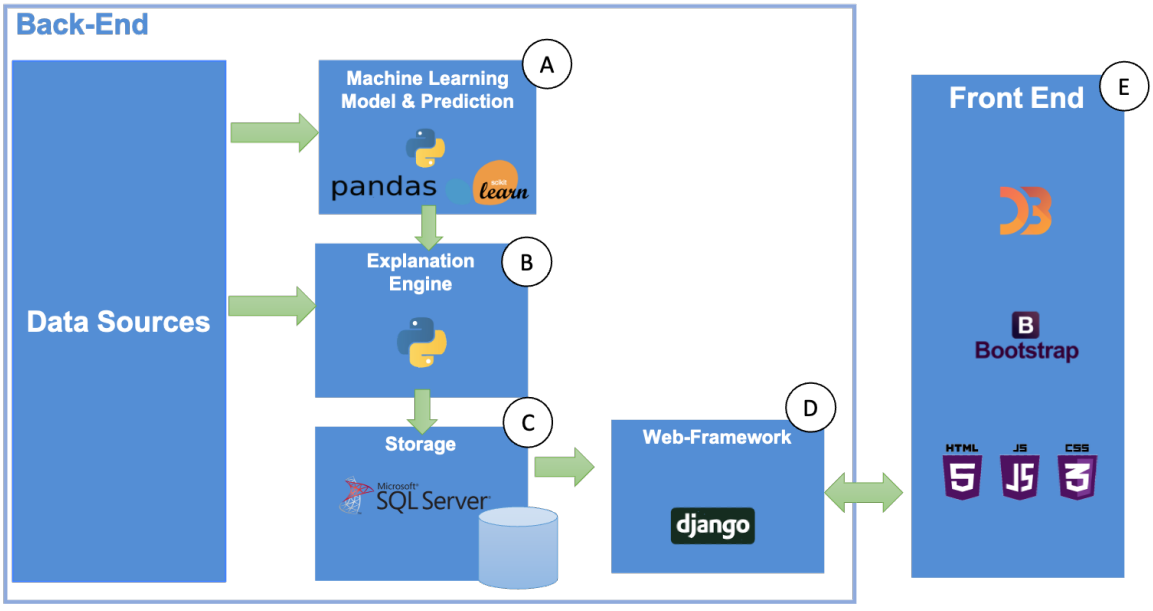


Figure 5.1: The evaluation of the proposed solution architecture.

Explanation Engine Ⓑ. The explanation engine was implemented in Python and our modified versions of the SHAP and Local Foil Tree packages.

Storage Ⓒ. For storing the results produced by the explanation engine, we used SQL Server.

Web-Framework Ⓓ. For implementing the Web-Framework, we used Django. It provides automation for many of the components that programmers need to develop for a website. Django's architecture is based on the Model, View and Template (MVT) paradigm previously explained in Section 4.2. In Django, the MVT paradigm is implemented in the following way:

- Model: represents the database component. In Django, each table corresponds to a model representation.
- View: the logic and interactions with the model and template layers are implemented in Python.
- Template: contains the HTML pages that are presented to the users.

Figure 5.2 shows the interaction between the layers described above. All results saved by the explanation engine are accessed through the model layer, then presented to the users with the interactions between the view and template layers.

Front-End Ⓔ. For implementing the interfaces, we used a mix of web technologies such as HTML, JavaScript, CSS, Bootstrap and D3.js. The proposed architecture shows to be feasible and functional when implemented. The set of technologies, if desired, can be replaced by others that perform similar functions.

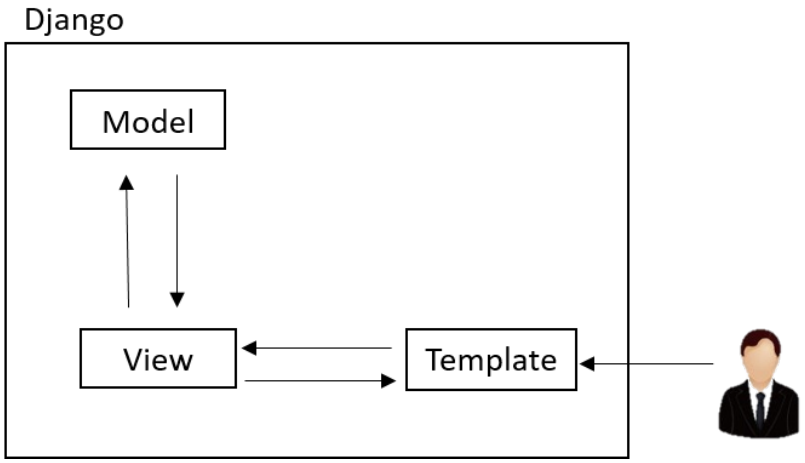


Figure 5.2: Django MVT paradigm; adapted from Dauzon *et al.* [DBR16, p. 6]

5.3 Over and Under-sampling Ratio Test

The two steps needed for re-sampling the dataset used in our case study involve determining the ratio values to balance the classes. Recall from Section 2.4 that the technique used for under-sampling is based on k-means clustering and for over-sampling on SMOTE and ADASYN.

Table 5.1 shows the trials of the different experiments for finding the balance ratio that could leverage the model's learning capabilities to differentiate the classes. The first row shows our baseline without the application of the re-sampling techniques. We can observe that the model performed poorly on identifying positive cases (churn) and exceptionally well to identify negative cases (loyal), given sensitivity and specificity, respectively. The model is biased towards the majority class. Like many other cases, having high true positives (i.e., correctly predict churn customers) and high true negatives (i.e., correctly predict loyal customers) are desirable. False positives may lead to unnecessary efforts and resources to retain customers who were not considering

Table 5.1: Classification model performance for the test set for different classes' ratios

Method	Ratio		Performance		
	Positive classes	Negative classes	Sensitivity (T_{test})	Specificity (T_{test})	Sensitivity (T_{train}) cross-validation
Baseline	1,859	69,160	65%	93%	67%
Clustering only	1,859	26,557	73%	81%	77%
	1,859	**18,590**	**81%**	**60%**	**81%**
Clustering & SMOTE	5,311	26,557	69%	87%	70%
Clustering & ADASYN	5,311	26,557	77%	69%	78%
	7,967	26,557	77%	69%	79%
	5,577	18,590	84%	48%	85%
SMOTE	20,748	69,160	67%	95%	67%
	34,580	69,160	67%	95%	67%
ADASYN	20,748	69,160	66%	69%	73%
	34,580	69,160	69%	84%	68%

leaving or cancelling service or product. False negatives lead to not taking actions to maintain a customer and losing him for the competitors, causing monetary loss. Thus, although the sensitivity is more important in our context, a reasonable specificity is also desirable. With this in mind, we experimented different ratios and evaluated the results towards a good sensitivity and moderate specificity. From the results in Table 5.1, we can observe that the best outcome for sensitivity, 84%, is reached when combining clustering and ADASYN but with the cost of a low specificity of 48%. Our second-best result for sensitivity is 81% when applying the undersampling clustering technique, and 60% for specificity. As the second-best result is slightly worse in terms of sensitivity but significantly better in terms of specificity, this is the ratio we chose to apply to our dataset. This configuration ratio also has the same sensitivity result of the cross-validation in the training phase, which characterizes a non overfitted model.

5.4 Feature Selection Evaluation

Recall from Chapter 3 that researchers have shown that humans have a better understanding of explanations with a low number of causes. Our initial model, used for balancing classes in the dataset in Section 5.3 had a total of 29 features. Thus, in this Section, we tested our model, reducing the number of features and evaluating the different numbers' impact on the model's performance. The goal is to find the minimum number of features that maintain or improve our model result regarding sensitivity and specificity.

We used the *Recursive Feature Elimination* [PVG+11] technique to select the features. It first trains the model with all features. The features are then ordered based on the degree of importance each one has for the model learning process. The least important feature is pruned, and the model retrained. The whole process repeatedly starts until the threshold, which is the desired number of features, is reached. Table 5.2 shows the experiments for the different number of features. We started with 10 features, and the results were worse than with the initial total of features. As we increase the number of features, the results get better until there is no difference in increasing the number of features. For instance, 13 or 14 features do not make any difference. Thus we stop the experiment with 13 features that present

Table 5.2: Sensitivity and specificity for different number of features

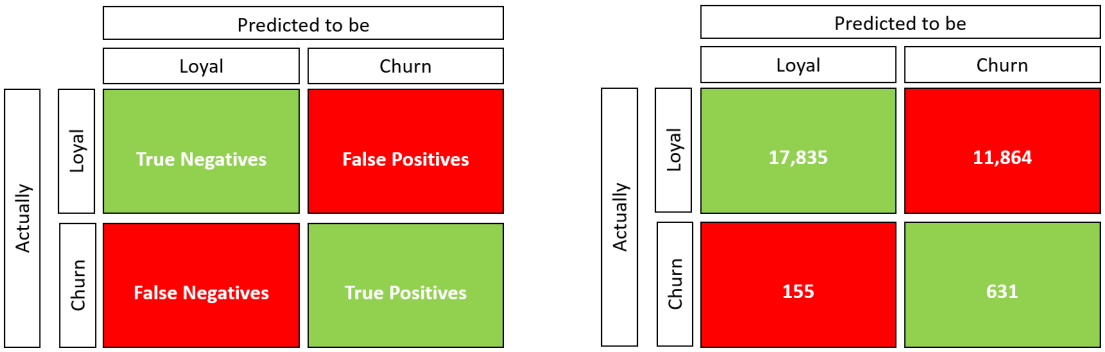
Number of Features	Sensitivity	Specificity
10	77%	66%
11	78%	62%
12	79%	61%
13	**81**%	**60**%
14	81%	60%

the same results as the initial number. With the application of this technique, we were able to reduce the number of features by 44%.

5.5 Random Forest Model Evaluation

The confusion matrix evaluates a model classification accuracy in terms of true and false positives and true and false negatives. Figure 5.3a shows how this confusion matrix is structured. The columns and rows represent the matches or mismatches between what the model predicted and what the reality is. When there is a hit between the prediction and reality, we have a true positive or negative and a false positive or negative otherwise. The results for sensitivity and specificity presented in the previous section were calculated based on the confusion matrix we will depict in this section.

Figure 5.3b shows the confusion matrix results for our predictive model. We



(a) Confusion matrix structure

(b) Confusion matrix outcome for our customer churn model

Figure 5.3: Confusion matrix

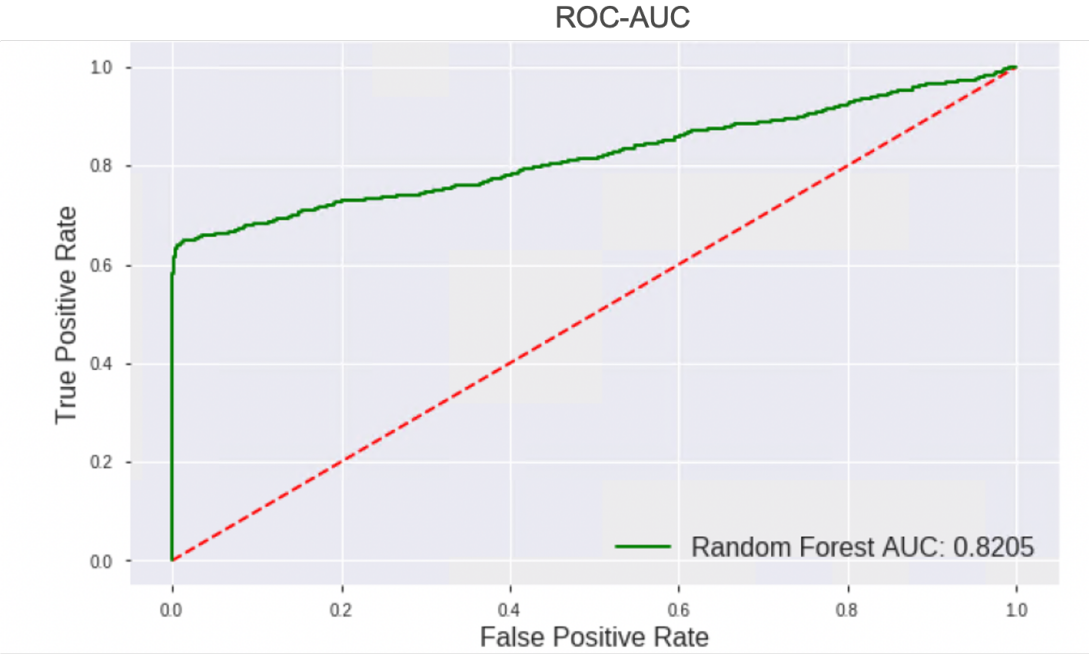


Figure 5.4: ROC-AUC

can compute the sensitivity, specificity, and ROC-AUC with the true positives and negatives and false positives and negatives numbers. The calculation of sensitivity, specificity, and ROC-AUC with the formulas explained in Section 2.2 is 81%, 60% and 82%, respectively. The ROC-AUC is presented in Figure 5.4.

5.6 Human Subjects Evaluation of the Interface

5.6.1 Participants

The study instructions were sent to a pool of 15 potential participants of mixed backgrounds and experiences. From this pool of potential participants, 9 people volunteered to participate in the study. All participants declared to have some familiarity

with the financial sector. The 9 participants have at least a bachelor's degree, and from those 4 also have a graduate degree. When asked about their previous experience with machine learning models, all participants declared to have it, and 6 participants declared to have experience also with explanation tools.

5.6.2 Study Design

Each potential participant received in their e-mail instructions about the research, a consent form, the link to our web-interface, a link to a Q&A forum where they could post questions anonymously, and a manual to guide them through the web-interface. Participants were asked to follow some steps: (1) read the consent form, (2) read the user manual, (3) access and use the web-interface freely, and (4) answer the survey. The questions in the survey were divided into three different sections:

- *Demographic questions*, where questions regarding their level of education, profession, experience in the financial sector and machine-learning were asked.
- *Interface evaluation*, questions about usability and usefulness of the screens developed for the interface.
- *Level of understanding*, questions regarding unlabeled customers where participants had to predict churn or loyal according to a given chart to attest their level of understanding of the explanations.

5.6.3 Results

Usefulness. To evaluate the usefulness of each one of the screens we proposed as explanation pieces in this work, we asked the participants if they thought that each particular screen was useful. Figure 5.5 shows the results for this question. The "home" and "expected loss" screens were evaluated as useful for all the participants. Regarding the local and global explanation screens, we can observe that 90% of the participants found the local explanations useful, against 80% for the global explanations. It was expected to have similar results between these two kinds of explanations, as they are constantly mentioned in other studies to serve different purposes but equally important. The "model recommendation" screen presented a modest result among the explanation methods, which around 69% of participants considered useful. The technique adopted in the "model recommendation" screen, contrasting explanations, was the least used and evaluated in the literature due to its novelty. It might explain its less popularity as there is room to explore it and improve explanations. However, this research shows that this technique is promising, as most users still found it useful.

Easiness of understanding. Participants were asked if they found the explanations easy to understand. Figure 5.6 shows the results regarding this question. For the "home" screen, 90% of participants evaluated it as easy to understand. For the 10% that disagreed, some commented about not being clear that the boxes were clickable and the lack of explanation of what constitutes customer churn. Both information was given in the user manual, but it seems some participants would prefer to have it in the interface itself. The "expected loss" screen was approved by 80%

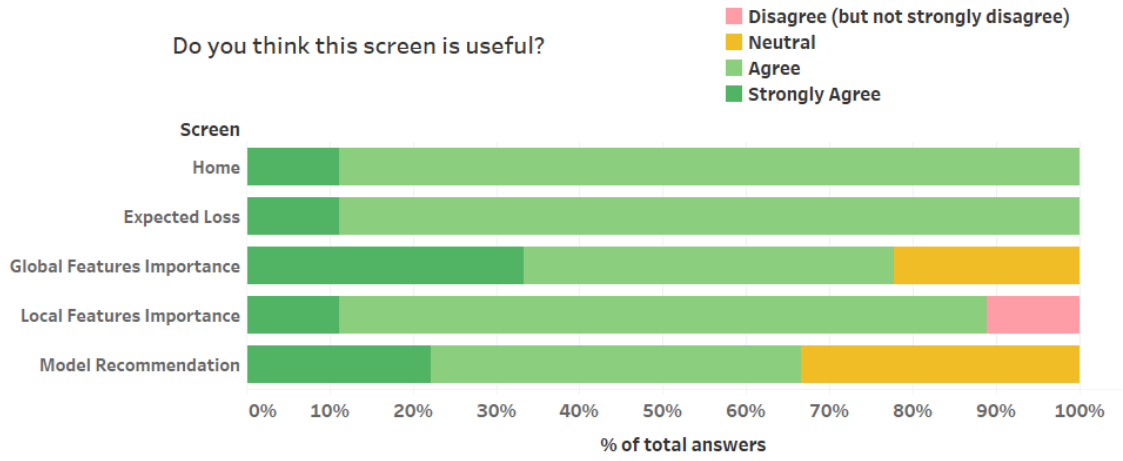


Figure 5.5: Usefulness evaluation

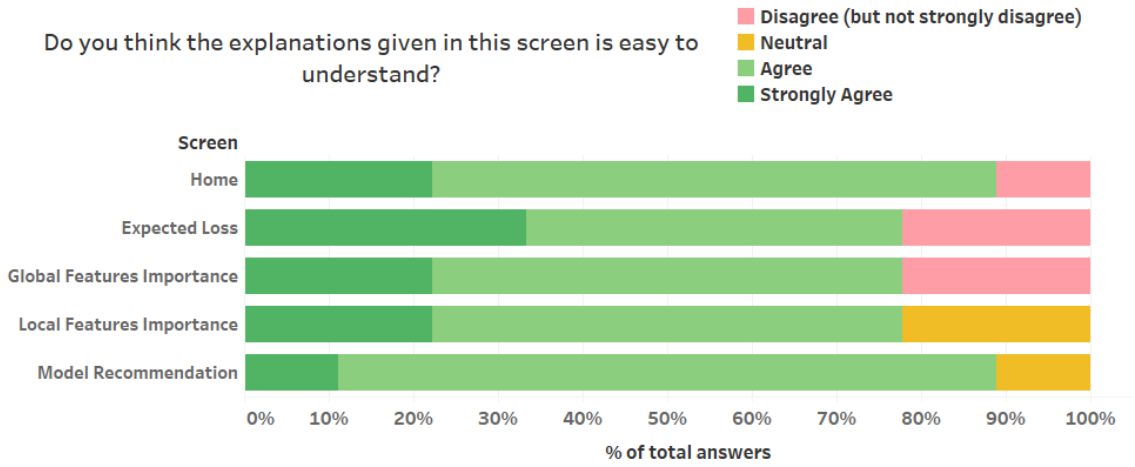


Figure 5.6: Easiness to understand evaluation

of the users. One of the participants commented that it would be better to have the list sorted only by the likelihood of churn, without combining it with the monetary impact. A second participant stated that it would be interesting to have different colours highlighting outliers. Global and local explanations screens were evaluated the same, which 80% of participants found them easy to understand. There were comments about integrating the data dictionary to the interface, as we provided it on

the user manual only. The "model recommendation" was better evaluated than local and global explanations, achieving 90% of approval. This better evaluation might be explained by the explanations given in plain English, not requiring visualization interpretability skills. One of the participants commented that the concept of providing specific actions to reduce customer churn is great.

Binary choice. Recall from Section 4.3.2 that we modified the available visualization presented in the SHAP package to make it easier to understand and interpret it. Figure 4.8a showed the available approach and Figure 4.8b our modified one. To evaluate if our modified version achieved the objective to be more interpretable, we asked participants to choose between explanations given for the same customer in each one of the approaches. All participants chose our modified approach as a better way to explain the model's reasoning for specific customers.

Search capability and interaction. We asked participants if they found the search functionalities and the interactive selection of customers in the table provided easy to use. Figure 5.7 shows the results for this question. We can observe that the majority of participants, around 90%, found it easy to use.

Level of understanding. To evaluate the understanding of the most complicated visualization in our solution, which is in the "local explanation" screen, we quiz the participants regarding the probability of churn of two customers without showing the churn likelihood or features' values. We showed two visuals in which participants had to choose if the customers had a high or low probability of churning. All participants classified the customers correctly. It confirmed that the visualization was easy to

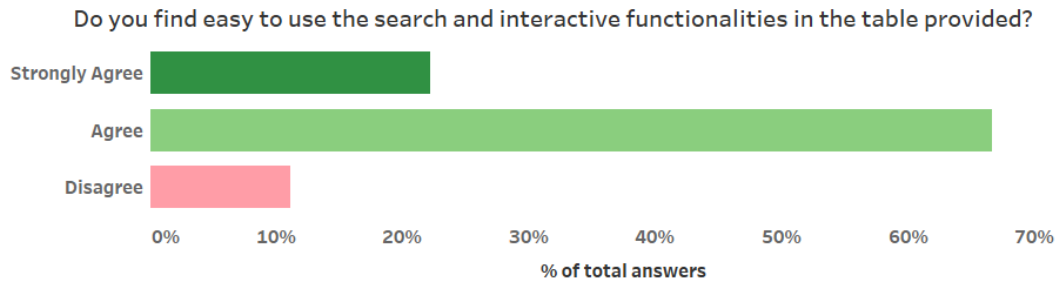


Figure 5.7: Easiness to use search and interactive functionalities

understand, and the user manual described it effectively.

5.7 Objective Evaluation of the Interface

A comparison in terms of the number of functionalities available between our proposed solution and existing ones is essential for a more objective evaluation. We selected a list of functionalities to be evaluated due to one of the outcomes of this work, which was to determine the most relevant techniques available in the literature for having satisfying explanations of machine learning predictive models. As we can observe from Table 5.3, there is a comparison between the solutions based on the different types of explanations available, target instances capability, search functionalities, and audience.

Recall from Section 3.5, we described the XAI Diagnostic [WYAL19] solution, which is the one that also focused on non-expert users when producing the explanations. Figure 5.8 shows the interface developed by Wang *et al.* This solution has no Global Explanation techniques in the interface. For local explanations, the middle chart uses Shapley Values to show feature importance, the same technique we use in

Table 5.3: Comparison between our proposed solution and existing ones

Solution	Functionality					Audience
	Global expl.	Local expl.	Contrastive expl.	Search table	Target instance capability	
ExplainExplore [Cv20]	√	√			√	Experts
TELEGAM [Hoh19]	√	√	√		√	Experts
GAMUT [HHC+19]	√	√	√	√	√	Experts
XAI Diagnostic [WYAL19]		√	√		√	Non-experts
Our proposed solution	√	√	√	√	√	Non-experts

our solution. The model used for explanation is a multi-class model, with five different classes. Unlike our solution, appropriate labelling and order based on the features' importance are not presented in the proposed chart. Wang *et al* stated that some users felt that each feature in isolation was responsible for a prediction, which can be attributed to the lack of appropriate labels. There are counterfactual explanations in the bottom part of the dashboard, which shows many different scenarios that could be formed with different feature values. Our solution focused on a technique that shows only one contrasting scenario for each instance, making it not overwhelming to the end-users. The XAI Diagnostic solution does not have an easy way of selecting instances based on feature values or filtering capabilities as the interactive table in our solution provides.

GAMUT [HHC+19] offers the same set of functionalities as our solution, but with differences in how the explanations are given and focuses on data scientists and machine learning practitioners. Figure 5.9 shows the GAMUT interface. The Global Explanation Ⓐ is given by Partial Dependence Plots (PDP), a visual approach that shows the interaction between one or more features and the model target outcome. GAMUT shows for each feature its interaction with the model outcome. In our

Figure 5.8: XAI Diagnostic [WYAL19]

solution, we showed all the features and their level of contribution in one chart. In the same charts used for the Global Explanations, GAMUT presents the possibility of creating counterfactual explanations built by their users. For instance, if a user wants to check how a different value for a specific feature would affect the model, he could select the desired value in the chart and check its effect. In our solution, we

Figure 5.9: GAMUT [HHC+19]

already present counterfactuals with the set of features and values that would change a prediction. For the Local Explanation Ⓑ, GAMUT uses a waterfall chart that shows how each feature contributed to the specific outcome. In the example given in the figure and in the paper, the explanations are for a regression model to predict house prices, and they showed how much a feature value added or subtracted to the house predicted price. They also stated that the interface was also tested in classification models, but it does not mention how it affects the explanations. In our solution, we use a tornado plot and show the degree of contribution, either positively or negatively, each feature had for the model outcome. Finally, the interactive table Ⓒ is very similar to the table we have in our solution. We observe that GAMUT's solution requires more technical expertise, as the users need to understand PDP plots and

hypothesize counterfactuals they want to test. In our solution, we also divided the screens for the specific kind of explanations, making it clear to the users the kinds of explanations they are using. Hohman *et al.*, who developed GAMUT, mentioned that some users were initially confused when first started using GAMUT regarding local or global explanations. In our solution, we also applied feature selection techniques to show only the most important features as explanations, as other studies have shown that explanations based on a high number of causes are not effective for humans [Mil19].

TELEGAM [Hoh19] is a continuation of GAMUT's work but focuses on more verbalization explanations using natural language techniques. The Global Explanations are now primarily explained by verbal explanations, and PDP plots are shown as an additional resource. The Local Instance explanation is the same as GAMUT. The interactive table is not presented in TELEGAM.

ExplainExplore [Cv20] uses a different approach for local and global explanations. The main capabilities of ExplainExplore are based on surrogate models. Recall from Chapter 3 that surrogate models are simpler models, such as linear and decision trees used to explain more complex models approximating their results. Figure 5.10 shows the graphical interface developed by Collaris *et al.* The users can select a different set of surrogate models and change their hyperparameters Ⓐ as desired. With this first part of the screen, we can observe that the solution is focused on expert users, requiring an understanding of how the selected surrogate models work and their hyperparameters. One of the visualizations shown in the center Ⓑ is similar to the one we have in our solution for the local explanations, showing the degree of contribution

each feature had for the model prediction outcome. At the top of the same visual (B1.1 and B1.2), the solution shows metrics to measure how well the surrogate model's predictions are faithful to the reference model. As the surrogates are simpler than the reference models, not all the explanations can be trusted. For some instances, the complexity of the decision made by the reference model is not achievable by the surrogate one. For the global explanations, the solution presents in the right section of the central visualization Ⓑ2 a plot that shows how each feature contributed to a higher number of instances in the model. Each line in this visualization represents an instance. Finally, the visualization in the right of the screen Ⓒ allows the users to find nearby data points to the one under analysis, searching for similar instances that present better explanations. In this same view, there is a functionality that is not precisely the contrastive approach of our solution, but that gives the user the possibility to add some perturbation to the instances to see how it affects the explanation. The paper mentions that the users can select the instances, but there is no explanation of how they do it. Also, it seems there is no interactive way to change between instances easily.

Recall from Section 3 that we also mentioned Temenos as a related work XAI explanation interface. However, as this is a proprietary tool, we could not have details about all the functionalities to compare.

From the comparisons, we can see that our solution delivers a satisfying number of functionalities and focuses on explanations for non-expert users. Recall from Chapter 3 that we have adapted state-of-the-art techniques to facilitate comprehension by non-expert users, such as normalizing Shapley Values to a comprehensible range that

Figure 5.10: ExplainExplore [Cv20]

does not require an understanding of game theory concepts. We also adapted the visualizations, labelling and colours. For the contrasting explanations, we adapted the verbalization explanations produced for categorical features and formatted the numerical feature's values. Our interface also serves as an abstraction layer for the non-expert users, as they do not need to learn how to code to use the available explanation libraries.

5.8 Discussion

One main goal of this study was to easily explain machine learning predictive models without requiring the end-users to have a technical background to understand and produce the explanations. At the same time, we want to produce the explanations for a real-world dataset and collect results based on issues of real scenarios. Real-world problems' data are usually more complex and challenging than most of the toy datasets available online, on which many works are based. Also, complicated

problems are harder to be explained. As explained before, we did extensive work to shape the data to be ready for the model and, consequently, to be explained. When we compare our work with the ones presented in this section, ExplainExplore and XAI Diagnostic [WYAL19; Cv20] also generated explanations for real-world cases.

In terms of the explanations available and functionalities, we observed that only our solution and GAMUT [HHC+19] provides an extensive set of functionalities. However, GAMUT still requires technical skills to understand the explanations and set up the interface to produce the results. We observed that none of the studies mentioned that the number of causes (features) in the explanations should be limited to the minimum to facilitate the understanding to humans, as demonstrated in the work of Miller [Mil19]. The target audience for the solutions is also a surprise, as only our solution and XAI Diagnostic [WYAL19] have the ultimate end-users in mind when designing and evaluating the interfaces. Although data scientists and machine learning practitioners also need to understand the model's reasoning, they are not usually the ones who will decide based on the model outcomes. As stated by Zhou and Chen [ZC18], it is necessary to include end-users in the process of machine learning explanations for the elaboration of explanations that will serve them.

As part of our human subjects evaluation, one of the questions asked participants to choose between an explanation given by our modified version of the SHAP package visualization or the original one, as illustrated in Figures 4.8a and 4.8b. All participants chose our modified version. This confirms the findings of Kumar *et al.* [KVSF20] and Kaur *et al.* [KNJ+20] that stated that SHAP visualization is a challenging method to understand by non-expert users.

5.9 Summary

This chapter showed the evaluation methods applied to the proposed architecture, the machine learning model used as the case study and the web-interface. We started the chapter with the evaluation of a functional version of our proposed architecture. Then, we presented the evaluation for the different ratio tests applied to the dataset to overcome the "class imbalance" problem. Next, we explained how we reduced the number of features input to the model by 44%, using the *Recursive Feature Elimination* technique. To close the machine learning model evaluation, we showed the confusion matrix to detail the models' accuracy.

To evaluate the web-interface developed in our solution, we conducted two different evaluations with human subjects and objective approaches. In human subjects, we recruited volunteers to use our interface and answer a survey. A total of 9 participants evaluated our solution, and the majority approved it regarding the usefulness, easiness to understand and instances' target and search functionalities. For the objective evaluation, we compared our solution against similar ones proposed in the literature. The solutions were compared based on the different explanation techniques available, target instances capability, search functionalities, and audience. To close the chapter, a discussion about the evaluation and results is given.

Chapter 6

Conclusions and Future Work

In this chapter, we present the conclusion and remarks of this work and directions for future works.

6.1 Conclusions

The ultimate goal of this thesis was to investigate the gaps and deficiencies of the Explainable Artificial Intelligence (XAI) field regarding explanations for machine learning predictive models for non-expert users and propose a solution focused on this group of users. Our investigation revealed that most of the solutions available were designed to serve data scientists and machine learning practitioners. Furthermore, some explanations tools were pointed out by related works [KNJ+20; KVSF20] stated in Chapter 2 as being a challenge to understand even for machine learning experts.

To develop our solution, we brought together proposed explanation tools by the XAI community, the social sciences and human-computer interaction concepts of how

humans better understand explanations and analysis of similar solutions' strengths and weaknesses. To test our solution's effectiveness, we sought to apply it to a complex real-world problem. Thus, we modelled a customer churn machine learning predictive model for a financial institution to serve as our case study. The architecture, components and techniques we used in the development are described in Chapter 4.

We evaluated our solution with various experiments separated into three aspects. The first focused on the proposed architecture, a second on the machine learning model and a third on the web-interface.

The proposed architecture was evaluated by implementing it with well-known technologies for each of the proposed sub-components. The architecture shows to be feasible and functional with our implementation.

The first experiment for machine learning was designed to evaluate the best ratio and technique to solve the "class imbalance" problem. Among the techniques and ratios tested, we achieved better performance with the clustering technique and 0.1 ratio ($1,859$ churn cases and $18,590$ loyal cases) that achieved 81% sensitivity and 60% specificity, respectively. A second experiment was conducted to reduce the number of features necessary to the model. We used the *Recursive Feature Elimination* technique for different thresholds (number of features). As a result, we were able to keep the same performance, reducing the number of features by 44%. The last evaluation showed the confusion matrix to our model, depicting the number of true positives and negatives and false positives and negatives for our classes.

The web-interface was evaluated with subjective and objective evaluations. We first compared our solution to similar ones in terms of the functionalities available and

the intended audience. GAMUT [HHC+19] is the solution that offers the same number of functionalities as our solution but requires technical skills to understand and produce the explanations. XAI Diagnostic [WYAL19] was designed for non-expert users but offers fewer explanations and functionalities than our solution. The human-subjects evaluation revealed that our solution was approved regarding usefulness and easiness to understand for most participants.

The machine learning model presented an excellent performance in terms of sensitivity, which is more valuable to solve our case study. The web-interface was well evaluated in objective and subjective terms and contributed to filling gaps in the literature regarding explanations to non-expert users and how the explanations are presented in general.

In Section 1.1, we stated the questions we would like to answer with this research's development. Let us recap our answers to these two questions:

Q1: How to produce more consumable and understandable explanations to end-users?

A1: To produce more consumable and understandable explanations, we brought knowledge from the social sciences and HCI when deciding the kind of explanations we would include in our solution and on the designing of each of them (Sections 3.5 and 4.3). Our human subjects evaluation showed that most participants considered the explanations given in our solution easy to understand. When we integrated our modified versions of the explanations techniques in a web-interface, we also eliminated the barrier of not having technical skills to manipulate the libraries with a computer programming language.

Q2: Does the simplification of the existing explanation tools help end-users understand models better?

A2: We have seen in the human subject's evaluation of our web-interface (Section 5.6) that all participants had chosen our version of the visual and explanation for the local explanations when compared to the one available in the SHAP package. When comparing our web-interface with the XAI Diagnostic [WYAL19] that also focus on non-expert users, they mentioned in their results that users complained about the overwhelming number of rules presented for the contrastive explanations. We present only one explanation for each instance in our solution, and this screen was evaluated by 90% of the participants as easy to understand.

6.2 Future Work

Our study was limited to classification prediction on tabular data. As ongoing and future work, we plan to adapt the solution to generate explanations for regression models and explore different explanation techniques for image and text data. With the increasing use in recent years of deep learning models to solve image and text data problems, exploring explanation techniques for these use-cases is also necessary.

Due to confidentially access requirements to the data used in our solution's case study, participants' number and diversity that evaluated our solution was limited. We intend to apply our solution to different data-sets and evaluate it with many people in future work.

Bibliography

[AB18] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[AH20] J. Adams and H. Hagras. A type-2 fuzzy logic approach to explainable ai for regulatory compliance, fair customer outcomes and market stability in the global financial sector. In *Proceedings of the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 194–201, 2020.

[AVW+18] A. Abdul, J. Vermeulen, D. Wang, Lim B. Y., and M. Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the 2018 Conference on Human Factors in Computing Systems (CHI)*, page 582:1–582:18, New York, NY, USA, 2018. ACM.

[BB12] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)*, 13:281–305, 2012.

[Bre01] L. Breiman. Randon forests. *Machine Learning*, 45:5–32, 2001.

[CBHK02] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research (JAIR)*, 16:321–357, 2002.

[Cv20] D. Collaris and J. J. van Wijk. ExplainExplore: Visual exploration of machine learning explanations. In *Proceedings of the 2020 IEEE Pacific Visualization Symposium (PacificVis)*, pages 26–35. IEEE Computer Society, 2020.

[DBR16] S. Dauzon, A. Bendoraitis, and A. Ravindran. *Django: Web Development with Python*. Packt Publishing, Birmingham, UK, 2016.

[DCL+18] A. Dhurandhar, P. Y. Chen, R. Luss, C. C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the Missing: Towards contrastive explanations with pertinent negatives. In *Proceedings of the 2018 Conference on Neural Information Processing Systems (NeurIPS)*, pages 592–603, 2018.

[DK17] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017.

[Ere18] K. Eremenko. *Confident Data Skills: Master the Fundamentals of Working with Data and Supercharge Your Career*. Kogan Page Ltd., UK, 2018.

[FRD19] A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of

prediction models simultaneously. *Journal of Machine Learning Research (JMLR)*, 20(177):1–81, 2019.

[GKBP15] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics (JCGS)*, 24(1):44–65, 2015.

[HHC+19] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 Conference on Human Factors in Computing Systems (CHI)*, pages 579:1–579:13, 2019.

[HKP12] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Burlington, MA, 3rd edition, 2012.

[Hoh19] F. Hohman. TELEGAM: Combining visualization and verbalization for interpretable machine learning. In *Proceedings of the 2019 IEEE Visualization Conference (IEEE-VIS)*, pages 151–155. IEEE, 2019.

[HYGS08] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[Kni14] V. Knight. Cooperative games. In *Game Theory*. https://vknight.org/Year_3_game_theory_course/Content/, 2014.

[KNJ+20] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 Conference on Human Factors in Computing Systems (CHI)*, pages 92:1–92:14, 2020.

[KVSF20] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler. Problems with shapley-value-based explanations as feature importance measures. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 5491–5500, 2020.

[Lip18] Z. C. Lipton. The mythos of model interpretability. *ACM Queue*, 16(3):31–57, 2018.

[LL17] S. M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 2017 Neural Information Processing Systems (NIPS)*, pages 4766–4775, 2017.

[LNA17] G. Lemaitre, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research (JMLR)*, 18:1–5, 2017.

[Mil19] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[Mol19] C. Molnar. Interpretable machine learning, 2019. `https://christophm.github.io/interpretable-ml-book/`.

[MST20] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* 2020)*, pages 607–617, 2020.

[PBM09] R. C. Prati, G. E. Batista, and M. C. Monard. Data mining with imbalanced class distributions: Concepts and methods. In *Proceedings of the 4th Indian International Conference on Artificial Intelligence (IICAI)*, pages 359–376, 2009.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.

[RSG16] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[SGK17] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Porceedings of the 34th International Conference on Machine Learning (ICML)*, pages 4844–4866, 2017.

[Sha53] L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games*, volume 2, pages 307–317. Princeton University Press, 1953.

[vRv+18] J. van der Waa, M. Robeer, J. van Diggelen, M. Brinkhuis, and M. Neerincx. Contrastive explanations with local foil trees. In *Proceedings of the 2018 ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2018.

[WMR18] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology (Harv J Law Tech)*, 31(2):842–887, 2018.

[WPB+20] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 26(1):56–65, 2020.

[WYAL19] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 Conference on Human Factors in Computing Systems (CHI)*, page 601:1–601:15, New York, NY, USA, 2019. ACM.

[YHSA20] F. Yang, Z. Huang, J. Scholtz, and D. L. Arendt. How do visual explanations foster end users' appropriate trust in machine learning? In *Proceedings of the 25th International Conference on Intelligent User Interfaces (IUI)*, page 189–201, New York, NY, USA, 2020. ACM.

[YRR19] C. Yang, A. Rangarajan, and S. Ranka. Global model interpretation via recursive partitioning. In *Proceedings of the IEEE 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS 2018)*, pages 1563–1570. IEEE, 2019.

[ZC18] J. Zhou and F. Chen. 2D transparency space—bring domain users and machine learning experts together. In *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent (HCIS)*, pages 3–19. Springer, 2018.

[ZLR+18] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood. Explainable AI for designers a human-centered perspective on mixed-initiative co-creation green open access added to TU Delft Institutional Repository 'you share, we take care!' – Taverne project. In *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 458–465. IEEE, 2018.