



POLITECNICO

MILANO 1863

Master of Science in Computer Science and Engineering
School of Industrial and Information Engineering

Enhancing Fraud Detection Through Interpretable Machine Learning

Master's Thesis

Federico Piccinini, 919884

Supervisors:

prof. dr. E. Della Valle Politecnico di Milano
M. Belloni MSc. HousingAnywhere

Milan, September 2020

Sommario

Con il termine *Automated Fraud Detection (Identificazione automatica di Frode)* viene indicato l'insieme delle azioni automatizzate (i.e. svolte da macchine) eseguite allo scopo di identificare utilizzi illegittimi di prodotti e/o servizi. Esso rappresenta un ambito emergente ed in espansione dove un numero sempre crescente di casi beneficia dell'utilizzo di moderne tecniche di intelligenza artificiale. Allo stesso tempo, numerose applicazioni di *Fraud Detection* non possono essere completamente automatizzate per motivi quali l'impossibilità di essere totalmente sicuri dell'illegittimità di un utilizzo o la necessità che la decisione finale sia presa da un umano. In quest'ottica, diventa chiara l'importanza dell'interazione tra la parte automatizzata del processo di identificazione (e.g. un algoritmo di intelligenza artificiale) e gli umani che interagiscono con esso. In questa tesi viene proposto un processo aziendale di identificazione di frodi applicato al dominio degli online marketplace. Tale processo, grazie alla combinazione di moderne tecniche di apprendimento automatico associate a *Machine Learning Interpretability (interpretazione di tecniche di apprendimento automatico)*, permette di ottenere un'interazione tra umani e macchine che sia efficiente ed efficace. Come prima cosa, un algoritmo di classificazione, basato sullo stato dell'arte dell'intelligenza artificiale, è stato implementato per distinguere tra utilizzi legittimi e fraudolenti. In aggiunta, quattro differenti metodi di *Machine Learning Explainability* sono stati implementati e valutati su applicazioni reali. Tra questi è stato progettato e proposto un nuovo approccio all'interpretazione dei modelli di apprendimento automatico denominato EVADE. Tale metodo, grazie ad una procedura di ottimizzazione basata su Algoritmi Genetici, ha ottenuto risultati comparabili allo stato dell'arte.

Abstract

With the term Automated Fraud Detection is intended the set of automated (i.e. carried out by machines) activities performed to detect illegitimate usages of services and/or products. It represents a rising and expanding field where more and more use cases are benefiting from the usage of modern artificial intelligence techniques. At the same time, several Fraud Detection applications can not be fully automated due to different reasons such as the impossibility of being totally sure about the illegitimacy of a use or the necessity of having humans making the final decision. In this view, the importance of the interaction between the automated part of the detection process (e.g. the machine learning algorithm) and the humans that are interacting with the tools becomes clear. Therefore, in this thesis we propose and design a fraud detection business process, applied to the domain of online marketplaces, which combines modern machine learning algorithms with predictions' interpretation so that the interaction between humans and machine is designed to be as smooth and efficient as possible. At first, a state-of-the-art machine learning classifier has been implemented to solve the problem of discriminating between which usages are legit and which are not. On top of this, four different machine learning explainability methods have been implemented and evaluated on real tasks. Among these methods, a novel approach to interpretable machine learning has been designed and proposed. This method, named EVADE, through the usage of an optimization procedure based on Genetic Algorithms, generates machine learning explanations which proved to achieve state-of-the-art performances.

Ringraziamenti

Un grazie particolare va al Prof. Florian Daniel, il primo sostenitore del progetto e supervisore nelle fasi iniziali di questa tesi. Allo stesso modo, sono grato al Prof. Emanuele Della Valle per aver preso in carico la supervisione della tesi nelle fasi successive e al Prof. Mykola Pechenizkiy, di Eindhoven University of Technology, per avermi permesso di realizzare l'importanza di XAI (*Explainable Artificial Intelligence*) nelle interazioni tra umani e macchine.

Ringrazio HousingAnywhere per l'opportunità che mi è stata data e per tutti i fantastici colleghi che ho conosciuto durante questa esperienza. Nonostante la sfortunata circostanza nella quale il mondo si è ritrovato, è stato un piacere condividere questi mesi con voi. Tra questi, un ringraziamento speciale è per Massimo, che ha creduto nel progetto dal primo momento e lo ha supervisionato passo dopo passo. L'eccezionale qualità del tuo lavoro e la passione con la quale ti applichi sono ammirabili e mi hanno motivato a fare del mio meglio durante tutto lo svolgimento della tesi.

Alla mia famiglia, grazie per aver messo da parte le vostre preoccupazioni ed avermi permesso di perseguire le mie ambizioni prima in un'altra città e successivamente in un altro paese, facendo sì che potessi intraprendere alcune delle esperienze più importanti della mia vita.

A Sofia, so di non essere stato sempre la persona più facile da assecondare. Grazie per il supporto e i tuoi preziosi consigli.

Agli amici di Milano, non vi menzionerò uno per uno ma voglio che sappiate che negli ultimi cinque anni siete stati la mia seconda casa, *letteralmente*. Sono sicuro che, anche quando saremo tutti sparsi in giro per il mondo, voi continuerete a rappresentare lo stesso per me.

Questa tesi è la tappa finale di un percorso che ha richiesto parecchi sacrifici, tempo ed energia. Tuttavia, alla fine di questi due anni, riconosco che quello che quest'esperienza mi ha lasciato è molto di più di quanto si è presa.

Keep pushing,

Federico

Contents

List of Figures	xii
List of Tables	xiii
1 Introduction	2
1.1 Context	2
1.2 Contribution	3
1.3 About HousingAnywhere	4
1.4 Fraud Attempt Example	5
1.5 Outline	7
2 Background	8
2.1 Business Process	8
2.2 Fraud Detection	10
2.3 Machine Learning Paradigms	11
2.3.1 Supervised Learning	11
2.3.2 Unsupervised Learning	12
2.4 Dealing With Imbalanced Data	12
2.4.1 Sampling	12
2.4.2 Cost Function	13
2.5 Feature Engineering	14
2.6 Clustering	15
2.7 Machine Learning Models	17
2.7.1 Decision Tree	17
2.7.2 Boosted Decision Trees	18
2.7.3 Light Gradient Boosting Machine (LightGBM)	18
2.7.4 Genetic Algorithms	19
2.8 Evaluation	20
2.8.1 Metrics	20
2.8.2 T-test For Model Selection	22
2.9 Machine Learning Interpretability	23
2.9.1 Interpretable Models	24
2.9.2 Explanations Evaluation	24

2.9.3	Global Surrogate	27
2.9.4	Local Surrogate	27
2.9.5	Shapley Values	28
2.9.6	Shapley Additive Explanations (SHAP)	30
2.9.7	Counterfactual Explanations	31
3	Related Works	32
3.1	Financial Statements	32
3.2	Credit Card	33
3.3	Online Marketplaces	33
3.4	Human-grounded Explanations Evaluation	34
4	Problem Statement	35
4.1	Problem Formulation	35
4.2	Data	35
4.3	Former Model	36
4.4	Research Directions	37
5	Approach	39
5.1	Performance Experiments Approach	39
5.1.1	Experiments Setting	39
5.1.2	Experiments Evaluation	40
5.2	Interpretability Experiments Approach	41
5.2.1	Experiments Settings	41
5.2.2	Experiments Evaluation	42
6	Experiments	43
6.1	Performance Oriented Experiments	43
6.1.1	Infrastructure Simplification	44
6.1.2	Objective Function	44
6.1.3	Missing Values Imputation	45
6.1.4	Bayesian Optimization	45
6.1.5	Categorical Features Encoding	46
6.1.6	Numerical Features Enhancement	46
6.1.7	Boosting Technique	47
6.1.8	Minor Experiments	47
6.2	Interpretability Oriented Experiments	48
6.2.1	Model-based Explanations	48
6.2.2	Local Surrogate Explanations	49
6.2.3	SHAP	50
6.2.4	Evolutionary Adversarial Explanation (EVADE)	50

7	Results	56
7.1	Performance Oriented Results	56
7.1.1	Successful Experiments	56
7.1.2	Discarded Experiments	59
7.1.3	Final Model Performance	59
7.2	Interpretability Oriented Results	61
7.2.1	Explanations Accuracy	61
8	Conclusions	67
8.1	General Contribution	67
8.2	Academic Contributions	69
8.2.1	Classification Contribution	69
8.2.2	Interpretability Contributions	70
8.3	Business Contributions	71
8.3.1	Model Performance Contributions	71
8.3.2	Explanations Contributions	72
8.4	Limitations & Future Work	74
	Bibliography	75

List of Figures

1.1	HousingAnywhere Logo	4
1.2	Listing Main Web Page	5
1.3	Listing Details Web Page	6
1.4	User Profile Web Page	7
2.1	Fraud Detection process as is at HousingAnywhere	9
2.2	SMOTE Visual Sample Generation	13
2.3	COEC Training Process	16
2.4	Classification boundaries comparison without (left) and with (right) COEC.	16
2.5	Simple Decision Tree	17
2.6	ROC Curve	21
2.7	Precision-Recall Curve	22
4.1	Legacy Machine Learning Model	36
5.1	Train-Test-Validation splits	40
5.2	Machine Learning Experiments Pipeline	40
5.3	Explanations Evaluation Task	41
6.1	Fitness Function Computation	53
8.1	Final HousingAnywhere Fraud Detection Process	68

List of Tables

2.1	Custom cost function $C(< \textit{true class} >, < \textit{predicted class} >)$	14
7.1	Legacy Model Performance	56
7.2	Simplified Model Performance	57
7.3	Categorical Features Encoding Comparison	57
7.4	Model Performance with Weighted Loss	58
7.5	Model Performance with Missing Value Imputation	58
7.6	Model Performance after Bayesian Optimization	58
7.7	Discarded Experiments Performances	59
7.8	Experiments Improvements Breakdown. Between brackets, the relative percentage improvement with respect to the previous iteration is given.	60
7.9	Overall Performance Improvements	60
7.10	Evaluation Level Accuracy, each cell contains the corresponding accuracy	62
7.11	Baseline without Explanations, each cell contains the corresponding accuracy	63
7.12	Model-based Explanations Accuracy	64
7.13	Local Surrogate Explanations Accuracy	64
7.14	SHAP Explanations Accuracy	64
7.15	EVADE Accuracy	65
7.16	Methods Comparison Accuracy	65

List of Acronyms

AI	Artificial Intelligence
ML	Machine Learning
COEC	Cluster-oriented ensemble classifier
LightGBM	Light Gradient Boosting Machine
GOSS	Gradient-based One-Side Sampling
EFB	Exclusive Feature Binding
PR Curve	Precision-Recall Curve
ROC Curve	Receiving Operating Characteristic Curve
LIME	Local Interpretable Model-Agnostic Explanations
SHAP	Shapley Additive Explanations
SVM	Support Vector Machine
BBN	Bayesian Belief Network
ANN	Artificial Neural Network
KNN	K-Nearest Neighbors
NLP	Natural Language Processing
DART	Dropout meets Multiple Additive Regression Trees
GA	Genetic Algorithm
EVADE	Evolutionary Adversarial Explanations
OS	Operating Systems
BO	Bayesian Optimization

Chapter 1

Introduction

In this section, the general scope of the research will be introduced. At the beginning, the context is proposed with an introduction about Fraud Detection in general, followed by a brief summary of the contributions brought by this thesis. Moreover, the chapter continues with a description of the company in which this research has been performed, namely HousingAnywhere, together with a concrete example of fraud in the company's application domain. Finally, the section is concluded with an outline about how the contents in the rest of the document are structured.

1.1 Context

With the term Fraud it is indicated the illegitimate usage of a product or a service which is aimed at gaining benefits in a different way from the ones that the product or service is intended for. Please note that, for the sake of this research, the term fraud and scam will be used as synonyms. Therefore, Fraud Detection represents the business process aimed at identifying such illegitimate usages.

Detecting frauds is far from a trivial task and it introduces several challenges. First, the fraudster is advantaged: it is not known when a fraud is going to be perpetrated and in almost every real world applications it is impossible to be 100% sure of the illegitimacy of a usage. As a consequence, an effective fraud detection process must be time responsive but at the same time it has to involve humans. For these reasons, the field of Automated Fraud Detection is growing and the number of applications where machine learning and data mining techniques are involved is rising dramatically, as showed in [15].

With that in mind, this thesis proposes an automated process for fraud detection applied to the field of online housing marketplaces. In this application domain, a fraud attempt is represented by the behaviour of a user, the fraudster, who misleads other users by offering them a fake service, in this case an accommodation to be rented, aimed at stealing money from them. At the same time, a meaningful fraud detection process should aim at identifying such behaviours as soon as possible and consequently remove

those listings form the online marketplace, so that fraudsters have minimal opportunities to interact with legit users. Furthermore, since this is one of those domains for which it is impossible to be totally sure about the illegitimacy of an item, the fraud detection process must involve humans in finally determining whether a listing is a fraud attempt or not.

However, interactions between humans and machine learning models require humans to understand the machines in order to trust and interpret their predictions so that the optimal decision can be made. Moreover, the increasing performances and complexity of modern machine learning techniques represent a relevant obstacle for humans in understanding why models are making certain decisions, thus making the interaction between humans and machines less efficient and accurate.

1.2 Contribution

To match the presented criteria, this thesis proposes as a solution a process based on machine learning techniques with a dedicated focus on the interaction between models and humans. Specifically, the first building block of the solution is a machine learning algorithm which performs an initial classification of items between the ones that are considered legit and the ones that could represent a fraud attempt. Later, listings which are considered to be a fraud attempt are elaborated by an AI interpretability module which is aimed at providing explanations to the classification decision taken by the machine learning model. Finally, the item is ready for the final human check, which will assess, based on the listing's characteristics, its classification outcome and explanations, whether the listing is a real scam attempt or a false positive.

To sum up the scope of this research, at first a machine learning model to solve a binary classification task has been developed, followed by the implementation and testing of different state-of-the-art machine learning explanations techniques. Among these, a novel technique has been designed which exploits the optimization capabilities of Genetic Algorithm to generate adversarial machine learning explanations, achieving state-of-the-art performances. Moreover, it has been empirically proven that human validations of machine learning predictions can benefit from the implementation of explanations both in accuracy and efficiency.

1.3 About HousingAnywhere

HousingAnywhere is an online housing marketplace¹, originally founded in 2009 in Rotterdam (Netherlands), which aims at providing customers, especially students and young professionals, a safe online marketplace to find the perfect accommodation to rent anywhere in the world.



Figure 1.1: HousingAnywhere Logo

The platform, which is now present in more than 100 countries, offers a two-sided marketplace where tenants can find a property that suits their renting needs, while landlords can offer their properties to the growing users base that periodically access the platform.

¹<https://housinganywhere.com>

1.4 Fraud Attempt Example

To clarify what is intended for fraud or scam attempt in this application domain, a real example is hereafter reported. At first, by looking at figure 1.2, it is possible to see how the item appears to be a legit one.

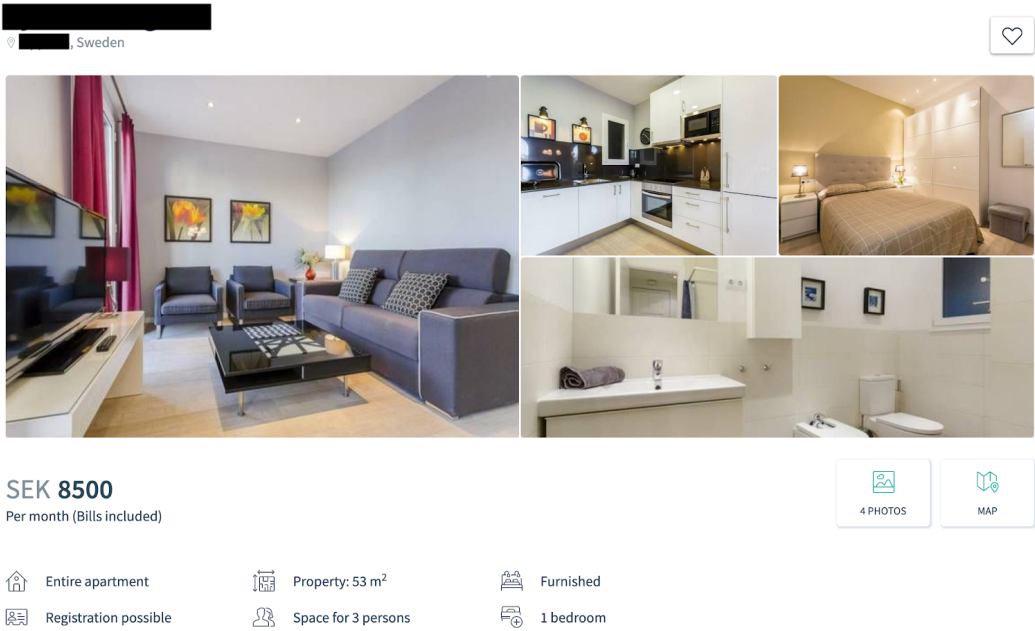



Figure 1.2: Listing Main Web Page


However, by taking a closer look to this listing’s details, showed in figure 1.3, it is possible to notice several anomalies. At first, the rent price, which is the equivalent of about 800 €, is significantly cheaper than the average rent price of the area for items of the same commodity level. The price convenience, together with nice pictures and a very detailed description, are usually used by fraudsters to attract as many users as possible and maximize the effectiveness of the fraud.


Moreover, useful piece of information is present also in the landlord’s profile, which can be seen in figure 1.4. Specifically, it is suspicious that the user registered right before posting the previous listing, which is also the only listing ever posted by him/her. In addition, both the profile picture and the user’s description are missing.


As it can be derived from this example, a lot of different factors contribute to the final decision of marking the listing as a fraud attempt or not, going from the user’s profile to the listing’s description, including also *hidden* information, such as IP addresses used to log in to the platform.


SEK 8500
Per month (Bills included)


 Entire apartment

 Property: 53 m²

 Furnished

 Registration possible

 Space for 3 persons

 1 bedroom

This refurbished loft apartment is light-filled and airy featuring pristine wooden floors, featured throughout the entire space, enhanced by its high ceilings. The large living space is made authentic through its eclectic and tasteful décor. Its interior combines contemporary design furniture with retro Design classics creating a genuine home built for a international palate. Enjoy the iPod Hi-Fi system and the flatscreen for great home entertainment

Bedroom
Number of bedrooms **1** Bedroom furnished Lock on bedroom

Areas
Kitchen **Private** Balcony/Terrace **Private** Wheelchair accessible
Bathroom **Private** Garden **Shared** Allergy friendly
Toilet **Private** Basement **Private**
Living room **Private** Parking **Shared**

Figure 1.3: Listing Details Web Page

Therefore, it becomes clear that an automated approach is needed, where a lot of different factors are considered and elaborated as fast as possible. However, it is fundamental, from a business perspective, that any kind of automated process is then flanked by a final human validation, thus having the role of a support tool for humans in making the final decision.

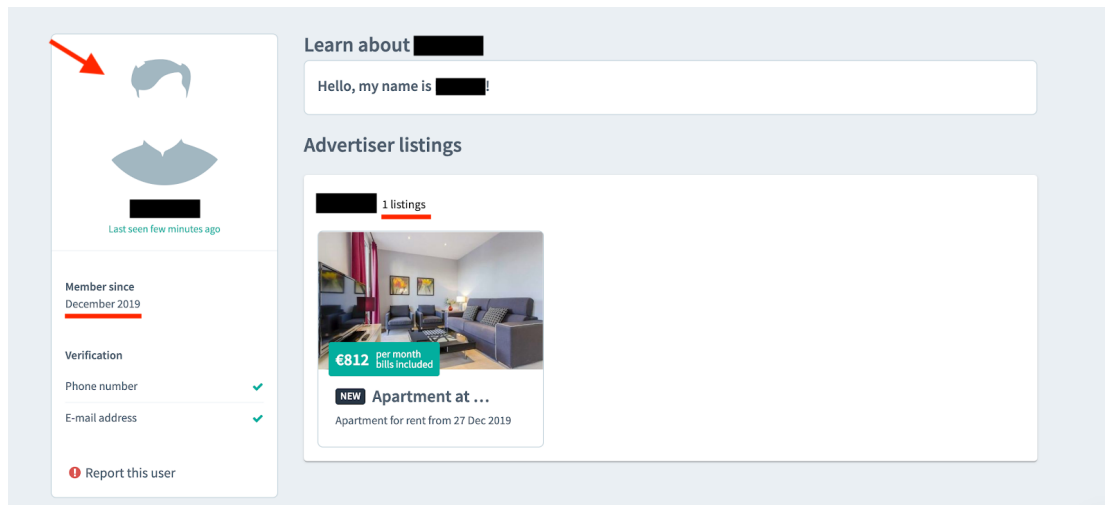


Figure 1.4: User Profile Web Page

1.5 Outline

The rest of the document is organized as follows: in chapter 2 the background knowledge on which this research is based is reported, followed by a selection of related works from the literature in chapter 3. Later, in section 4, the problem is presented and formalized and the subsequent research directions are presented; in chapter 5 the approaches on which the experiments, widely described in chapter 6, are based are stated. The results achieved by these experiments are then presented in chapter 7. Finally, the conclusions of the research are stated in chapter 8 with a specific focus on the contributions brought by this thesis to the business process and to the scientific community.

Chapter 2

Background

In this section, the background knowledge on which the thesis is based is presented. At first, a general introduction of the business process and the the domain of automated fraud detection will be given while in the latter part of the section the technical machine learning knowledge will be presented.

2.1 Business Process

In order to understand where technology can bring the most value in HousingAnywhere’s Fraud Detection process, a proper analysis of the system as it is will be performed in the rest of the section.

First, it is important to recall the meaning of *fraud* from HousingAnywhere’s perspective, which for the focus of this research can be considered as synonym of scam. With the term *fraud* it is identified the act of publishing an item (i.e. housing accommodation) on the platform with the final aim of stealing money from legit users through payments which are not matched with the service such users are thinking to purchase, as showed in section 1.4. In most cases, this is realized by scammers by engaging as many users as possible in conversations, by publishing extremely appealing accommodations, and then drive the communication outside the safe platform, where they can take full control of the situation and get rid of the safe payments process.

Therefore, it is in HousingAnywhere’s best interest to identify *scams* being published on the platform as soon as possible and responsively archived them, so that users are not involved in dangerous conversations with scammers. As previously mentioned, it is also impossible to be certain about the legitimacy of a user and, subsequently, it is required that humans take the final decision using artificial intelligence as a support tool.

The Fraud Detection system currently in place, which can be summarized in figure 2.1, is composed by the following steps:

1. As soon as items are published on the platform, they are sent to the AI algorithm to be analysed.
2. The algorithm estimate the probability of an item of being fraud attempt and if it is above certain levels, the accommodation is marked and sent to the Customer Solution department of HousingAnywhere along with its probability score.
3. Items that are sent to the Customer Solutions department are subjected to human validation, who will take the final decision about the accommodation being a *scam* or a false alert.
4. An additional automated step is in place for accommodations for which the confidence of them being a *scam attempt* is significantly high. Specifically, the candidate *scammer* is temporary denied to access the platform until a human check. In case the result of the human validation indicates that such user is a legit one, he/she will be allowed to access the platform again as soon as possible.

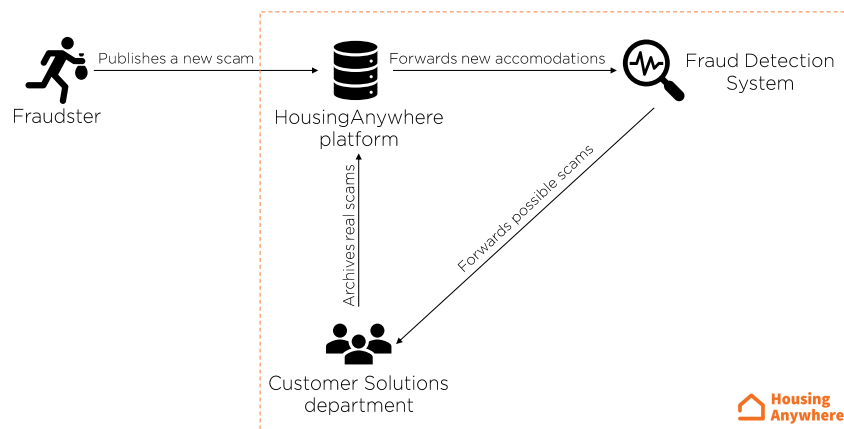


Figure 2.1: Fraud Detection process as is at HousingAnywhere

For how the system is organized, from a business perspective it is possible to define some key challenges:

- Human-machine interactions required predictions to be interpreted, and consequently trusted. Complex machine learning algorithms often produce predictions based on patterns which are not easily understandable by humans. Therefore it is fundamental to provide the Customer Solutions department with predictions which can be understood and trusted.

- It is also extremely important to reduce as much as possible the number of false alerts sent to such department. Having a big number of false alerts increases the workload which is required to be performed by humans and consequently increasing costs, and, at the same time, it can lead humans to make more wrong decisions.

2.2 Fraud Detection

This research is focused on the development of fraud detection systems aimed at the identification of specific unintended usages of services. Such systems, in most cases, are required to operate in time-sensitive environments, where being time responsive makes the difference between success and failure from a business perspective. Moreover, as stated in [28], it is impossible to be sure about the illegitimacy of a set of actions which are suspected to be fraudulent, thus suggesting the usage of structured and mathematical systems for analysing such suspected behaviours. Combining the previous requirements with the increasing amount of available information about user's actions drives intuitively the focus on the analysis of automated fraud detection systems, with special attention on Data Mining and Machine Learning techniques.

The number of applications of Artificial Intelligence for Fraud Detection is growing rapidly, as stated in [15], with a global market size of USD 20 billion in 2018 and it is expected to reach more than 100 billion by 2026. Among the several application domains of Fraud Detection, the most common, presented in [4], are *credit cards*, *money laundering*, *telecommunications*, *computer intrusions* and *medical and scientific frauds*. On the other hand, the huge amount of investments is not followed by a proportioned amount of publicly available researches on the topic. Even though in [4] several automated techniques applied in the past are presented, including rule-based algorithm, neural networks, tree-based methods, genetic algorithms, Bayesian networks, meta-learning algorithms and ensembling approaches, none of them were applied to the emerging field, from a Fraud Detection perspective, of Online Housing Marketplaces, which will be the setting of this research.

Another reason which limits the exchange of ideas about Fraud Detection research lies in the continuous evolution of fraudsters' approach and consequently adaptation of detection systems. Releasing very detailed information about the structure of a fraud prevention system could led fraudsters to discover and exploit possible flaws, as reported also in [18].

From a technical point of view, Machine Learning and Data Mining approaches to fraud detection have to face common challenges. First, the datasets about frauds tends to be heavily unbalanced towards legit usages. Particularly, as stated in [29], the ratio between legit users and fraudsters in most of cases varies from 100 to 1 to 100000 to 1, increasing significantly the difficulty of learning for Machine Learning algorithms due to not having enough samples, thus information, about the minority class. Furthermore,

it is non-trivial to evaluate the performance of such algorithms, since the unbalanceness of the data could bias some evaluation metrics towards the majority class. For example, a naive algorithm which always classifies instances as belonging to the majority class would achieve very good accuracy even though it does not bring any value to the fraud detection task since it would never identify any fraud.

Lastly, Online Marketplaces belong to the class of domains introduced before where it is not possible to be sure about the legitimacy of cases. This apparently harmless issues has negative impact on the performance of Machine Learning algorithms. Particularly, using as knowledge manually labelled data, which could contains errors, requires the algorithm to be robust towards noise in the input data.

In the rest of the chapter, the presented issues will be analysed along with relevant related researches from the literature.

2.3 Machine Learning Paradigms

In this section, two machine learning paradigms, which will be treated in the scope of this research will be presented, namely Supervised and Unsupervised Learning.

2.3.1 Supervised Learning

From a ML perspective, this task can fall under the umbrella of supervised learning problems. Supervised Learning indicates a category of machine learning tasks where given a set of couples (x, Y) , the goal is to learn a function f , such as $f(x) = Y$, which matches an input x_i with its corresponding output Y_i . The learning phase is considered supervised because it is based on a set of given samples x_i for which the associated output Y_i is known.

Binary Classification

One of the most common tasks which belongs to the class of supervised learning is classification. It is defined as a classification task, a problem for which the target output Y is represented by a label, associating items with their membership class.

Moreover, this research is based on a general binary classification problem, where the number of different classes is two and an item belongs only to one of them, specifically fraud attempts or legit items, as formalized in the following formula:

$$f(x) = \begin{cases} 1 & \text{if item belongs to the positive class} \\ 0 & \text{if item belongs to the negative class} \end{cases} \quad (2.1)$$

2.3.2 Unsupervised Learning

With the term unsupervised learning, it is intended a machine learning paradigm aimed at extracting information from unlabeled data. Specifically, the techniques which belong to this class are often based on the exploitation of pattern naturally present in the data.

Clustering

Among the approaches belonging to the class of unsupervised learning algorithms, clustering is probably the most widely used. A clustering task has to be intended as the process through which a set of objects, in this case data instances, are divided into different groups based on certain similarity criteria.

2.4 Dealing With Imbalanced Data

As explained exhaustively in [13], for imbalanced data it is intended every setting in which the distribution of instances belonging to different classes is significantly different. Even though this definition includes also multi-class scenarios, the focus of this research will be on binary classification tasks.

As introduced at the beginning of this chapter, Automatic Fraud Detection tasks are characterized by heavily imbalanced datasets, where the number of legit users significantly outperforms the one of fraudsters. As a consequence, many attempts can be found in the literature addressing this issue in similar settings. Among these, we will analyse sampling techniques and custom cost functions.

2.4.1 Sampling

The focus of this section will be on sampling techniques aimed at mitigating imbalanced datasets. Specifically, it is possible to define two main strategies: undersampling and oversampling.

For Undersampling is intended the process of mitigating the unbalancess by removing samples from the majority class. As carefully explained in [13] and [16], this could be done either *randomly*, by selecting random samples from the original dataset, or *informed*, where with more complex algorithm the aim is to overcome information loss generated by *random* undersampling. As claimed in [3, 7], such approach has proved to improve classification performances when applied to Random Forest models and Support Vector Machines on imbalanced datasets.

On the other hand, oversampling is the process of data augmentation of the minority class, aimed at mitigating the class imbalance problem and facilitating learning. As for undersampling, the easiest approach to oversampling is to *randomly* replicate instances of the minority class. However, as stated in [13], *random oversampling* can lead to

overfitting on the replicated instances (i.e. bad performances on new and never seen samples). To avoid this behaviour, *synthetic oversampling* has been introduced. Specifically, it is worth mentioning *SMOTE*, a synthetic oversampling strategy, proposed in [5], which proved to be successful in several application domains. The underlying idea of SMOTE, which can be seen in algorithm 1, is to generate a new sample obtained as the perturbation of existing ones.

Algorithm 1 SMOTE Underlying Concept

- 1: let x be a vector representing the sample you want to base your replication on
 - 2: let y be x 's nearest neighbor
 - 3: compute $d \leftarrow y - x$
 - 4: let r be a random number $\in (0, 1)$
 - 5: compute the new sample $z \leftarrow x + (d \cdot r)$
-

A visual representation of the generation of a new sample in a simple 2-feature scenario, can be found in figure 2.2.

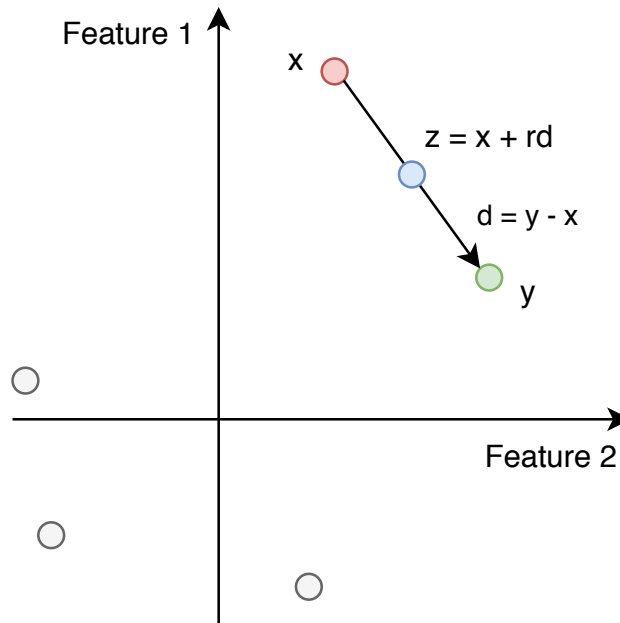


Figure 2.2: SMOTE Visual Sample Generation

2.4.2 Cost Function

While sampling techniques act before the learning phase, cost function approaches influence the actual learning process. Even though originally this technique was not developed to address imbalanced learning task, it has intuitively been applied and has

empirically proven to achieve better performances than traditional sampling techniques on certain imbalanced datasets [40, 24, 20].

As explained in [13, 16], the idea beyond applying custom cost functions to imbalanced datasets, whose structure can be seen in table 2.1, is to weight more classification errors on the minority class in a way that it compensates class imbalance.

	Predicted Class	
	0	1
TrueClass	0 C(0, 0)	C(0, 1)
	1 C(1, 0)	C(1, 1)

Table 2.1: Custom cost function $C(< true\ class >, < predicted\ class >)$

2.5 Feature Engineering

Real life datasets contain information represented in complex ways, using large number of attributes for the description of a single record. As a consequence, a proper feature engineering process showed its usefulness, throughout the literature, in improving fraud detection systems' performance.

One example of such process is presented in [3], in which modelling artificial features empirically proved to be fundamental for improving the performances of the applied Machine Learning models, specifically Support Vector Machines [34], Random Forest [6] and Logistic Regression [14], for detecting Credit Card frauds. Overall, most of artificial features model the comprehensive behaviour of the user by enriching every transaction instance with features such as number of different currencies used, number of transactions in the same day, average amount spent in the last months and many others.

Furthermore, in [19] an approach is proposed, related to Online Banking Fraud Detection, which aims at modelling the fraudster behaviour among different accounts. The underlying and intuitive idea is based on the hypothesis that most fraudsters perform several frauds at the same time, accessing the platform with different identities. Even though in [19] this is modelled using users' devices identifier, it is up to the specific application domain to define how to represent this feature, since they may have different logic and functionalities.

In [18], the authors suggested to model features to incorporate spatial information. Specifically, the rationale beyond it is that anomalies in spatial information can be an indicator that a fraud is being perpetrated. In the approach proposed in [18], an example of anomaly in spatial information is represented by a relevant discrepancy between shipping and billing addresses of the same user.

Finally, in [10] an attempt in modelling fraudsters profile is described. In the specific application domain, which is Telecommunication Fraud Detection, the authors claimed to have achieved better performances by modelling users' profile, that is representing explicitly features such as if the user was active outside working hours. The hypothesis beyond this specific feature, which seems to be confirmed by the claimed results, is that most frauds are perpetrated outside working hours.

As it can be derived from the previous examples, a domain-specific modelling of features, mainly aimed at representing user's behaviour, plays a key role in enhancing automated fraud detection. Therefore, similar approaches, tailored to HousingAnywhere application domain, must be considered.

2.6 Clustering

In this section we will analyse the combination of unsupervised and supervised learning, namely clustering and classification. There are several attempts in the literature which address this issue, aimed at using clustering methods to enhance final classification performances.

Among these approaches, in [12] the author provides a technique, whose logic can be seen in algorithm 2, based on the underlying idea that clustering can be used to split classes and simplify the learning task for the classifier. To do so, each class is internally clustered and the classifier is trained to predict the specific cluster rather than the original class. The mathematical intuition beyond it is that the resulting classification task is based on a higher number of decision boundaries, even if those are linear ones, approximating non-linear decision surfaces. This intuition is also verified by the achieved results which showed an improvement in performances for linear classifiers, while no improvement is reported for non-linear classifiers.

Algorithm 2 Clustering Inside Classes (CIC) Algorithm

- 1: **Training with CIC**
 - 2: Require: A set W with $K \geq 2$ classes, an integer $k \geq 1$
 - 3: for $j = 1, \dots, K$ do
 - 4: Partition class L_j into k clusters.
 - 5: end for
 - 6: Train classifier R using all training data to recognize all $k \cdot K$ clusters.
 - 7: **Classification with CIC**
 - 8: Require: A point x .
 - 9: Let $i = R(x), i = 1, \dots, k, \dots, k \cdot K$
 - 10: Return class of cluster i .
-

Another strategy has been proposed in [35] which combines clustering with an en-

semble of classifiers. Specifically, the algorithm, which training process be seen is figure 2.3, is based on the idea of a two-layer classifier. Firstly data has to be clustered. After that, at the first layer, several classifiers are trained to predict the probability of each sample to belong to every cluster. Then, the last layer is composed by a classifier which is trained to predict, given the cluster confidence matrices handed by the first layer classifiers, the target class.

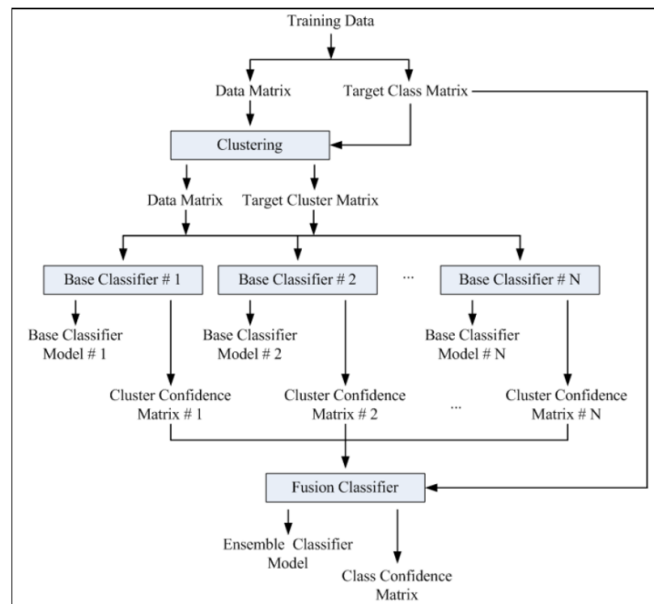
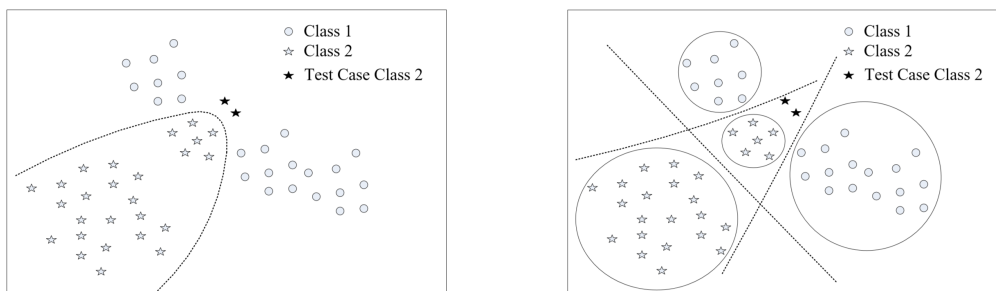


Figure 2.3: COEC Training Process

The intuition beyond it, similarly to the previous example, is that learning clusters' boundaries is an easier task and therefore the final classifier can achieve greater performances. A graphical representation of this concept can be seen in figure 2.4. The authors claimed that this approach, namely Cluster-Oriented Ensemble Classifier (COEC), outperforms traditional bagging and boosting techniques by 4-5%.



(a) Classification boundaries without clusters (b) Classification boundaries with clusters

Figure 2.4: Classification boundaries comparison without (left) and with (right) COEC.

2.7 Machine Learning Models

In this section, the background knowledge about the machine learning techniques used throughout this research is reported. Specifically, an introduction to the LightGBM algorithm is given, alongside its elementary building blocks. Moreover, in the final part of the section, an introduction to Genetic Algorithm will be proposed.

2.7.1 Decision Tree

Decision Trees are simple tree-based tools which can be used as a support in the decision making process. Furthermore, they proved to be an effective tool for machine learning classification tasks, both as plain decision trees and as the basis of more complex algorithm, such as Random Forest or LightGBM.

Decision trees are composed by an intuitive structure which can be intuitively understood by humans and it is composed by the following characteristics:

- Each node represents a split point where items belong to only one of node's children, based on a set of conditions.
- Each leaf of the tree represents the decision outcome for the items that follow the path to that leaf.

A graphical representation of a simple decision tree, with just three nodes and four leaves is showed in figure 2.5.

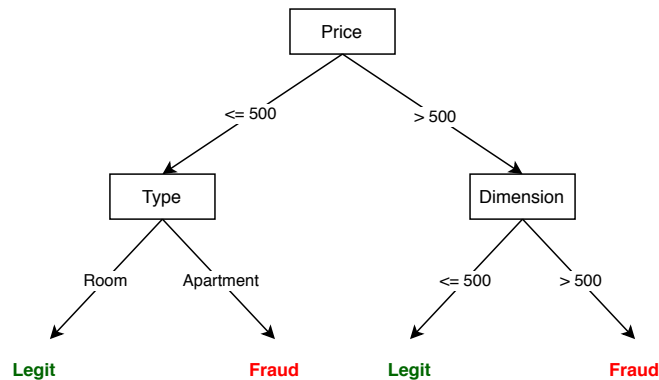


Figure 2.5: Simple Decision Tree

As an example, based on the tree in figure 2.5, the following decisions would be taken:

- Item with $Price = 350$ and $Type = Apartment$ would be classified as *Fraud*.
- Item with $Price = 1000$ and $Dimension = 250$ would be classified as *Legit*.
- Item with $Price = 1000$ and $Dimension = 1000$ would be classified as *Fraud*.

2.7.2 Boosted Decision Trees

One of the most effective techniques of combining decision trees in machine learning application is called Boosting. This approach is based on the simple idea of training several weak learners and combine them afterwards instead of training a single complex model. The singularity of the training process is represented by the fact that learners are trained successively and, at each iteration, the aim is to *learn* where previous learners failed. Even though weak learners can be represented by several different algorithms, for the scope of this research we will focus on the usage of decision trees as weak learners.

A general pseudo-implementation of this technique is presented in algorithm 3. However, it is important to state that the algorithm is intended to be a general introduction to boosting, while in the literature several different approaches have been presented which details are outside the scope of this research.

Algorithm 3 Boosted Trees

- 1: **Training of Boosted Trees**
 - 2: For each iteration i do:
 - 3: Train a simple decision tree M_i
 - 4: Calculate the error of M_i
 - 5: Increase the importance of areas where the classifier is not working correctly and decrease it where the classifier is accurate enough.
 - 6: **Classification using Boosted Trees**
 - 7: For each trained weak learned M_i :
 - 8: Accumulate the prediction value
 - 9: Return the final classification decision based on the combination of all weak learners.
-

2.7.3 Light Gradient Boosting Machine (LightGBM)

LightGBM is a recently developed approach to boosted trees, developed by Microsoft and presented in [17], which is claimed to achieve state-of-the-art performances in several machine learning applications with the advantage of being highly optimized from a computational perspective. From a general point of view, the boosting is performed by including in the loss function a gradient component to be optimized throughout the iterative training process.

The singularity of this approach relies in two techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB).

GOSS is aimed at reducing the number of samples by keeping only the ones with the highest gradient values, therefore assuming gradients as proxy for the amount of information carried by a specific sample. The selected samples are then used to assess

the information gain and build decision trees and their split conditions.

On the other hand, EFB is used to bundle together features that rarely appear with non-zero values in instances (e.g. a set of one-hot encoded features) therefore reducing the horizontal dimension of the input data and consequently reducing the learning complexity.

2.7.4 Genetic Algorithms

Genetic Algorithms (GA) [23] are an optimization technique based on simulating Darwin's theory of evolution on a specific domain, where instances are subjected to an iterative process of selection and evolution.

GA are composed by several key elements:

- **Population:** a set of instances which goes through the iterative process.
- **Fitness:** a function $f : x \rightarrow \mathbb{R}$ which maps an instance X , belonging to the population, to a value which represents a measure of how much that sample adapts to the environment, and consequently, how much is likely to survive to the next iteration.
- **Elitism:** a phenomena for which a set of the most fit instances is brought forward to the next iteration.
- **Crossover:** the process of creating a new instance for the next iteration by combining two parents instances belonging to the current one, similarly to what happens with genes in evolution theory.
- **Mutation:** a phenomena that randomly changes instances by modifying their features (genes).

In practice, the algorithm follows this structure:

1. It initially generates a random population of N instances, which composes the starting point of the algorithm.
2. Each sample belonging to the current population is evaluated through the fitness function f .
3. Through *elitism*, a defined percentage of the best candidates (i.e. the one with the highest fitness value) is brought over to the next iteration.
4. Several rounds of crossover are performed, until the target size of the new population is reached. The parents selection is performed through Tournament Selection [9], where a subset of candidates is sample with replacement from the current population. After sampling, the two best candidates in the subset are selected as

parents and a new offspring is generated and added to next iteration's population. The motivation behind the choice of Tournament Selection lays in the trade-off between exploration of candidates with different characteristics and computational efficiency.

5. The final step in generating the new population is mutation. A random perturbation of random candidates is performed by changing the values of their features.
6. The steps 2 – 5 are repeated for any generation.

2.8 Evaluation

In this section the background knowledge about the evaluation criteria that will be used in evaluating experiments throughout the rest of the research is given. At first, an introduction about classification metrics will be presented alongside each one's strengths and weaknesses. Meanwhile, in the last part of the section a commonly used statistical test in the field of machine learning, named t-test, will be introduced.

2.8.1 Metrics

In many real applications it is convenient to evaluate performances by using an individual metric (i.e. a number) to ease the understating and the comparison of different models. Before introducing the metrics it is necessary to introduce some measures:

- P : number of instances belonging to the positive class.
- N : number of instances belonging to the negative class.
- TP : true positive, number of instances correctly classified as belonging to the positive class by the model.
- TN : true negative, number of instances correctly classified as belonging to the negative class by the model.
- FP : false positive, number of instances wrongly classified as belonging to the positive class by the model while belonging to the negative class.
- FN : false negative, number of instances wrongly classified as belonging to the negative class by the model while belonging to the positive class.
- TPR : true positive rate, defined as $\frac{TP}{P}$.
- FPR : true positive rate, defined as $\frac{FP}{N}$.

At first, let's look at **Accuracy** (formula 2.2), defined as the ratio between the number of samples correctly classified over the total number of instances. Even though accuracy is an intuitive measure of performance, it has been shown in [13] that it can be an inappropriate measure for classification tasks with imbalanced classes. This can be easily verified by assuming to have a trivial model which always classifies samples as belonging to the majority class. If evaluated in a scenario in which the imbalance ratio

is 1000:1, it will achieve an almost perfect accuracy, while it is clear that it does not help in solving the classification problem since it will never identify any fraud.

Other very common metrics are **Precision** and **Recall** (formula 2.2 and 2.3). The first one is a measure of how good is the model when it predicts that an instance belong to the positive class. The latter, on the other hand, is a measure of how complete is the model in recognizing all cases belonging to the positive class. Even though these metrics may be valid individually for some specific tasks, in most cases, including Fraud Detection, it is more important to consider both metrics combined at the same time. To address this issue, **F1-score** has been introduced, which considers how good the classification is in general, including both precision and recall.

$$\text{Accuracy} = \frac{TP + TN}{P + N}; \text{Precision} = \frac{TP}{TP + FP} \quad (2.2)$$

$$\text{Recall} = \frac{TP}{TP + FN}; \text{F1-score} = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \quad (2.3)$$

In order to perform detailed comparison among different models, it can be helpful to consider graphical approaches such as **Receiver Operating Characteristic (ROC)** and **Precision-Recall (PR)** Curves.

The ROC Curve, which can be seen in figure 2.6, represents the plot of TPR over FPR, allowing for a comprehensive analysis over the whole spectrum of how models enforce the trade-off between TP and FP.

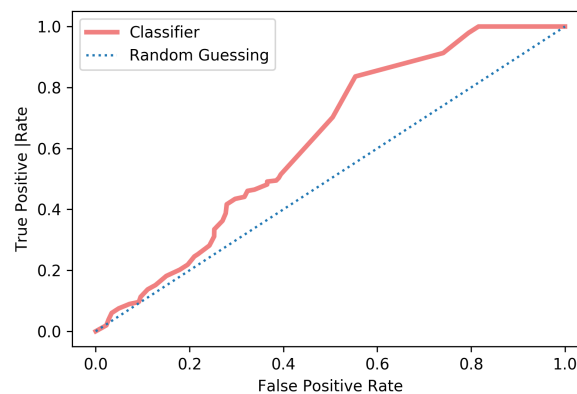


Figure 2.6: ROC Curve

As stated in [13], for highly skewed distribution the ROC curve could be overoptimistic over the real performances of the classifier. To address this issue, the Precision-Recall Curve, plotted in figure 2.7, can be useful by providing an unbiased representation of model's performances.

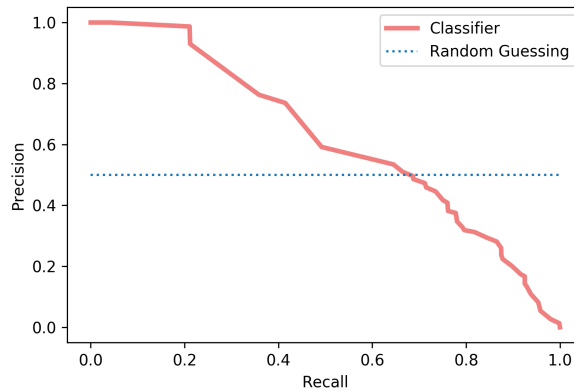


Figure 2.7: Precision-Recall Curve

Finally it is possible to state that for the focus of this research, namely a classification task with imbalanced classes, the most appropriate metric for the evaluation of a single model is **F1-score**, while for comparing different algorithms the **Precision-Recall Curve** can come in handy.

2.8.2 T-test For Model Selection

T-test is a statistical test aimed at assessing if two statistical populations have significantly different characteristics. In the field of machine learning, and specifically the model selection phase, this test is used to assess if one model is performing significantly better than another one. It specifically requires the two models to be evaluated over a set of independent and identically distributed (i.i.d.) tasks.

Given μ_1 and μ_2 respectively the mean of the two different populations, the t-test can be performed either paired or unpaired. In the paired t-test it does not matter if the discrepancy in performance is either positive or negative, thus resulting in the following hypotheses:

$$\begin{cases} H0 : \mu_1 = \mu_2 \\ H1 : \mu_1 \neq \mu_2 \end{cases}$$

On the other hand, the unpaired t-test is aimed at testing the discrepancy in just

one direction, therefore the new hypotheses can be formalized as follows:

$$\begin{cases} H0 : \mu_1 = \mu_2 \\ H1 : \mu_1 > \mu_2 \text{ or } \mu_1 < \mu_2 \end{cases}$$

For what concerns the scope of this research, the interest will be on paired t-test with the specificity of having the same number of experiments for both populations. Therefore, the procedure to be followed to perform such test can be decomposed in the following steps:

1. Select a significance probability α and calculate one set of k evaluations for each population, namely $h(A)$ and $h(B)$.
2. Calculate the set of k errors δ , where each value is calculated as $\delta_i = h(A)_i - h(B)_i$.
3. Calculate the mean error $\bar{\delta} = \frac{1}{k} \cdot \sum \delta_i$
4. Calculate the t-value as

$$t\text{-value} = \frac{\sqrt{k} \cdot \bar{\delta}}{\sqrt{\frac{\sum_i (\delta_i - \bar{\delta})^2}{k-1}}} \quad (2.4)$$

5. Assess the p-value, through distribution table given the t-value and the degrees of freedom df , which in this specific test settings is equal to $df = k - 1$.
6. Finally, similarly to the traditional procedure for statistical test over the null hypothesis, the following scenarios open up:

$$\begin{cases} p > \alpha \rightarrow \text{accept the null hypothesis } H0, \text{ therefore models have comparable performances} \\ p \leq \alpha \rightarrow \text{refuse the null hypothesis, models have significantly different performances} \end{cases}$$

2.9 Machine Learning Interpretability

Complex Machine Learning models can produce very accurate results which are not, by definition, easily to justify and explain. However, as humans, we tend to trust what we can understand while discarding what we find unclear. For this reason, in many Machine Learning applications, a trade-off between performances and interpretability is considered when choosing on which model to rely on.

Especially in application domains like Fraud Detection, where the model discriminates between fraudsters and legit users, it is fundamental to understand model choices, so that human can take a more informative decision based on model's prediction. At the same time, explanations can be used as a debugging method which allows to have a *window* for looking inside the model and, for example, spotting artificial bias inside it.

That said, the focus of this research will be on model-agnostic explanation at prediction level, where for model-agnostic explanations are intended all techniques which allow to treat the underlying model as a black box, so that such approaches can be easily scaled both on new problems or on the same problem with different settings.

For the rest of this research, we will define a prediction as interpreted if the related explanations (i.e. reasons for which a certain value has been predicted) have been generated.

2.9.1 Interpretable Models

Before moving forward, it is important to define the the concept of *interpretable machine learning model* and which items belong to this class. A machine learning model can be defined as interpretable if it generates predictions which correlation with the input features can be easily assess. Therefore a generic model can be defined as interpretable if it exists a function which is able to correlate each feature with its contribution to a specific prediction.

The following algorithms, as listed in [25], belong to the class of interpretable machine learning models:

- Decision Trees, because the prediction is the results of a specific path in the trees and it can be easily coupled with features' value through node splits.
- Linear Regression, thanks to the fact that the model is based on a linear combination of input features, thus it is itself the coupling function.
- Logistic Regression, for the same reason of Linear Regression.
- K-nearest Neighbours, because the prediction can be interpreted by looking at the neighbour instances and their features.

2.9.2 Explanations Evaluation

Even though the evaluation of explanations in interpretable machine learning is still a open research topic, a categorization of such task has been proposed in [8]. Specifically, three different categories have been defined as follow:

- *Function level evaluation*: this approach does not require direct human interaction but a proxy task, which has been previously evaluated by humans, must be present. The advantage here lays in using a proxy task to exploit and project already present human knowledge on the explainability task to be evaluated.
- *Application level evaluation*: here the idea is to let final users evaluate the explanations. This approach is the most complete one but it requires an experimental setup with humans, evaluation criteria and baselines available.

- *Human level evaluation*: this technique is a simplification of *application level evaluation* in which users are not required to be domain expert, therefore making the evaluation procedure easier to be performed.

Explanation Technique Properties

Given the increasing number of proposed techniques towards interpretable machine learning, a proper way of comparison is needed. Therefore, [25, 32] introduced a set of properties which can be used also as comparison criteria. On top of these properties, it is possible to elaborate the following:

- *Expressive power*: measure of how the explanation technique generates its explanations. For example, explanations can be generated as the result of an interpretable machine learning model (e.g. Decision Tree) or weighted sum of features.
- *Translucency*: how much the explanation method has to rely on knowledge coming from the underlying machine learning model. Since the scope of this research will be on model-agnostic approaches, all the techniques which will be analysed are required to have zero translucency (i.e. they do not require internal knowledge of the model to be explained).
- *Portability*: intuitive measure of how adaptable is the explanation technique to be used for explaining different models. This measure can also be seen as the inverse of translucency, which means that the scope of this research will be on techniques with high portability, according to the definition of model-agnostic models.
- *Complexity*: measure of how complex is the explanation approach, including computational and logical complexity.

Explanation Properties

Assessing the quality of a machine learning explanation is a non-trivial task, due to its nature, which requires to estimate the quality of information and how it has been provided from a human perspective. Even though it is still an open challenge to formally define how to calculate such values, a set of explanation's properties have been defined in [25, 32] from which the following list can be extracted:

- *Fidelity*: measure of how well the explanations approximate the behaviour of the underlying model. In case of surrogate models, fidelity can be seen as the inverse of the error between predictions of the original model and the surrogate one. Moreover, for certain use cases, such as local surrogate, it makes sense to measure the so called *local fidelity*, namely fidelity calculated by focusing only on a specific subset of the input space.
- *Consistency*: this measure defines how much the explanations vary by changing the model to be explained and keeping similar predictions. It is important to

mention here that consistency is not necessarily a desired property; for example, in a scenario in which explanations are used to debug several black box models built on different features, it may be desirable to have different explanations, otherwise the whole debug process could become useless.

- *Stability*: similar concept to consistency with the main difference of measuring variance at explanation level. Specifically, it measures how much the explanations vary when performed on similar instances. As for classical machine learning models, low stability (i.e. high variance) can lead to unreliable results and reproducibility difficulties.
- *Degree of importance*: easily measurable property which represents if explanations are based on features with different importance. This property is in most cases desirable since it allows for more informative explanation if needed, otherwise features importance can just be neglected.
- *Representativeness*: measure of how many instances can be explained by the same explanations.
- *Comprehensibility*: this property is the most important and, at the same time, the hardest to estimate. As it can be imagined, it's a measure of how good humans can understand the provided explanations. Furthermore, comprehensibility is influenced by several factors, including some which do not depend on the explanation model directly (for example, humans who interpret the explanations), making very difficult to find a general and accurate formal definition of it. Nonetheless, [25] proposed to estimate comprehensibility by looking at how easy it is to guess the output of the predictive model by using only the provided explanations.

This approach to model's interpretability was first introduced for Random Forest in [6], and its underlying idea is based on estimating how much each feature is important in producing the final prediction. Here, the concept of feature's importance is measured as the discrepancy between the prediction error with the original data and the one generated with a copy of such data where the values of the specific feature have been shuffled, aimed at removing information from that feature. Feature's importance is then intuitively considered directly proportional to the increase in prediction error.

The main advantage of this approach relies in the simplicity and intuitiveness of the importance measure while it can suffer from different limitations. First, the shuffling of the analysed feature inserts randomness and can lead to measures which vary a lot among different runs affecting reproducibility. Furthermore, another side effect of shuffling a feature is the possible generation of unrealistic samples. Finally, this framework does not allow to analyse correlated features, since a set of features may have effect only if all of them have specific values simultaneously.

2.9.3 Global Surrogate

An approach to interpret black box machine learning models, is to use global surrogate model. In this context, for surrogate model, it is meant a machine learning model which belongs to the class of *interpretable* models and approximate as precise as possible the behaviour of the black box one. In practice, this is realized by the following steps:

1. Train the black box model M , of which predictions need to be explained.
2. Select a dataset D , retrieve its predictions P by feeding M with D .
3. Train a new model I , belonging to the class of interpretable models, on the same dataset D but using as target P .

In order to assess the *fidelity* of the surrogate model, namely how good is approximating the original black box one, an approach using R-squared has been proposed in [25]. R-squared is usually used in machine learning tasks to evaluate how much information (i.e. variance) is explained by a specific model with respect to a reference one, which in most cases is a constant model. However, for the sake of this application, the reference model is considered to be the black box one, obtaining the following formulation:

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^n (\hat{y}_*^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^n (\hat{y}^{(i)} - \hat{y})^2} \quad (2.5)$$

where SSE and SST represents respectively the sum of squared errors and the sum of squared total. Moreover, errors are calculated as the difference between the i -th prediction of the surrogate model $\hat{y}_*^{(i)}$ and the corresponding one of the original model $\hat{y}^{(i)}$. On the other hand SST is calculated with respect to the average prediction of the black box model \hat{y} .

The surrogate interpretable model can then be used to provide explanations, which precision is strictly dependant on how good the surrogate model is approximating the behaviour of the main one. However, it is worth mentioning that is common for global surrogate to not being able to precisely imitate the behaviour of the main model. On the other hand, if the global surrogate model is able to achieve similar performance, and interpretability is needed, it is worth reconsidering the choice of a black box model in the first place.

2.9.4 Local Surrogate

As introduced in the previous section, it often happens that global surrogate model are a not sufficiently good approximation of black box models. Subsequently, a new approach has been introduced in [31] called *Local Interpretable Model-agnostic Explanations (LIME)*, based on the intuitively assumption that is easier to build more precise surrogate models by building several of them, each one specialized on a specific subset

of the features space (i.e. in the surrounding of the sample which prediction has to be explained).

This technique is applied by following these steps:

1. Given a input sample x , a set of predictions is generated using the black box model f which predictions have to be explained. Specifically, the predictions are generated for x and on other synthetic input samples obtained by small perturbation of x . For simplicity we will refer to this input subset as X'
2. An interpretable model m is now trained over X' and its predictions achieved through f .

Formally, this problem can be defined as:

$$\text{explanation}(x) := \arg \min_{m \in M} L(f, m, \pi_x) \quad (2.6)$$

where m represents a model belonging to the set of interpretable models M , f represents the black box model which predictions have to be explained, π_x represents the dimension of the proximity subset around the sample x and L represents a loss function which evaluates how precise m is approximating f .

To evaluate local surrogate models, *local fidelity* is used. This approach is analogue to the one proposed for global surrogates with the only difference that the fidelity measure is evaluated locally.

While this approach address the performance limitation of global surrogates by providing a more flexible technique, it still has the open challenge of how to formally define what is meant for proximal surrounding of a sample, which is currently approached by choosing among the different kernels the one which makes more sense for a specific application. Moreover, it has been empirically showed and reported in [1], that the variance between explanations of similar points can be very high, making interpretations harder to be trusted.

2.9.5 Shapley Values

Shapley Values represent an approach to estimate individual's contribution to a common result. This concept has been originally introduced in [33] applied to the domain of Game Theory and, only recently, it has been used to explain machine learning models.

In this scenario, the underlying idea of such technique is to estimate the marginal contribution of a single feature to the final prediction. In practice, it is possible to define the Shapley value for a specific feature as the average marginal contribution of that feature over all possible coalitions, where coalitions have to be intended as a fixed set of features.

To make it clearer, let's look at an example. Assuming to have a simple dataset with just three features about online accommodation: price, creation hour and surface area. Let's also assume to have a model which assigns probabilities to instances of being a fraud attempt. To calculate the Shapley value of feature *price*, the following steps have to be performed:

1. Calculate the overall average prediction of the model over the whole input space.
2. Generate all possible coalitions out of the remaining features, which in this case are
 $\{\emptyset, \{creation\ hour\}, \{surface\}, \{creation\ hour, surface\}\}$
3. For every coalition, calculate the marginal contribution as the difference between the prediction over the coalition with and without *price*. Please note that when the feature value is not fixed, such value has to be sampled from the set of possible values for that feature. For example, for the coalition $\{surface\}$, the value of *creation hour* has to be sampled from its distribution. Similarly, when it is stated to calculate the prediction without *price*, it is intended that the value of price for that prediction is sampled from feature's distribution instead of being assigned to the original instance value (i.e. the one which prediction has to be explained).
4. The Shapley value for feature *price* can now be easily calculated as the average of all marginal contributions obtained in the previous point.

Formally, the Shapely Value of a feature x_j has been defined in [25] as the following weighted average of marginal contributions:

$$\phi_j(val) = \sum_{S \subseteq \{x_1, \dots, x_p\} \setminus \{x_j\}} \frac{|S|!(p - |S| - 1)!}{p!} (val(S \cup \{x_j\}) - val(S)) \quad (2.7)$$

where S is the subset of features belonging to a coalition, p is the total number of features on which the model has been trained, X represents the vector of feature values of the instance to be explained and $val(S)$ is a function returning the prediction of the model for feature values S and marginalized over the features that are not included in S .

Assuming to have a prediction function \hat{f} , the prediction function $val(S)$ can be defined as:

$$val_x(S) = \int \hat{f}(x_1, \dots, x_p) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X)) \quad (2.8)$$

where one integral is performed for each feature which is not in the set S .

To clarify the theoretical concept, let's look at a simple example where the set of features is $\{x_0: creation\ hour, x_1: surface, x_2: price\}$ and it is required to calculate the Shapley value ϕ_2 of feature *price*.

The procedure to be followed is:

1. Calculate the average prediction of the model as base value.
2. Generate all possible coalitions $\{\emptyset, \{creation\ hour\}, \{surface\}, \{creation\ hour, surface\}\}$
3. For each coalition assess the marginal contribution between the coalition with and without the value for feature x_2 .
As an example, for the coalition $\{creation\ hour, surface\}$, the marginal contribution is defined as the difference between the predictions over two different samples. The first instance where the features *creation hour* and *surface* have their original values and the feature *price* is sampled from its distribution. While the latter is composed by a tuple where all features have their original values (i.e. original *price* value is assigned).
4. Compute the Shapley value ϕ_2 as the weighted average of all value assessed at step 3.

Due to its strong mathematical basis, this approach guarantees a fair distribution of contributions over the feature space, which was not provided with LIME. On the other hand, in many real cases, when the number of features is sufficiently large, assessing the contribution of all features and relative subsets become unfeasible. As a result, in practical use cases, it is usually combined with sampling techniques which select a subset of features on which the computation has to be performed.

2.9.6 Shapley Additive Explanations (SHAP)

SHAP is a model-agnostic technique used to explain machine learning predictions which combines two of the previously introduced concepts: Shapley Values and LIME. Specifically, SHAP method is based on the idea of representing the explanations as a linear addition of features contribution as follow:

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j \quad (2.9)$$

where M is the maximum coalition size (i.e. the total number of features), $z' \in \{0, 1\}^M$ is a coalition vector in which every element indicates if a certain feature is present (1) or not (0) in the coalition. Moreover, g represents the explanation function and $\phi_j \in \mathbb{R}$ is the Shapley Value (i.e. the marginal contribution) of the j -th feature.

On top of this underlying idea, an efficient implementation of the algorithm named **TreeSHAP** has been proposed in [21]. TreeSHAP, as the name suggests, takes advantage of tree-based model structures to speed up the computation of Shapley Values, so that it can be applied without the need of approximation techniques (e.g. features sampling). As a result, this method can be consider as a class-specific implementation of a model-agnostic explainability method. However, this implementation does not prevent this method to be considered as model-agnostic since the same results achieved using TreeSHAP can be calculated also with standard SHAP but it would requires more computational power.

2.9.7 Counterfactual Explanations

As explanations have the absolute goal of bridging the gap between man and machine and considering the natural comparison capabilities of humans, it becomes intuitive to explain machine learning predictions by the means of counterfactual explanations. This approach is based on using as explanation the differences between the focus instance and a counterfactual one which endorses the following criteria [25]:

- The counterfactual instance should be as similar as possible to the one to be explained, so that the comparison can be done in a meaningful way.
- The counterfactual instance should be evaluated by the black box model with an opposite prediction from the one to be explained. In this specific use case, if we want to explain why a sample has been classified as positive, the counterfactual instance should be aimed at belonging to the negative class. In certain applications, where it is impossible to generate samples with a specific target probability, some tolerance is needed to satisfy this requirement.
- A way of summarizing instances should be present so that the differences between the counterfactual and the original sample can be interpreted easily by humans (e.g. selecting a specific subset of features).

Formally, the generation of a counterfactual instance can be summarized as the following optimization problem:

$$\textit{explanation}(x) := \arg \min_y (D_1(x, y) + D_2(t, P(y))) \quad (2.10)$$

where X is the input space, $x \in X$ is the instance to be explained, $y \in X$ is the counterfactual instance to be generated, D_1 is an arbitrary distance function between two instances belonging to X , P is the prediction function of the black box machine learning model, t is the target prediction and D_2 is another arbitrary distance function computed among two predicted values.

For the sake of this research, a differentiation will not be made between counterfactual and adversarial explanations.

Finally, this approach can be seen as generating the most similar sample to the original one, which allows for a different prediction and then use samples' discrepancies to explain the original predicted value.

Chapter 3

Related Works

In this section a selection of related works from the literature are presented, where automated fraud detection processes have been applied and experimented. However, given the limited amount of publicly available researches about fraud detection in online marketplaces, even more in the housing sub-domain, the vast majority of the following examples are borrowed from different application domains.

For what concerns machine learning interpretability combined to automated fraud detection processes, no publicly available researches have been found in the literature. However, a similar evaluation of machine learning explanations borrowed from a different application domain will be reported.

3.1 Financial Statements

Financial statements are a category of financial documents, generally used by investors and analysts, aimed at reflecting faithfully the financial health of a company. Due to their widespread usage, it is extremely important that these statements are reliable and correct which, unfortunately, it is not always the case.

To address this issue, in [27], a comparison of several statistical and machine learning techniques applied to the detection of fraud between financial statements has been proposed. From a machine learning point of view, the problem has been designed as a binary classification one (i.e. either the statement is legit or it is a fraud) and it has been tackled with the following approaches: Artificial Neural Network, Logistic Regression, Bagging, Stacking, Support Vector Machine(SVM) and Decision Tree.

At first, the performances of the different techniques were compared, highlighting SVM and logistic regression as the most promising ones, and later an analysis of the most important features for each algorithm has been performed revealing which are considered the most important. Finally, both the machine learning models and the insights have been proposed to the community as a framework aimed at improving automated fraud detection in financial statements analysis.

3.2 Credit Card

One of the domain with the largest amount of publicly available researches about Fraud Detection, if not the largest one, is Credit Card Frauds, where for fraud it is intended an illegitimate usage of a credit card aimed at subtracting money from the unaware owner of the card.

In several of these researches, [4, 28, 22, 39], the problem is set as a binary classification task which, in most cases, is tackled through supervised learning approaches. In this scenario, a wide set of different machine learning techniques have been applied, including for example Artificial Neural Network (ANN) [22, 18], Bayesian Belief Network (BBN) [22] and K-Nearest Neighbors (KNN) [38].

The main insights that can be derived from the broad list of examples of different algorithms applied to the task of automated credit card fraud detection is that machine learning algorithms and data mining techniques represent the state-of-the-art for this task.

3.3 Online Marketplaces

As introduced at the beginning of this chapter, the amount of publicly available research about automated fraud detection in online marketplaces is very limited, especially if we narrow the domain to housing marketplaces (i.e. where the offered items are represented by accommodations).

For what concern online marketplaces in general, in [30] promising results have been showed by the application of SVM to detect merchant frauds, namely items published on an online platform with the only objective of stealing money from other customers through fake offerings. However, this implementation differs from the application domain on which this research is based on because it does not include the interaction of humans in the detection process.

Focusing on housing marketplaces instead, in [26] an analysis has been carried out which, even though it did not end up in the implementation of an automated fraud detection process, it revealed several insights about common patterns in online frauds. Among these insights, it has been showed that IP addresses and phone numbers of advertisers are an extremely valuable information that allows to discriminate a large portion of illegitimate usages.

On top of this, the approach which can still be considered the one that suits best the research scope of this thesis is presented in [2], which represents a previous research performed at HousingAnywhere over the topic of automated fraud detection. In that case, an ensemble machine learning model was implemented and it empirically proved that machine learning algorithms largely outperformed rule-based detection systems.

3.4 Human-grounded Explanations Evaluation

As widely explained in section 1, in HousingAnywhere Fraud Detection process the final decision is made by humans, with AI as a support tool. Consequently, it becomes intuitive to let humans evaluate machine learning explanations so that insights not merely based on theoretical concepts can be derived and applied to business applications.

In this view, one work that stands out from the literature is presented in [37]. Specifically, the authors evaluated the impact of machine learning explanations over an alert control business process, where humans were required to validate alerts coming from a machine learning algorithm. In that research, the explanation algorithm that has been chosen is SHAP and the comparison has been made between the same human validation process with and without predictions' interpretation.

At first, the analysis of textual data collected from humans during the experiments showed that explanations have an impact on human reasoning by, for example, bringing the attention over features that may would have been neglected otherwise. However, the results showed that, as opposite to what can be intuitively expected, explanations did not bring any statistically significant improvements in task's effectiveness and mental efficiency.

Chapter 4

Problem Statement

In this section a formal description of the problem that will be tackled with this research is given together with a description of the available data and the technical starting point of the research. Moreover, at the end of the chapter the research questions are presented.

4.1 Problem Formulation

The introduced problem can be formalized as a binary classification task aimed at discriminating whether a newly published item on the platform is a scam attempt or not.

As stated in section 2.3.1, a binary classification task requires to learn the following function:

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is scam attempt} \\ 0 & \text{if } x \text{ is a legit item} \end{cases} \quad (4.1)$$

The goal is then to approximate the function f as good as possible by training on historical fraud and legit cases.

4.2 Data

The data on which the analysis is based can be summarized in four different categories:

- **Listing-specific.** These attributes are represented by all the information about the accommodation that has to be checked. Among these features, the most important that are worth mentioning are rent price, accommodation type, amenities and location.
- **User-specific.** Here the information is represented by a set of features summarizing user profile, such as email address or phone number.

- **Platform-specific.** These attributes are aimed at representing the behaviour that the user had on the online marketplace. In this category it is worth mentioning features such as used IP addresses, time of day of interactions and different locations from which the user accessed the platform.
- **City-specific.** To have a reference point against which a new item can be compared to discover anomalies, a set of city-specific characteristics is used, such as average price or average length of listing’s description.

Finally, it is worth mentioning that the dataset is composed by roughly 15% of fraud attempts and 85% of legit listings.

4.3 Former Model

The aim of this section is to describe the machine learning model as it was at the beginning of the research, so that it can be used throughout the rest of the analysis as a benchmark for performance. We will refer to such model as *legacy model*.

Specifically, the legacy model is composed as an ensemble of six different classifiers, as showed in figure 4.1, where five of them are instances of a gradient boosting algorithm named *Light Gradient Boosting Machine (LGBM)*, introduced in [17]. These five models are built on different temporal selection of data, aimed at mitigating the effect of data shifting and incorporating in the ensemble a temporal sensitive representation.

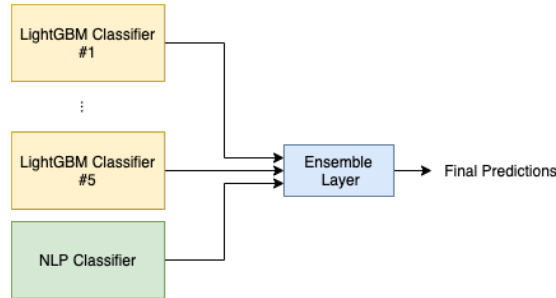


Figure 4.1: Legacy Machine Learning Model

On the other hand, the last model of the first layer is represented by a *Natural Language Processing (NLP)* classifier, aimed at extrapolating knowledge from textual information, which in this use case are represented by accommodations’ description.

These six models are then used to provide predictions independently which are then sent to the ensembling layer that is responsible for combining such predictions into the final ones, as showed in the following formula:

$$P = \sum_{c \in C} p_c \cdot w_c \quad (4.2)$$

where P represents the set of final predictions, c represents a classifier belonging to the class of C of the previously introduced six classifiers, p_c represents the resulting intermediate predictions of model c and w_c represents the contribution weight given from p_c to the final predictions.

4.4 Research Directions

In this section the research directions will be presented, as a result of previously introduced background knowledge and problem settings.

From a machine learning perspective, the task of classifying between legit and illegitimate usages presents several challenges and the subsequent research questions arise. At first, As mentioned in [3], in most domains, fraudsters are always adapting and evolving the way they perpetrate frauds. Moreover, as it has been demonstrated in other use cases from the literature ([3, 19, 18, 10]) how effective features engineering and manipulation steps are in enhancing model's learning phase. On top of this, to make the task even more challenging, the learning process has to be performed in an imbalanced classification scenario.

As a result the following research direction can be defined:

It is possible to design and implement an automated fraud detection process applied to the domain of online marketplaces which:

- *It is robust towards the adapting behaviour of scammers*
- *It includes ad-hoc solutions to deal with categorical and numerical features, including proper features engineering steps.*
- *It is able to learn a meaningful behaviour from an imbalanced dataset.*
- *It is based on a light structure that facilitates debugging, maintenance and future developments, which are extremely valuable characteristics from a business perspective.*

On the other hand, for what concerns interpretable machine learning several other challenges arise. From a general point of view:

*Can machine learning interpretability enhance a fraud detection process?
If so, how to properly integrate machine learning explanations inside an automated fraud detection process?*

By looking more in details, the following research questions arise:

How to faithfully evaluate and compare machine learning explanations applied to an automated fraud detection procedure?

Focusing on interpretable machine learning and once the evaluation framework has been defined:

How does the class of model-agnostic explainability techniques compare to model-based approach?

How the proposed model-agnostic approaches compare among themselves? Which one is more adapt to fraud detection in online marketplaces?

Chapter 5

Approach

In this chapter, it will be explained how the answers to the previously stated research questions will be sought from a general perspective, while the specific details related to individual experiments will be presented in chapter 6.

Specifically, the focus of this section will be on the formalization of how experiments have been performed and evaluated and it will be divided in two parts: the first one, where the approach to performance oriented experiments is presented, and a latter one, where the focus will be on the approach to explainability experiments.

5.1 Performance Experiments Approach

As introduced before, how performance oriented experiments have been conducted is reported here. At first, the settings of the experiments are treated, while in the final part their evaluation is analysed.

It is important to notice that here the focus is on all experiments aimed at seeking an answer to the research questions related to improving the machine learning classifier.

5.1.1 Experiments Setting

As for many machine learning applications, in order to properly evaluate the generalization capabilities of the model, the dataset has been divided into training and testing data. Specifically, the split has been performed over the data arranged chronologically, so to avoid any leakage of information from the future (i.e. samples in the training set which happened chronologically after samples in the test set, possibly biasing model evaluation). Similarly, the same split has been performed over the training set itself so that a smaller set, used for validation and model selection, is generated.

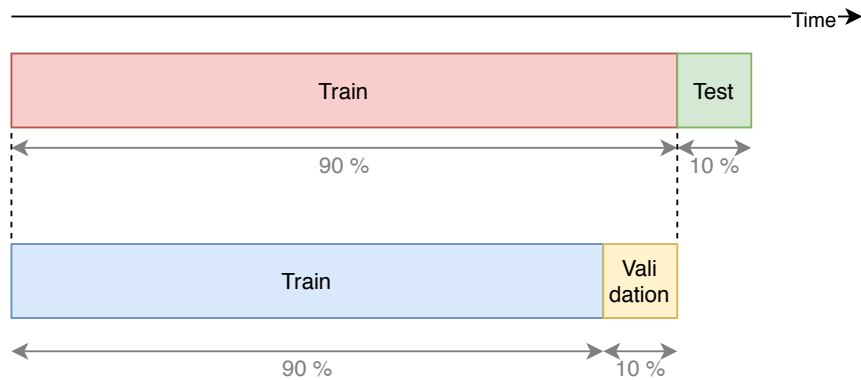


Figure 5.1: Train-Test-Validation splits

As it can be seen from figure 5.2, the experiments pipeline is composed by an initial iterative process, called model selection, where experiments are implemented and then evaluated over the validation set. Once experiments are completed and the actual performance of the selected model has to be established, the model is trained over the whole training set (train + validation) and evaluated over a test set which has not been seen by the model before, so that an unbiased evaluation can be derived.

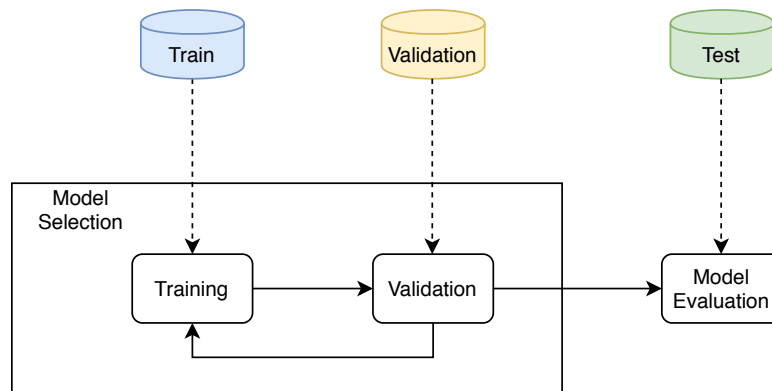


Figure 5.2: Machine Learning Experiments Pipeline

5.1.2 Experiments Evaluation

For what concerns how these experiments have been evaluated, it is necessary to recall some of the metrics introduced in section 2.8.1.

Specifically, all experiments have been evaluated by collecting Precision, Recall, F1-score and Area Under the Precision-Recall Curve. These metrics are then used to make comprehensive comparisons among different experiments with a special attention on Recall, which, due to the sensitiveness of the application domain, is required to be above 80% to consider an experiment as successful.

5.2 Interpretability Experiments Approach

In this section, a description of how explainability techniques have been tested is presented, where at first the structure of an individual experiments is given, while in the last part the evaluation criteria are analysed.

5.2.1 Experiments Settings

In order to understand how interpretability experiments have been performed it is important to introduce the concept of an explanations evaluation task.

For the sake of this research, an evaluation task consists of the human validation (i.e. finally marking an item as scam attempt or legit) of 60 different predictions produced by the machine learning model in the past, thus extracted from an historical database and for which the true class is known.

As it can be seen from figure 5.3, at first 60 listings, which in the past have been marked as fraud attempts by the machine learning model, are selected from the historical database. It is important to specify that, to obtain complete and valuable results, the listings have been chosen from the following different categories:

- R0H0, real legit accommodations correctly validated as legit by the human checking them.
- R0H1, real legit listings accommodations wrongly reported as scam by the human who checked them.
- R1H0, scam attempts wrongly validated as legit items by the human who checked them.
- R1H1, scam attempts correctly reported as scams by the human who checked them.

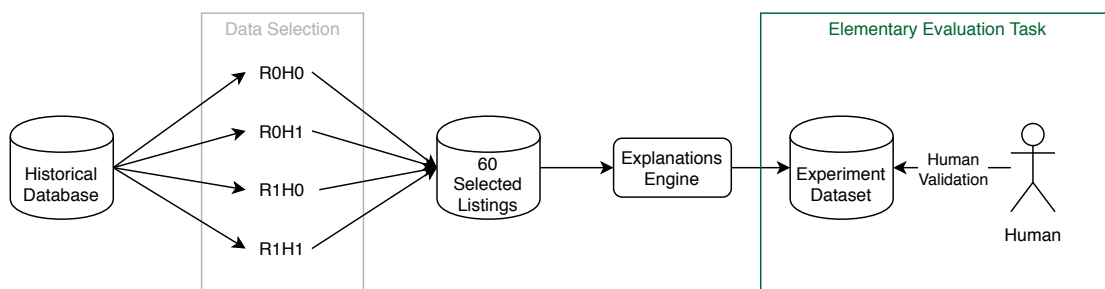


Figure 5.3: Explanations Evaluation Task

Once selected, those 60 listings are analysed by the interpretability algorithm which generates the corresponding explanations. Finally, the selected items and their explanations are validated by the human conducting the experiment and the results are collected.

5.2.2 Experiments Evaluation

Even though several qualitative measures have been proposed in section 2.9.2, it has been decided that the most suitable metric to evaluate the impact of explanations is accuracy, previously defined in formula 2.2, represented by the ratio between correct validations and the total number of validations, which in this case is 60 per task.

The decision of relying on accuracy is based on several reasons. From a business perspective the priority is to validate whether machine learning explanations could improve the interaction between humans and machine. The simpler, yet objective, way of doing that it has been identified in comparing the accuracy of the human validation process with and without machine learning explanations. Furthermore, from an academic perspective, given the novelty of the application domain and the implementation of a novel algorithm, accuracy, thanks to its objective evaluation, it has been considered as a valid choice for this scope too.

It is also important to mention that machine learning explanations could impact several other factors which are worth to be analysed more, such as whether and in which way they affect human's thinking process, if the validation speed changes and if this has an impact on accuracy. However, since it has been decided to focus on accuracy as evaluation metric for the scope of this thesis, these factors will not be evaluated here but it is worth to consider them in any future continuation of this analysis.

Chapter 6

Experiments

In this section, the list of experiments, both successful and not, performed in the scope of this research is presented. Overall, experiments have been divided in two categories: performance oriented, if aimed at improving model performances over the binary classification task, and explanations oriented, if dedicated to enhancing the human capabilities of understanding machine learning predictions. Finally, the results related to the experiments presented in this section can be found in chapter 7.

6.1 Performance Oriented Experiments

At first, the focus will be on performance oriented experiments, namely all those experiments which are aimed at improving the classification performance of the machine learning model. In the scope of this research, several experiments belong to this category, going from the modification of the underlying machine learning structure to changing algorithm's technical detail such as cost function and hyperparameters' value.

It is important to specify that all the experiments reported in this section followed the approach presented in section 5.1 and they are aimed at seeking answers to the research questions related to the improvement of the machine learning classifier.

For the sake of completeness, such research directions are reported again here:

It is possible to design and implement an automated fraud detection process applied to the domain of online marketplaces which:

- *It is robust towards the adapting behaviour of scammers*
- *It includes ad-hoc solutions to deal with categorical and numerical features, including proper features engineering steps.*
- *It is able to learn a meaningful behaviour from an imbalanced dataset.*

- *It is based on a light structure that facilitates debugging, maintenance and future developments, which are extremely valuable characteristics from a business perspective.*

6.1.1 Infrastructure Simplification

The first experiment performed consists in building a more robust, lighter and reliable infrastructure so that all other experiments will follow. To do so, a new model, composed by an individual LightGBM classifier has been built. Such model has then been trained with all historical data, in contrast with the legacy model approach which requires different models built on different temporal slices of data. Furthermore, the new model does not include the NLP classifier which was present in the legacy one.

6.1.2 Objective Function

One of the main limitation of the legacy model is dealing with imbalanced classes, that in this case means that samples of one class are roughly five times more frequent than instances belonging to the other class. As a result, predictions of minority class samples are weaker and the optimal working point of the legacy model is shifted towards the majority class. In practice, this is reflected in skewed predicted probabilities towards 0, assuming class 1 as the minority one. Furthermore, the optimal working point, for this business case, of the legacy model is achieved with a classification threshold of roughly 0.2 (e.g instances with predicted probability greater or equal than 0.2 are classified as belonging to the positive class, otherwise to the negative class) which, besides being not ideal from a performance perspective, it also makes calculated probabilities less meaningful and harder to be understood (i.e. predictions at that point are just a custom score indicating a certain confidence in being a scam or not but they do not represent anymore meaningful probabilities).

To address this issue, the new model has been training using a modified objective function able to deal with imbalanced classes. Specifically, the objective used is the minimization of a weighted Cross Entropy loss function, defined as follow:

$$L(p, y, c) = -(c \cdot y \cdot \log p + (1 - c) \cdot (1 - y) \cdot \log (1 - p)) \quad (6.1)$$

where p represents the predicted probability, y represents the true class to which the instance belongs and c represents a cost factor which can be used to give more importance to a certain type of error.

6.1.3 Missing Values Imputation

As often happens when dealing with data coming from a business case, the dataset on which the analysis is built contains missing values. Even though LightGBM is able to deal with missing values, in critical cases an ad-hoc approach can bring more value. Specifically, by combining the results of the data exploration and the information gathered by debugging the LightGBM model, two features have been imputed manually. The assumption that has been made is Missing Completely At Random (MCAR). MCAR means that there is no relationship between a missing data point and any other values in the dataset, thus making the likelihood that a sample is missing completely random. This assumption is due to the fact that no correlation has been found during data exploration, therefore no reason is present to invalidate MCAR.

Two binary features, which represent respectively whether there is an anomaly in the location data of the user and whether the accommodation includes bedroom information or not, are imputed using a conservative approach of being 0 until proven otherwise.

6.1.4 Bayesian Optimization

LightGBM, as many other complex machine learning algorithms, allows for a huge amount of different configurations by choosing between tens of parameters. To find which configuration performs best in a smart way (i.e. exhaustive search of all possible combinations is computationally unfeasible), Bayesian Optimization has been used. As objective function to be optimized, F1-score has been picked as the best proxy for this specific use case.

The underlying idea of Bayesian Optimization is to approximate the objective function with an initial Gaussian Process and consequently update the distribution with the observed values and Bayes Theorem, where the posterior distribution is the result of the combination of the prior and the observation.

Thanks to distributions approximation, BO is able to produce an approximation of the real unknown objective function and, by balancing the trade-off between exploration (i.e. testing new input values) and exploitation (i.e. testing input values in the surrounding of an optimal point), it is able to generate a set of meaningful input values with a feasible computation complexity.

6.1.5 Categorical Features Encoding

Three different ways of dealing with categorical variables have been implemented and evaluated:

- **One Hot Encoding:** the most common technique when it comes to encoding categorical data. This technique is based on replacing a specific categorical feature f with n new binary features, one per different value in feature f . Moreover, for each instance, only the new binary feature corresponding to the value assumed by feature f is set to 1 while all the others to 0. The main advantage is that it does not introduce any ordinal relation between different categories but it can lead to an exploding number of features.
- **Label Encoding:** is an approach based on substituting each value of a specific feature with a correspondent numerical value. This technique has the advantage of not introducing any new feature but at the risk of generating artificial ordinal relations between different categories.
- **Fisher's Encoding:** novel technique implemented in [17] and based on Fisher's grouping theory introduced in [11], that is meant to be optimal for decision trees. The underlying idea is to split the values of a specific categorical features into groups, where the in-group variance is minimized using Fisher's technique, and then let the decision trees find the optimal split over these subsets.

6.1.6 Numerical Features Enhancement

In this section, all the experiments related to the enhancement of numerical features are reported. In order to find the best representation possible of numerical features for the machine learning model, the following experiments have been implemented:

- Discretization of numerical features using quantiles.
- Normalization of numerical features by the mean of different approaches such as L1-norm, L2-norm and maximum value.
- Applied Principal Component Analysis (PCA) as a numerical features selection technique. For PCA it is intended a statistical technique aimed at finding an alternative representation (i.e. with a different set of axis) of the input data where the variance, used a proxy for amount of information, over these axis is maximised. Then, the features selection step is realized by selecting only a subset of new components (i.e. principal components) which account for most of the variance, thus reducing the features space dimensions.

6.1.7 Boosting Technique

As an alternative to the traditional boosting technique, a novel approach, namely Dropout meets Multiple Additive Regression Trees (DART), introduced in [36] have been implemented and tested. The idea behind it is to apply the concept of dropout, borrowed from the domain of deep neural network, to boosted decision trees. Similarly to the original dropout technique, the aim is to improve the generalization capabilities of the final machine learning model by skipping some training iterations for certain individual trees.

6.1.8 Minor Experiments

In this section, a selection of experiments that have been quickly implemented and tested are reported. However, since they are partially out the scope of this research, they have not been investigate deeply. Among these, it is worth mentioning:

- Attempt to ease the learning task by balancing the dataset with synthetic samples generated with the SMOTE algorithm.
- Ensemble a simple Neural Network with the new simplified infrastructure (i.e. a single LightGBM model). Attempts have been made both by training the Neural Network on the same features and on different representations of the input space, such as PCA and Clustering.
- Feed a LightGBM model with new engineered features based on Principal Component Analysis (PCA), K-means clustering over numerical features and K-means clustering over PCA components.
- Train a LightGBM model with Focal Loss, a novel loss designed for dense object detection in computer vision application and empirically proved to be competitive also for tabular data in unbalanced scenarios.

6.2 Interpretability Oriented Experiments

In this section, all experiments focused on explaining the prediction of the underlying machine learning model are reported. Specifically, four different approaches will be analysed: Model-based, Local Surrogate, SHAP and Adversarial explanations.

All experiments presented here are structured on the corresponding approach presented in section 5.2 and aimed at answering interpretability-related research questions, specifically:

*Can machine learning interpretability enhance a fraud detection process?
If so, how to properly integrate machine learning explanations inside an automated fraud detection process?*

How to faithfully evaluate and compare machine learning explanations applied to an automated fraud detection procedure?

*How does the class of model-agnostic explainability techniques compare to model-based approach?
How the proposed model-agnostic approaches compare among themselves? Which one is more adapt to fraud detection in online marketplaces?*

6.2.1 Model-based Explanations

Even though, as explained previously, the focus of this research is on model-agnostic explainability methods, it has been decided to investigate a model-specific method to have a baseline against which model-agnostic approaches can be compared.

As stated before, the underlying model is an instance of LightGBM, which among other offered features, it allows to store a value for all trees' leaves while generating a specific prediction. Such values are used to assign node importance in the final prediction and therefore they can be used to assess which features contribute more in generating the final prediction value.

As introduced before, in this specific scenario (i.e. binary classification task) the final outcome of the model is calculated as follow:

$$f(x) = \frac{1}{1 + \exp^{-x}} \quad (6.2)$$

where the outcome $f(x) \in (0, 1)$ represents the probability of an instance of belonging to the positive class and $x \in \mathbb{R}$. While generating a prediction, the LightGBM model goes through all the trees which is made of and each traversed leaf contributes with its features to increasing or decreasing the values of x . During the training process, such contributions are established so that the model can generate meaningful predictions in

testing phase. For example, let's think about a trivial example where instances are composed by only two features: *name* and *age*. Let's also assume that all and only the instances which have *name* = *Federico* and *age* ≥ 23 belongs to the positive class. As a consequence, if we take a simple instance composed by *name* = *Federico* and *age* = 18, the value of x will be affected by a positive contribution from the feature *name* and a negative one caused by *age*. By tracking all these contributions to the final value of x , which consequently generates the prediction probability, it is possible to determine which features are considered to be more important in assessing the final decision.

For the purpose of the application domain, the resulting explanations have been generated by providing the three most influential factors both towards the positive and the negative class as it can be seen from the following examples:

Model-based Explanation Example

- The model thinks it is a possible scam because:
 - User's email domain is *.web*
 - The accommodation is 300€ cheaper than city's average rental price.
 - User and accommodation are in different countries.
- On the other hand, for the following reasons it can be legit:
 - User connected his/her Facebook profile.
 - Listing has been created using MacOS as OS.
 - User's email contains only one number.

6.2.2 Local Surrogate Explanations

As introduced before, local surrogate models represent a model-agnostic approach to explainability where the behaviour of a black box model is explained by interpreting a surrogate one which belongs to the class of interpretable machine learning models and approximate accurately enough the behaviour of the main algorithm for a given sample.

In this experiment, as surrogate explainable model a binary ridge classifier has been implemented, that is converting the classification labels to $-1, 1$ and than approaching the problem as a ridge regression task. The predictions of the surrogate model are generated through the following formula:

$$y = w_0 + \sum_i w_i \cdot x_i \quad (6.3)$$

Such predictions are then considered explainable because each feature x_i contribution can be assessed by its associated weight w_i .

Consequently, the explanations are generated by providing humans with the most important features (i.e. the ones having their weights with the largest magnitude). The features associated with the largest positive weights are the ones providing the most important contribution towards the positive class, while, on the other hand, the features with the largest negative weights are the ones increasing the instance's probability of belonging to the negative class. The produced explanations are presented with the same layout as the model-based ones in section 6.2.1.

6.2.3 SHAP

TreeSHAP, introduced in section 2.9.6, is a efficient implementation of SHAP explainability technique, which speeds up the computation by leveraging the tree-based structure of the underlying machine learning model.

To recall, this approach is an attempt to explain machine learning predictions by assigning to each feature a contribution factor (i.e. Shapley value), calculated as the weighted average marginal contribution of a specific feature applied to different input samples.

Similarly to previous methods, explanations for an input sample are generated by providing humans with the features which have the highest marginal contribution, either negative or positive, to the final prediction. As a result, explanations are presented with the same layout as the one in section 6.2.1.

6.2.4 EVolutionary ADversarial Explanation (EVADE)

The last approach that has been implemented in this research belongs to the class of adversarial explanations. Particularly, the idea is to exploit the natural ability of humans in comparing objects by providing a set of instances with specific characteristics alongside the one to be explained. In order to make the comparison meaningful, the adversarial instance has to enforce two basic principles:

- Being as similar as possible to the instance to be explained, so that it differs from the original instance by the smallest subset of important features.
- Being evaluated by the machine learning model in an opposite way with respect to the instance to be explained, so that an adversarial comparison can be made.

To do so, a novel technique named **EVADE** has been developed and implemented, that by means of genetic algorithms (introduced in section 2.7.4), it is able to produce synthetic instances and generate machine learning explanations out of them.

At the end of the genetic optimization procedure, the best candidate (i.e. the one with the highest fitness value) of the last population is than used as a counterexample of the instance to be explained. The core of the optimization method relays in designing

the correct fitness function so that, at the end of the optimization process, the best candidate matches the criteria set at the beginning: being as similar as possible to the original instance and, at the same time, being evaluated in a significantly different way from the machine learning model.

In order to generate the initial population, instances are created using sampling techniques. The sampling procedure assumes that each feature value's is sampled from a uniform distribution, where feature's domain is assessed by looking at the values assumed by a certain feature over the training instances. It is important to mention that, while for categorical features the number of possible values is limited, this does not hold for continuous features. Therefore, it has been decided to discretize all numerical features into buckets so that, also for these features, the domain size becomes limited. While the assumption of underlying uniform distribution may not be valid for all use cases, it reveals to be a valid trade-off between accuracy and complexity for this research.

Similarly, the previous sampling procedure is also used when mutation has to be applied, with the difference that instead of being applied at instance-level (i.e. for all features), it is applied only to the features that have to be mutated.

For the purpose of this experiment, an ad-hoc fitness function has been defined as the weighted sum of two components: a first part accounting for the discrepancy between the best candidate's prediction and the one of the original instance, while a second part considers how different the two instances are.

Formally, the first part is defined as follows:

$$\Delta_{max} = \max(1 - t, t) \quad (6.4)$$

$$f_{pred}(X', p, t) = \frac{\Delta_{max} - |p(X') - t|}{\Delta_{max}} \quad (6.5)$$

where $t \in [0, 1]$ represents the target prediction probability for the candidate (e.g. if it is required to explain an instance belonging to the positive class, a good value for t can be 0) while Δ_{max} is the maximum possible discrepancy between a certain probability and the target one in a binary classification problem (i.e. predicted probabilities belongs to $[0, 1]$). Furthermore, f_{pred} is a function that given a candidate instance to be evaluated X' and a prediction function p and a target prediction probability t returns a score in the domain $[0, 1]$. The prediction function p , which returns the predicted probability given a specific instance, represents the behaviour of the underlying black box machine learning model. Consequently, f_{pred} evaluates to 1 when the instance X' is evaluated with a prediction probability equal to the target one, while it returns 0 when such prediction has the highest discrepancy possible (i.e. Δ_{max}) from the target one.

As introduced before, the second building block of the fitness function focuses on the similarity between a candidate and the original instances. In practical machine learning

applications it is common to have instances composed by both categorical and numerical features. As a consequence, a procedure for assessing similarities between instances must provide a way to deal with these different types of features. To do so, the similarity function has been designed in two different parts.

The first component has been dedicated to numerical features and calculated using the following formula:

$$f_{num}(X_{num}, X'_{num}) = 1 - sim(X_{num}, X'_{num}) \quad (6.6)$$

$$f_{num}(X_{num}, X'_{num}) = 1 - \frac{X_{num} \cdot X'_{num}}{\|X_{num}\| \times \|X'_{num}\|} = 1 - \frac{\sum_i^{nf} x_{num_i} \cdot x'_{num_i}}{\sqrt{\sum_i^{nf} x_{num_i}^2} + \sqrt{\sum_i^{nf} x'_{num_i}^2}} \quad (6.7)$$

where $f_{num} \in [0, 1]$ is the fitness function related to numerical features and $sim \in [0, 1]$ represents the cosine similarity between two instances X and X' . These instances represent the subsets of the original instances obtained by considering only numerical features. Furthermore, in the breakdown of the similarity function sim , nf represents the number of numerical features, x_{num_i} represents the value of the instance for the i -feature

As introduced before, an additional part which extends the usage of the technique by keeping into account categorical features has been designed in the following way:

$$f_{cat}(X_{cat}, X'_{cat}) = \frac{|X_{cat} \cap X'_{cat}|}{|X_{cat}|} \quad (6.8)$$

where $f_{cat} \in [0, 1]$ represents the fitness function calculated over categorical features as the ratio between features which share the same value between X_{cat} and X'_{cat} over the total number of categorical features $|X_{cat}|$. In this case, X_{cat} and X'_{cat} represents the features subsets of the original instances where only the categorical features are considered.

Finally, the global fitness function is composed by the weighted sum of all the components presented so far and represented by the following formula:

$$f(X, X', p, t, W) = f_{pred}(X', p, t) \cdot w_{pred} + f_{num}(X_{num}, X'_{num}) \cdot w_{num} + f_{cat}(X_{cat}, X'_{cat}) \cdot w_{cat} \quad (6.9)$$

Which can be visually represented as in figure 6.1.

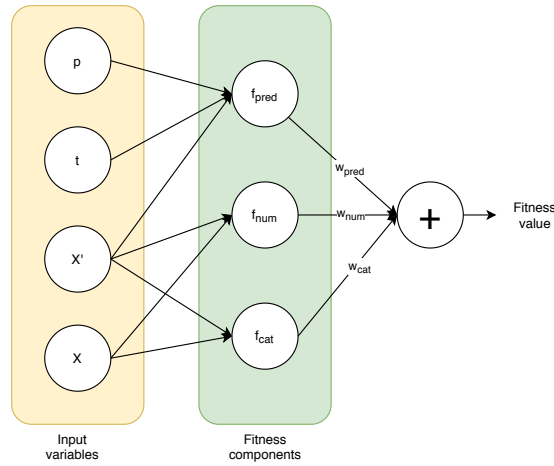


Figure 6.1: Fitness Function Computation

Where, to recall, the parameters are defined as follows:

- X is the original instance to be explained.
- X' is the synthetic generated counterfactual example of which the fitness value has to be assessed.
- p is the prediction function of the machine learning model which needs to be interpreted.
- t represents the target prediction probability that the generated instance X' should ideally have.
- W is composed by the weights of the individual components of the fitness function, namely $w_{pred}, w_{num}, w_{cat}$.

At the end of the iterative evolution process, the final population is used to generate the explanations. To do so, the best candidate, represented by the instance with the highest fitness value in the population, is used as the counterfactual sample. However, to make comparisons understandable and actionable by humans, the similarities and differences between the synthetic and the original instance have to be summarized by proposing only a meaningful subset of important features. In this implementation, the importance of a feature has been based on how many synthetic instances, belonging to the final population, have the value of a certain feature changed from the original one. Such criteria has been designed as follows:

$$g(X, X', f) = \begin{cases} 1 & \text{se } X_f = X'_f \\ 0 & \text{otherwise} \end{cases}$$
$$imp(P, f, X) = \frac{\sum_{i=1}^{|P|} g(X, P_i, f)}{|f|} \quad (6.10)$$

Where $g(X, X', f)$ represents a simple function that evaluates to 1 if, for a specific feature f , both instances X and X' have the same value, 0 otherwise. The importance for a feature f , over a population P and the original instance to be explained X , is then calculated by $imp(P, f, X)$ as the number of samples in the population where feature f assumes a different value than the original one in X . Moreover, the result is then normalized by the cardinality of the feature (i.e. the number of different values it can have) so that the impact of high-cardinality features on importances is mitigated.

As an example, if we assume to have instances composed by just two features f_1 and f_2 , a population of ten candidates and f_1 is in all the samples different from the original value while f_2 differs from the original one only for half of the cases. Furthermore, let's assume that $|f_1| = 10$ (i.e. f_1 can have ten different values) and $|f_2| = 2$. As a result, the importance value of $imp(P, f_1, X) = \frac{10}{10} = 1$ and $imp(P, f_2, X) = \frac{5}{2} = 2.5$. As it can be seen from the previous example, even though f_1 has assumed a value different from the original one for more times than f_2 , thanks to the normalization factor, f_2 has been evaluated as more important.

As a last step, all features which have different values in the best candidate and in the original instance are selected and ranked accordingly to their importance, assessed with the previously stated procedure. The final explanation is then generated by providing humans with the comparison of the top most important features between the synthetic and original instances. The idea is basically to show humans which values would be required the original sample to have so that it would be evaluated in the opposite way.

For the purpose of this application, the top five important features are proposed with the following structure:

Adversarial Explanation Example

The model thinks it is a possible scam because:

- Listing has been published right after user registration instead of after 24 hours.
- Listing's rent is 200€ cheaper than average of the city instead of having a comparable rent price w.r.t. to city average.
- Listing's description is empty, instead of being 10000 characters long.
- Listing has been published in July instead of January.
- Listing has been created using an unidentified OS instead of Mac-OS.
- Listing's minimum rental period is not defined instead of being defined.

Chapter 7

Results

In this section the results of the experiments mentioned in section 6 are reported following the same structure of that section: a first part dedicated to the results achieved in performance oriented experiments, while in the last part all results related to interpretability are reported.

7.1 Performance Oriented Results

For the purpose of this research, multiple metrics have been taken into account so that most aspects of the behaviour of the model can be observed and a comprehensive decision can be taken. Specifically, from the set of metrics introduced in 2.8.1, Precision, Recall, F1-score and Area Under the Precision-Recall Curve have been selected.

The final machine learning model has been achieved as a combination of all successful experiments presented in section 6.1 that proved to improve the performances of the original model, introduced in 4.3 and reported here for the sake of completeness:

Precision	Recall	F1-score	PR AUC
0.4986	0.8036	0.6154	0.7399

Table 7.1: Legacy Model Performance

7.1.1 Successful Experiments

The first experiment that has been implemented, so that it would work also as a foundation for all following ones, it has been the machine learning infrastructure simplification as described in section 6.1.1.

The performances of the new simplified model are the following:

Precision	Recall	F1-score	PR AUC
0.5357	0.8015	0.6422	0.7229

Table 7.2: Simplified Model Performance

As it can be seen for table 7.2, the new model achieved a comparable performance with respect to the legacy one (table 7.1) by having slightly lower Recall and PR AUC but higher Precision and F1-score. As a result this simplified model will be used as the basis for next experiments since the trade-off between performance and model complexity it has been considered worth it.

As a second step, it has been empirically showed that the categorical features encoding that worked best for the scope of this research is the one based on Fisher’s grouping theory previously stated in section 6.1.5. Specifically, the following table compares the performance of the different encoding techniques:

Encoding	Precision	Recall	F1-score	PR AUC
One-Hot	0.5299	0.813	0.6416	0.7734
Label	0.4976	0.8015	0.614	0.7306
Fisher	0.5357	0.8015	0.6422	0.7229

Table 7.3: Categorical Features Encoding Comparison

As a consequence of these results, it has been decided to continue with the tree-optimized encoding based on Fisher’s grouping. Please note that Fisher’s encoding has been used also by the simplified model presented before in table 7.2, therefore they have the exact same performances.

On top of new infrastructure, a new objective function, presented in section 6.1.2, has been used. To recall, it is composed by the minimization of a weighted Cross Entropy loss function, defined as follow:

$$L(p, y, c) = -(c \cdot y \cdot \log p + (1 - c) \cdot (1 - y) \cdot \log(1 - p)) \quad (7.1)$$

where p represents the predicted probability, y represents the true class to which the instance belongs and c represents a cost factor which can be used to give more importance to a certain type of error.

After fine tuning the parameters to be assigned to c , the overall performances of the new model increased and the optimal working point of the classifier shifted towards a less skewed decision threshold. In practice, this can be seen by the model producing on average higher predicted probabilities (i.e. it is *easier* that an instance belonging to the

positive class will have a predicted probability greater or equal 0.5), allowing for achieving comparable recall values to the legacy model but with a more traditional decision threshold of 0.5 instead of 0.2.

Overall, the performances of this model are:

Precision	Recall	F1-score	PR AUC
0.5687	0.8053	0.6667	0.7696

Table 7.4: Model Performance with Weighted Loss

Since performances increased significantly, this objective function will be used for training the LightGBM classifier also in next experiments.

Furthermore, another experiment that empirically proved to be successful is the Missing Values Imputation proposed in section 6.1.3. Specifically, the achieved performances at the end of this experiment can be found in the following table:

Precision	Recall	F1-score	PR AUC
0.5801	0.8015	0.6731	0.7681

Table 7.5: Model Performance with Missing Value Imputation

As it is shown in table 7.5, the imputation increased precision, recall and F1-score while it slightly hit PR-AUC. Overall, since the new performances are considered to be better with respect to the use case, this imputation will be performed in future experiments.

As a last successful performance oriented experiment, it is worth to mention that performing hyper-parameters tuning by the mean of Bayesian Optimization led to the following performance:

Precision	Recall	F1-score	PR AUC
0.6046	0.8053	0.6907	0.7681

Table 7.6: Model Performance after Bayesian Optimization

Table 7.6 shows again a performance improvement with respect to the previous experiment and therefore Bayesian Optimization has been selected to become a component of the final machine learning model.

7.1.2 Discarded Experiments

Even though several experiments brought positive results, some of them did not improve the performance of the model and therefore they have been discarded. For the sake of completeness their results will be reported in this section, while the implementation details behind them are explained extensively in section 6.1.

Experiment	Precision	Recall	F1-score	PR AUC
Discretization	0.544	0.8015	0.6481	0.7667
L1 Normalization	0.5765	0.8053	0.672	0.7722
L2 Normalization	0.541	0.8168	0.6505	0.7739
Max Normalization	0.5	0.8168	0.6203	0.7348
PCA	0.4277	0.813	0.5605	0.7008
DART boosting	0.5146	0.8092	0.6291	0.7662
Clustering over PCA	0.5558	0.8168	0.6615	0.7619
Over-sampling with SMOTE	0.5518	0.813	0.6574	0.7716
Focal Loss	0.8107	0.5229	0.6357	0.754
Ensemble with Neural Network	0.6046	0.8053	0.6907	0.7657

Table 7.7: Discarded Experiments Performances

As it can be seen from table 7.7, several experiments did not achieve good enough performances to be included in the final model. Moreover, it is important to notice that all experiments listed in this table are built on the basic LightGBM model introduced before.

7.1.3 Final Model Performance

In this section, a summary of the previous results will be made with an exclusive focus on the successful experiments and their contribution to the final model.

First, a breakdown of the single improvements will be made followed by a comprehensive comparison aimed at capturing the whole picture of performance related improvements, from the beginning to the end of the experiments.

Looking at the individual experiments' contribution, the following table can be drawn:

Experiment	Precision	Recall	F1-score	PR AUC
Initial Baseline	0.4986	0.8036	0.6154	0.7399
Model Simplification	0.5357 (+7.4%)	0.8015 (-0.3%)	0.6422 (+4.3%)	0.7299 (-1.4%)
Objective Function	0.5687 (+6.2%)	0.8053 (+0.5%)	0.6667 (+3.8%)	0.7696 (+5.4%)
Missing Values Imputation	0.5801 (+2%)	0.8015 (-0.5%)	0.6731 (+1%)	0.7681 (-0.2%)
Bayesian Optimization	0.6046 (+4.2%)	0.8052 (+0.5%)	0.6907 (+2.6%)	0.7681 (+0%)

Table 7.8: Experiments Improvements Breakdown. Between brackets, the relative percentage improvement with respect to the previous iteration is given.

By looking at table 7.8, the step by step improvements can be clearly seen. It is worth mentioning that the experiment which brought the overall largest improvements is the introduction of the cost-sensitive objective function which revealed to be especially suitable for an unbalanced dataset. Moreover, the impact of the new objective function on the working point of the classifier (i.e. making the classifier working in a balanced scenario and shifting the working point towards a more traditional decision threshold of 0.5) can be also seen from the relevant improvement brought by this experiment in the Area Under the Precision-Recall Curve, with a relative increment of +5.4%. The underlying motivation relies in the concept that the more the working point of the classifier is skewed (i.e. very low or high decision threshold to get the desired performances) the smaller the area where the classifier works best is, and consequently the lower the PR AUC value will be.

On the other hand, the comprehensive picture given by the set of performance oriented experiments can be defined as follows:

Experiment	Precision	Recall	F1-score	PR AUC
Initial Baseline	0.4986	0.8036	0.6154	0.7399
Final Model	0.6046 (+21.3%)	0.8053 (+0.2%)	0.6907 (+12.2%)	0.7681 (+3.5%)

Table 7.9: Overall Performance Improvements

As it can be seen from table 7.9, the final machine learning model achieved a 21.3% increase in precision with respect to the starting point of the experiments while keeping

basically the same recall. As a consequence, also F1-score is 12.2% higher and PR AUC is 3.5% higher than the initial model.

7.2 Interpretability Oriented Results

In this section the results related to interpretability oriented experiments will be stated. Specifically, four different explanation methods will be analysed with their performances evaluated both at application level and human level (2.9.2).

Specifically, humans were required to validate or not the predictions of the machine learning model, simulating the actual process of scams detection at HousingAnywhere presented in section 2.1. The main difference from the real scams detection process is that machine learning predictions were presented not only with a probability score of being a scam attempt but also with the predictions' explanation, so to assess if they can be used to facilitate the process of human interpreting machine learning predictions.

The results that will be presented in the rest of the section have been achieved over interpretability evaluation tasks structured as presented in section 5.2.

7.2.1 Explanations Accuracy

It is possible to divide the explanation experiments by two different criteria:

- **Evaluation Level:** as introduced before, evaluating explanations is far from being a trivial task. Specifically, due to the nature of the use case of this research, a human-based evaluation is required. Therefore, two of the methods introduced in section 2.9.2 that match this criteria have been used. First, an application level evaluation has been performed by assessing the interpretability impact over humans with prior domain knowledge over the task, and secondly a human level evaluation, where no prior knowledge is required.
- **Explanation Method:** another differentiation among explanations experiments can be made by looking at which method has been used to generate such explanations. Specifically, in the scope of this research, four different methods have been implemented: Model-based, Local Surrogate, SHAP and EVADE.

Evaluation Level

For these experiments, four humans participated and validated the machine learning predictions. Specifically, half of the participants were domain experts at the beginning of the experiments, while the other half was new to the domain of scams detection, so that both human-level and application-level evaluations have been assessed.

In table 7.10, the performances over the two different evaluation levels are reported, both overall and at individual task level. Specifically, for each task and category the

accuracy is stated, calculated as the ratio of correct validations over the total number of validation, as explained in section 5.2.

Evaluation	R0H0	R0H1	R1H0	R1H1	Overall
Human Level #1	0.65	0.7143	0.6154	0.55	0.6167
Human Level #2	0.7	0.8571	0.4615	0.65	0.65
Human Level #3	0.8	0.8571	0.7692	0.7	0.7667
Human Level #4	0.8	0.8571	0.3077	0.55	0.6167
Human Level Overall	0.7625	0.8214	0.5385	0.6125	0.6708
Application Level #1	0.65	0.8571	0.8462	0.7	0.7333
Application Level #2	0.7	0.2857	0.6923	0.7	0.65
Application Level #3	0.55	0.7143	0.7692	0.65	0.65
Application Level #4	0.8	0.8571	0.9231	0.7	0.8
Application Level Overall	0.675	0.6786	0.8077	0.6875	0.7083

Table 7.10: Evaluation Level Accuracy, each cell contains the corresponding accuracy

Since the overall results in table 7.10 do not show a clear performance discrepancy between humans with and without background knowledge, a proper scientific validation has been performed.

Specifically, a statistical paired t-Test, introduced in section 2.8.2, has been implemented to assess if the performance difference between the two evaluation approaches is statistically significant. Therefore, the following hypotheses are formulated:

$$\begin{cases} H0 : \mu_1 = \mu_2 \\ H1 : \mu_1 \neq \mu_2 \end{cases}$$

Where μ_1 and μ_2 represent respectively the mean accuracy of evaluation at human and application levels.

To assess how t-value is calculated is it necessary to recall the following definition:

$$t\text{-value} = \frac{\sqrt{k} \cdot \bar{\delta}}{\sqrt{\frac{\sum_i (\delta - \delta_i)^2}{k-1}}} \quad (7.2)$$

With the observed data, the variables assumed the following values:

- $\alpha = 0.05$
- The set of errors $\delta = \{0.17, 0.03, 0, 0.03\}$.
- The median error $\bar{\delta} = 0.0458$.
- The degrees of freedom $df = 3$.
- The t-value = 1.84
- The associated p-value = 0.16

As a result, p-value $> \alpha$, thus it is not possible to refuse the null hypothesis with a 95% statistical confidence since such p-value indicates a 16% probability that the discrepancy in performance is due to randomness.

Explanation Method

In this section, the performances of the four different interpretability methods will be presented. Furthermore, it is important to specify that, since from the results of the previous section it emerged that there is no statistically significant difference between the performances of Human Level Evaluation and Application Level Evaluation, all further analysis will not take into account this differentiation anymore. Finally, each explanation method has been tested over two tasks and compared both with a general baseline and among the other explanation approaches.

First, it is worth mentioning that a baseline for comparison has been created by evaluation human's validations in a scenario with just the prediction probabilities and without explanations. Such baseline, which performances can be found in table 7.11, is used as a starting point for the evaluation of all interpretability approaches.

Evaluation	R0H0	R0H1	R1H0	R1H1	Overall
Baseline	0.8	0.5714	0.3077	0.5	0.5667

Table 7.11: Baseline without Explanations, each cell contains the corresponding accuracy

As introduced in section 6.2, the first explainability approach that has been implemented is called **Model-based** and, even though it does not fully match the scope of this research about Model-agnostic interpretability methods, it has the purpose of providing a solid reference point for all future experiments. In the following table, the

performances of humans over Model-based explanations are reported, both overall and at individual task level.

Evaluation	R0H0	R0H1	R1H0	R1H1	Overall
Model-based #1	0.65	0.8571	0.8461	0.7	0.7333
Model-based #2	0.65	0.7143	0.6154	0.55	0.6167
Model-based Overall	0.65	0.7857	0.7308	0.625	0.675

Table 7.12: Model-based Explanations Accuracy

Once the model-based reference performances have been collected, it is time to see how model-agnostic techniques compare to it.

The first one to be introduced is **Local Surrogate**, presented in 2.9.4 and 6.2.2. This approach to interpretability is based on simulating the behaviour of the underlying black box machine learning model with an white box one, which aimed at being a reliable approximation for a specific subset of the input space. Such method achieved the following results:

Evaluation	R0H0	R0H1	R1H0	R1H1	Overall
Local Surrogate #1	0.7	0.8571	0.4615	0.65	0.65
Local Surrogate #2	0.7	0.2857	0.6923	0.7	0.65
Local Surrogate Overall	0.7	0.5714	0.5769	0.675	0.65

Table 7.13: Local Surrogate Explanations Accuracy

Another Model-agnostic explainability method experimented is SHAP, introduced in section 2.9.6 and 6.2.3, which showed the following performances:

Evaluation	R0H0	R0H1	R1H0	R1H1	Overall
SHAP #1	0.55	0.7143	0.7692	0.65	0.65
SHAP #2	0.8	0.8571	0.7692	0.7	0.7667
SHAP Overall	0.675	0.7857	0.7692	0.675	0.7083

Table 7.14: SHAP Explanations Accuracy

Finally, the last model-agnostic method which have been used in this research is **EVADE**, introduced in section 2.9.7 and 6.2.4. Such approach achieved the following performances:

Evaluation	R0H0	R0H1	R1H0	R1H1	Overall
EVADE #1	0.8	0.8571	0.9231	0.7	0.8
EVADE #2	0.8	0.8571	0.3077	0.55	0.6167
EVADE Overall	0.8	0.8571	0.6154	0.625	0.7083

Table 7.15: EVADE Accuracy

Given the individual tasks' performances, it is interesting to look at the method by method comparison summarized in the following table:

Method	R0H0	R0H1	R1H0	R1H1	Overall
No Explanations	0.8	0.5714	0.3077	0.5	0.5667
Model-based	0.65	0.7857	0.7308	0.625	0.675
Local Surrogate	0.7	0.5714	0.5769	0.675	0.65
SHAP	0.675	0.7857	0.7692	0.675	0.7083
EVADE	0.8	0.8571	0.6154	0.625	0.7083

Table 7.16: Methods Comparison Accuracy

By looking at table 7.16, the following insights can be derived:

- All interpretability methods proved to significantly improve the accuracy of humans' validations with respect to a scenario without explanations.
- Even though Local Surrogate achieved higher accuracy than the case where no explanations were given, it also empirically proved to be not as competitive as the other methods on this use case.
- Even though Model-based explanations took advantage of model's internal information, it has been showed that both SHAP and EVADE explanations were able to outperform it.
- Even though SHAP and EVADE explanations achieved the same accuracy value overall, by looking at the categories breakdown a pattern arises. While SHAP seems to have more balanced score throughout all categories, EVADE shows higher variance where very high accuracy scores have been achieved on legit items validation while performances struggle for what regards scam attempts. An intuitive justification to EVADE higher variance can be found in the process of generating explanations. Specifically, EVADE is based on another machine learning technique (i.e. Genetic Algorithms) thus introducing an additional moving part in the system. However, this represents only an hypothesis which testing is outside the scope of this thesis and represent an interesting topic for future researches.

- The categories where the validation process benefits more from explanations are R0H1 and R1H0, which represent items on which the validation process failed in the past.

Chapter 8

Conclusions

In this section, all the results achieved throughout the mentioned experiments will be summarized and their impact on the process of Fraud Detection at HousingAnywhere analysed. At first, a general summary will be given, with a specific focus on answering the research questions asked at the beginning of the analysis. Meanwhile, in the second part of the section, a more detailed analysis of the contributions brought by this research will be given, both from an academic and a business perspective. Finally, the section will end with considerations related to the limitations of this research alongside pending questions representing the basis for a continuation of this work.

8.1 General Contribution

This research proposes an implementation of an automated process for Fraud Detection applied to an online housing marketplace, where humans and machine learning algorithms work together to produce an accurate and solid business process for detecting scam attempts and consequently prevent scams from happening.

To make it possible, at first, a machine learning model which tackles the task of classifying items between possible scams and legit listings has been developed. For what concerns the binary classification task, the final machine learning model that has been presented achieved a 60% precision and a 80% recall, with an increment of +21% in precision and comparable level of recall from the starting reference point.

However, given the sensitivity of this application domain, where the decision of marking an item as a possible scam has impact on customers, the final classification decision has to remain in the hands of humans and artificial intelligence acts as a support tool. Therefore, it has been decided to implement a machine learning interpretability process so to ease the interaction between humans and machines. This specifically unfolds in generating, alongside *traditional* predictions, what have been called machine learning explanations. In details, explanations consist of textual information proposed in a human comprehensible way which bring insights on what causes the machine learning

model to generate a certain prediction. In the current application domain, explanations are represented as a set of factors which have brought the model to mark certain items as possible scam attempts.

Concretely, the integration of machine learning interpretability into the business process has been achieved through the implementation, evaluation and comparison of four different explainability techniques. Among these, SHAP and EVADE were the two approaches that achieved the best results by increasing the accuracy of humans predictions' validation up to 70%, with a +25% improvement with respect to a scenario were no explanations were given.

Especially, a relevant contribution is represented by the novel approach to Adversarial Explanations that has been developed, named EVADE. By the means of Genetic Algorithms, synthetic adversarial instances are generated so that they can be used as a counterexample explanation to the original model prediction. On top of the novelty of this approach, EVADE also empirically achieved state-of-the-art performances by obtaining the highest accuracy among the different methods together with SHAP.

The resulting fraud detection process can be found in figure 8.1, where it is possible to see that each new item is scanned by the machine learning model and if it is thought to be a scam attempt, it is forwarded to the interpretability module which generates, alongside the prediction probability, the machine learning explanations. Aside the new flow, highlighted in green, it is also shown the legacy flow, namely the process were no explanations were generated, for comparison purposes.

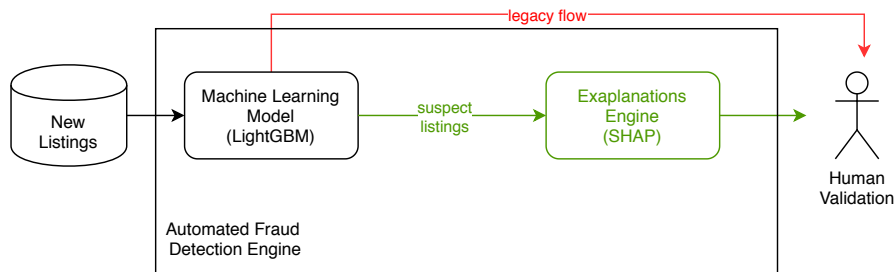


Figure 8.1: Final HousingAnywhere Fraud Detection Process

8.2 Academic Contributions

In this section, the academic contributions brought by this research and its experiments will be summarized.

The first general contribution is represented by the research itself. Specifically, as introduced in the beginning of this thesis, Automated Fraud Detection is a field which suffers from a shortage of publicly available researches, data and use cases caused by the sensitiveness of its applications. In this perspective, this research is aimed at standing as a comprehensive analysis of the whole Fraud Detection business process at HousingAnywhere, with a specific focus on the interaction between humans and machines.

Moreover, it is also important to point out the novelty of this type of research for this application domain. Specifically, this thesis represents one of the few publicly available researches about automated fraud detection and the first attempt to integrate interpretable machine learning with automated fraud detection in the online housing marketplaces.

8.2.1 Classification Contribution

For what concerns the machine learning classification task, the following experiments empirically demonstrated their positive contributions:

- **Objective Function for unbalanced data.** It has been shown that, even though objective functions have not been originally designed to address the problem of imbalanced classes, a cost function which assigns different importance to different classification errors can bring significant benefits both in classification performances and in mitigating the effect of building skewed classifiers.
- **Categorical Features Encoding.** It has been validated that traditional categorical features encoding techniques, such as One-Hot and Label Encoding, do not work as good as Fisher's grouping encoding with tree-based machine learning algorithm.
- **Missing Values Imputation.** Even though missing values are a well-known and debated topic in the field of Machine Learning, the importance of imputation for tree-based algorithm and for this specific application domain has been validated by the performance enhancement brought by such technique.
- **Bayesian Optimization.** Similar to the previous point, hyper-parameters tuning is a widely discussed and researched topic which development is not part of the scope of this thesis. However, it is important to notice that the performance improvements brought by this technique validate the effectiveness of automated hyper-parameters tuning in modern machine learning applications.

8.2.2 Interpretability Contributions

The major scientific contributions brought by this thesis relate to the field of interpretable machine learning. Specifically, this research represents in the first place a comparison of different explainability evaluations levels, namely application and human one. Furthermore, several state-of-the-art explainability techniques have been compared on a real task by a human-grounded evaluation.

On top of this, a novel approach to Adversarial explanations, namely EVADE has been designed, implemented, tested and compared to state-of-the-art techniques. Specifically, by the means of Genetic Algorithms, this technique generates synthetic instances which are able to fulfill the requirements of:

- Being as similar as possible to the original instance which prediction has to be explained.
- Being evaluated from the black box machine learning model as different as possible from the original instance.

The synthetic instance is then used to generate machine learning explanations by the means of comparison with the original sample.

From a performance point of view, this method reached the same accuracy of SHAP in human-grounded evaluation tasks, where SHAP has been chosen as the reference method for state-of-the-art machine learning interpretability as suggested in [25].

Alongside this novel approach to Adversarial explanations, three other interpretability methods have been implemented and their performances compared. As a consequence, the following scientific contributions can be derived:

- Machine Learning explanations improved in all cases the accuracy of humans validating machine learning predictions compared to a scenario where no interpretations to predictions were given. This result appears to be in contrast with what has been presented in [37], where explanations did not bring any statistically significant improvement in task's utility. However, given the different application domains it is also difficult to make a fair comparison with the information gathered so far. Therefore, it will be interesting, as a future work, to perform additional human-grounded evaluations, on different tasks, so that a comprehensive picture of the effectiveness of these methods can be assessed.
- SHAP and EVADE achieved the highest accuracy scores, validating in the first place the quality of the novel Adversarial approach and showing that SHAP is still a solid proxy for state-of-the-art explanations.
- Model-based explanations did not bring relevant advantages over model-agnostic ones, being significantly outperformed both by SHAP and EVADE. Therefore,

model-agnostic interpretability methods appear as more appealing thanks to their adaptability and performances.

- Explanations based on local surrogate models empirically proved to not be competitive in this use case, compared to the other explainability methods.

8.3 Business Contributions

In this section, the contributions mainly related to the business of HousingAnywhere and its fraud detection process will be given. At the beginning the impact of improving the classification algorithm is stated followed by the contribution brought by enhancing human-machine interactions with interpretable machine learning.

8.3.1 Model Performance Contributions

First, by looking at the classifier's performances improvements, previously presented in table 7.9, a clear step forward can be seen. Furthermore, it is also important to mention that the new model relies on a significantly lighter machine learning infrastructure than the legacy one, bringing the following benefits for HousingAnywhere:

- The new model requires less computational power given its simplified architecture, therefore allowing for a cheaper cloud infrastructure to support it.
- A lighter and faster machine learning model increases also the scalability capabilities of the fraud detection process, handling more traffic in a more resource-efficient way.
- A simplified machine learning infrastructure allows also to maintain and debug the model easier.

On the other hand, from a pure classification performance point of view, the final model results in:

- +23% in precision, approximately reducing the false positive rate of one legit item every ten.
- +0.2% in recall, meaning that the uplift in precision does not hit the recall capabilities of the model which has been maintained at the same level of the beginning of the research.
- +12.2% in F1-score, which means that the final classifier has a much better trade-off between precision and recall than the legacy one.
- +3.5% in Area Under the Precision-Recall Curve, showing a similar pattern that the one showed by F1-score.

In order to clearly assess the impact of the previously mentioned performance improvements on the business of HousingAnywhere it is necessary to introduce some business related knowledge. If a rate of 4000 new accommodations published on the platform each month and 5% of scam attempts are considered, with the legacy machine learning model, the Customer Solutions department of HousingAnywhere would have to check 320 listings ¹ per month with approximately only half of them representing real scam attempts, while the other half is composed by false positive alerts. Thanks to the new machine learning model precision boost, under the same rate of new listings and scam attempts, the amount of items that would have to be validated by humans drops to approximately 265 ², with 55 less listings to be checked per month. The main benefits brought by these improvements can be summarized as follows:

- **Time Saving**, the first intuitive benefit is that, given a reduced number of items that have to be checked, the overall required working time is reduced too. Specifically, assuming a realistic rate of 5 minutes required to check a listing on average, it is possible to approximately quantify the saved time in 5 working hours per month.
- **Errors Reduction**, assuming that the current human error rate is maintained, a fewer number of items to be checked would mean that also the total number of validation errors would be reduced. Specifically, assuming a realistic error rate of 10% (i.e. human wrongly marks as scam attempt a legit listing or vice-versa), the total number of validation errors would drop from 32 ³ to approximately 27 ⁴, with a 16% error rate reduction.

8.3.2 Explanations Contributions

Based on the technical results achieved through the implementation of explanations alongside the predicted probabilities, it has been decided to integrate explanations inside HousingAnywhere fraud detection process based on the interpretability method SHAP.

The decision of relying on SHAP is based on the results presented in section 7.2, where both SHAP and EVADE achieved the highest accuracy score among all methods but SHAP also proved to be more consistent throughout the different items categories.

The business benefits of integrating machine learning explanations can be summarized as follows:

- **Human Training**. By looking at the evaluation of explanations at Application-level and Human-level, presented in section 2.9.2, it is possible to derive an unexpected benefit of explanations. Specifically, given that with a 95% confidence it is

¹80% of real scam attempts are detected with a 50% precision, therefore $\frac{4000 \cdot 0.05 \cdot 0.8}{0.5} = 320$

²80% of real scam attempts are detected with a 60% precision, therefore $\frac{4000 \cdot 0.05 \cdot 0.8}{0.6} = 267$

³10% of 320 checked listings

⁴10% of 265 checked listings

possible to state that there is no statistically significant difference between the performance of the two different evaluation levels, it means that human validation is not affected by the presence of background knowledge of the task. From a business perspective, this means that the current training process for Customer Solutions employees dedicated to gaining expertise in the process of detecting possible scam attempts can be either reduced or avoided thanks to the implementation of machine learning explanations. The first intuitive benefit of removing the personnel training process is time saving and cost reduction. Moreover, the whole business process would become more scalable since there would be no barrier (i.e. no training needed) for humans in participating in HousingAnywhere fraud detection process.

- **Higher Accuracy.** As showed before, the human validation process benefits from explanations by achieving a much higher accuracy overall. Specifically, the improvements brought by the introduction of SHAP explanations are the followings:
 - The overall accuracy increased from 57% to 71%.
 - The accuracy over legit listings wrongly marked as scam attempts in the past, increased from 57% to 86%.
 - The accuracy over scam attempts wrongly validated as legit listings in the past, increased from 31% to 62%.
 - The accuracy over scam attempts correctly mask as scams in the past, increased from 50% to 63%.

Estimating the actual impact of these performance improvements on the business process is a challenging task due to the fact that the future distribution of listings over the different categories (i.e. R0H0, R0H1, R1H0, R1H1) is unknown and it indirectly depends also on the machine learning model. However, the results showed distributed improvements over different categories therefore ensuring that, independently from the distribution of listings over categories, it is very likely that explanations would bring a reduction over the number of errors committed by humans during the validation process.

8.4 Limitations & Future Work

On top of the contributions brought by this thesis, new technical challenges open up, either addressing current limitations or bringing further improvements.

Among these, the followings are the ones that are considered to be the most logical to be researched about next:

- Improving the machine learning classifier by the means of ensembling the current one with others of different nature (i.e. not tree-based algorithm). This scenario brings intuitive attention to the area of deep neural networks, which proved to achieve outstanding results in several classification tasks. Moreover, having a model-agnostic interpretability process will allow to easily modify the underlying machine learning infrastructure without it being affected.
- During the execution of the research it emerges that data in natural language, like item's description, which are currently not used from the machine learning model can contain valuable information. However, given that the application domain is based on an international platform, textual data can be present in several different languages and formats. Therefore, one of the next challenges which could bring positive results is to investigate the usage of such textual information through modern Natural Language Processing technique. Finally, such model can be ensembled with the current structure in order to improve the performances over the binary classification task.
- Due to the fact that interpretable machine learning is a recent and widely open research field, several further analyses can be made on explanations. Among these, the ones that stand out are:
 - Iterate the evaluation experiments by collecting more data and then investigate each method's characteristics. Specifically, it would be interesting to assesses both quantitative, such as variance in the results, and qualitative measures, such as the ones introduced in section 2.9.2 like comprehensibility, consistency and representativeness.
 - Investigate the computational performances of the different explainability methods which have been neglected from the scope of the research but plays a key role in real business application.
 - Iterate over EVADE both by improving the underlying structure, such as with non-linear fitness function or modelling and sampling from features distribution (instead of current uniform assumption), and by applying the technique to different interpretability tasks so that a comprehensive picture of algorithm's performances can be derived.

Bibliography

- [1] David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods. *CoRR*, abs/1806.08049, 2018. 28
- [2] Massimo Belloni. Scam detection in online housing offers: Model ensembling against dataset drifting, 2019. 33
- [3] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J. Christopher Westland. Data mining for credit card fraud: A comparative study. *Decis. Support Syst.*, 50(3):602–613, February 2011. 12, 14, 37
- [4] Richard J. Bolton and David J. H. Statistical fraud detection: A review. *Statistical Science*, 17:2002, 2002. 10, 33
- [5] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011. 13
- [6] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. 14, 26
- [7] Chao Chen and Leo Breiman. Using random forest to learn imbalanced data. *University of California, Berkeley*, 01 2004. 12
- [8] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017. 24
- [9] Yongsheng Fang and Jun li. A review of tournament selection in genetic programming. pages 181–192, 10 2010. 19
- [10] Tom Fawcett and Foster Provost. Combining data mining and machine learning for effective user profiling. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, page 8–13. AAAI Press, 1996. 15, 37
- [11] Walter D. Fisher. On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53(284):789–798, 1958. 46

- [12] D. Fradkin. Clustering inside classes improves performance of linear classifiers. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pages 439–442, Nov 2008. 15
- [13] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, Sep. 2009. 12, 14, 20, 22
- [14] David W. Hosmer and Stanley Lemeshow. *Applied logistic regression*. John Wiley and Sons, 2000. 14
- [15] Fortune Business Insights. Fraud prevention and prevention market, 2020. 2, 10
- [16] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449, October 2002. 12, 14
- [17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017. 18, 36, 46
- [18] Yufeng Kou, Chang-Tien Lu, S. Sirwongwattana, and Yo-Ping Huang. Survey of fraud detection techniques. volume 2, pages 749 – 754 Vol.2, 02 2004. 10, 14, 33, 37
- [19] Stephan Kovach and W.V. Ruggiero. Online banking fraud detection based on local and global behavior. In: *Proc. of the Fifth International Conference on Digital Society*, 01 2011. 14, 37
- [20] X. Liu and Z. Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 970–974, Dec 2006. 14
- [21] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *CoRR*, abs/1802.03888, 2018. 30
- [22] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. Credit card fraud detection using bayesian and neural networks. 08 2002. 33
- [23] K. F. Man, K. S. Tang, and S. Kwong. Genetic algorithms: concepts and applications [in engineering design]. *IEEE Transactions on Industrial Electronics*, 43(5):519–534, 1996. 19
- [24] Kate McCarthy, Bibi Zabar, and Gary Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? In *Proceedings of the 1st International Workshop on Utility-Based Data Mining*, UBDM '05, page 69–77, New York, NY, USA, 2005. Association for Computing Machinery. 14

-
- [25] Christoph Molnar. *Interpretable Machine Learning*. 2019. <https://christophm.github.io/interpretable-ml-book/>. 24, 25, 26, 27, 29, 31, 70
- [26] Youngsam Park, Damon McCoy, and Elaine Shi. Understanding craigslist rental scams. In Jens Grossklags and Bart Preneel, editors, *Financial Cryptography and Data Security*, pages 3–21, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg. 33
- [27] Johan Perols. Financial Statement Fraud Detection: An Analysis of Statistical and Machine Learning Algorithms. *AUDITING: A Journal of Practice*, 30(2):19–50, 05 2011. 32
- [28] Clifton Phua, Vincent C. S. Lee, Kate Smith-Miles, and Ross W. Gayler. A comprehensive survey of data mining-based fraud detection research. *CoRR*, abs/1009.6119, 2010. 10, 33
- [29] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Mach. Learn.*, 42(3):203–231, March 2001. 10
- [30] Shini Renjith. Detection of fraudulent sellers in online marketplaces using support vector machine approach. *International Journal of Engineering Trends and Technology (IJETT)*, 57:48–53, 03 2018. 33
- [31] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016. 27
- [32] Marko Robnik-Šikonja and Marko Bohanec. *Perturbation-Based Explanations of Prediction Models*, pages 159–175. 2018. 25
- [33] L.S. Shapley. A value for n-person games, 1953. 28
- [34] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. 14
- [35] Brijesh Verma and Ashfaqur Rahman. Cluster-oriented ensemble classifier: Impact of multicluster characterization on ensemble classifier learning. *IEEE Trans. Knowl. Data Eng.*, 24(4):605–618, 2012. 15
- [36] Rashmi Korlakai Vinayak and Ran Gilad-Bachrach. DART: Dropouts meet Multiple Additive Regression Trees. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 489–497, San Diego, California, USA, 09–12 May 2015. PMLR. 47
- [37] Hilde J. P. Weerts, Werner van Ipenburg, and Mykola Pechenizkiy. A human-grounded evaluation of shap for alert processing, 2019. 34, 70

- [38] Masoumeh Zareapoor, Seeja K.R., and Afshar Alam. Analysis on credit card fraud detection techniques: Based on certain design criteria. *International Journal of Computer Applications*, 52:35–42, 08 2012. 33
- [39] Masoumeh Zareapoor and Pourya Shamsolmoali. Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia Computer Science*, 48:679 – 685, 2015. International Conference on Computer, Communication and Convergence (ICCC 2015). 33
- [40] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, Jan 2006. 14