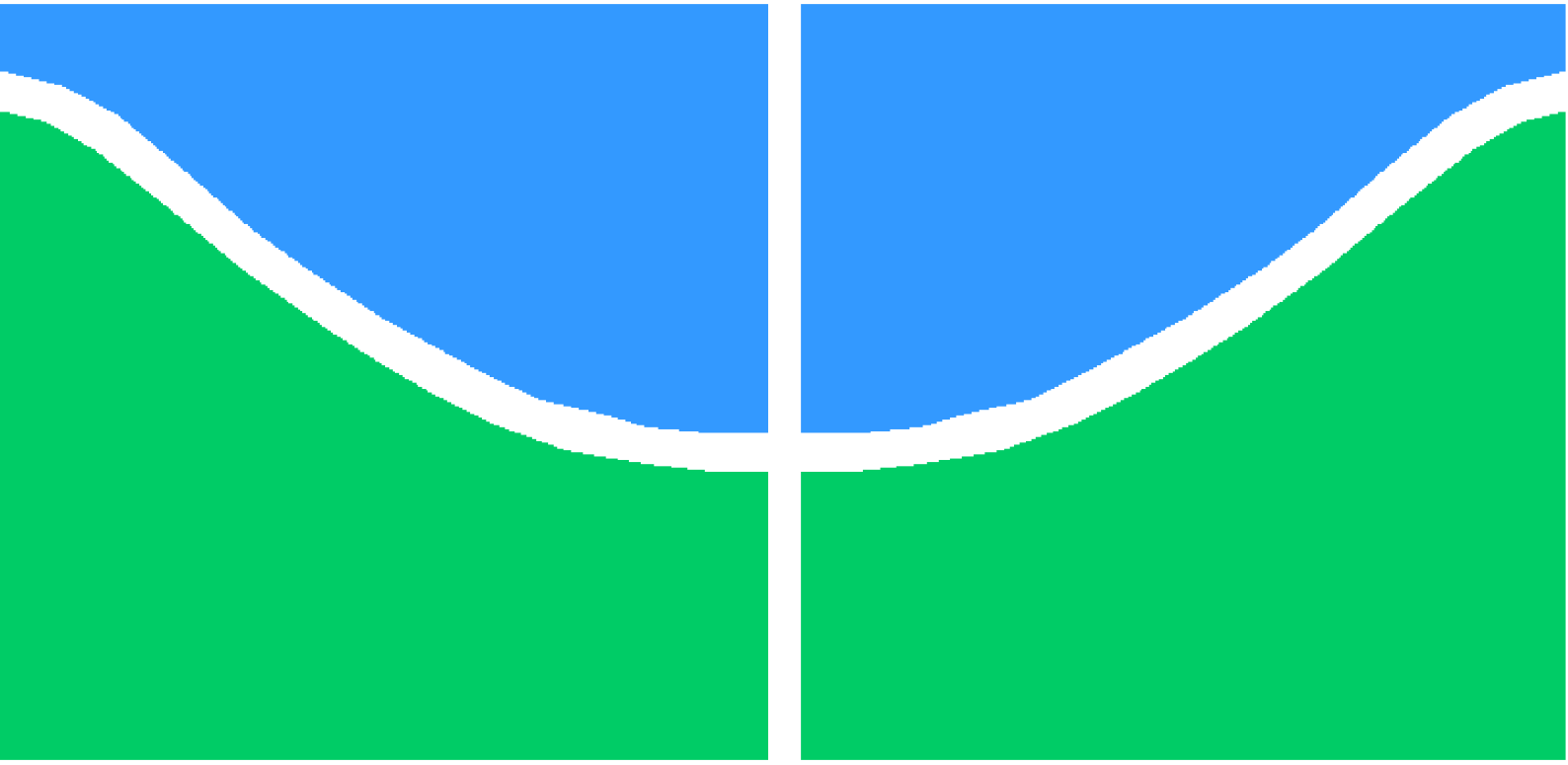

University of Brasília - UnB
Faculty UnB Gama - FGA
Software Engineering

Feature Selection using SHAP: An Explainable AI approach

Author: Miguel Pimentel da Silva
Supervisor: Nilton Correia da Silva, PhD

Brasília, DF
2021

Miguel Pimentel da Silva

Feature Selection using SHAP: An Explainable AI approach

In partial fulfillment of the requirements for the degree of Bachelor of Software Engineering.

University of Brasília - UnB

Faculty UnB Gama - FGA

Supervisor: Nilton Correia da Silva, PhD

Brasília, DF

2021

Miguel Pimentel da Silva
Feature Selection using SHAP: An Explainable AI approach/ Miguel Pimentel da Silva. – Brasília, DF, 2021-
63 p. : il. color.) ; 30 cm.

Supervisor: Nilton Correia da Silva, PhD

Undergraduate Thesis – University of Brasília - UnB
Faculty UnB Gama - FGA , 2021.

1. SHAP. 2. Feature Selection. I. Nilton Correia da Silva, PhD. II. University of Brasília. III. Faculty UnB Gama. IV. Feature Selection using SHAP: An Explainable AI approach

CDU 02:141:005.6

Miguel Pimentel da Silva

Feature Selection using SHAP: An Explainable AI approach

In partial fulfillment of the requirements for the degree of Bachelor of Software Engineering.

Approved. Brasília, DF, 23 de Maio de 2021:

Nilton Correia da Silva, PhD
Supervisor

Fabricio Ataides Braz, PhD
Examiner

Jerônimo da Silva Avelar Filho, Mastering
Examiner

Brasília, DF
2021

Acknowledgements

First of all, this moment is more than a dream. In this incredible and challenging journey, many people were very special to me.

I thank my mother, Carolina, for all the support throughout my life. Despite that, I also thank my grandparents, uncles, and aunts for all their advice. I thank my friends for being with me on this journey, giving me strength when things seemed complicated.

I also thank the UnB for being a reference in Software Engineering. I am especially grateful to Professor Nilton, who supported me throughout this final challenge.

Certainly, graduation was one of the biggest challenges in my life, without words to describe how relevant and important this is to me. Thanks to everyone that contributes to me.

Abstract

In the last decade, Artificial Intelligence (AI) appears to be in many different areas in human lives. Many times those AI models are based on complex algorithms and neural networks, also called as black boxes. In recent years, tools have emerged with the objective of explaining the operation of black boxes, i.e, SHAP. Studies have shown that these tools can be used as a feature selection tool, which can improve the accuracy of the models and reduce the computational costs of model training. The main objective of this work is to understand how much explainability tools can assist in the feature selection process from three perspectives: Performance, Training Time; and Accuracy. Those metrics were evaluated based on two practical experiments. The first one using the Cancer Breast Dataset and the second one using the Credit Card Fraud dataset. Each experiment was carried out for the following models: Random Forest, XGBoost, Catboot, and LightGBM. As result, we were able to conclude that SHAP, in addition to bringing explanability, can bring performance gains in a machine learning model.

Key-words: Explainable AI. SHAP. Feature Selection.

List of Figures

List of Tables

List of abbreviations and acronyms

AI	Artificial Intelligence
CNN	Convulational Neural Network
FN	False Negatives
FP	False Positive
LIME	Local Interpretable Agnostic Model
ML	Machine Learning
UnB	University of Brasilia
ULB	Université Libre de Bruxelles
SHAP	SHapley Additive exPlanations
TCC2	Undergraduate Thesis 2
TP	True Positives
TN	True Negative
XAI	Explainable Artificial Intelligence

Contents

Introduction

In the last decade, Artificial Intelligence (AI) appears to be in many different areas in human lives. Today, this technology is used in different scenarios, such as: self-driving cars; insurance claim; analyses of tumours; credit card claim; and many others.

Thus, many times those AI models are based on complex algorithms and neural networks, also called as black boxes. Black box is a term used to refer to a system for which we can only observe the inputs and outputs, but not the internal workings.

The lack of transparency of black boxes create questions about trust of those models and applications in real life. In that context, Explainable Artificial Inteligence (XAI) provides insights about how to understand the outputs provided by those models.

In recent years, tools have emerged with the objective of explaining the operation of black boxes, i.e, SHapley Additive exPlanations (SHAP).

Moreover, studies have shown that these tools can be used as a feature selection tool, which can improve the accuracy of the models and reduce the computational costs of model training.

Therefore, the main objective of this work is to understand how SHAP, an explainability tool can assist in the feature selection process from three perspectives: Training Time; Storage; and Performance. In this work, SHAP was used because it is a model agnostic tool based on concepts and theories that are widespread in game theory, Math Research Field.

Those metrics were evaluated based on two practical experiments. The first one using the Cancer Breast Dataset and the second one using the Credit Card Fraud dataset. Cade experiment was carried out for the following models: Random Forest, XGBoost, Catboot, and LightGBM. The methods and materials section provides more details about the experiments.

0.1 Problem

Deciding which features to select in machine learning models can be a tricky task, since it can affect different perspectives, not restricting only the understanding of a specific domain. In this way, feature engineering has two main objectives:

- Preparing the proper input dataset
- Improving the performance of machine learning models.

This step is within the data preparation, second, this stage can include up to 80 percent of the efforts required in the development of a machine learning process (SHEARER, 2000).

The main idea of this work is to understand the use of Shapley values as a feature selection tool, using studies case using SHAP as the base.

0.2 Research Goals

The main objective of this work is to understand how SHAP, an explainability tool, can assist in the feature selection process from the following perspectives: Performance; training time; and storage.

In order to make these results more comprehensive, different datasets and models were used. The next subsection is going to present in detail the specific research objectives.

0.3 Specific Objectives

As mentioned before, to understand how SHAP work as a feature selection tool, this work focused on the following perspectives:

- **Training time**: According to Marcílio and Eler (MARCÍLIO; ELER, 2020), one of the biggest benefits of the feature selection is the training time in which the model is executed in the dataset. In this sense, the training time without feature selection was compared to that with feature selection in order to understand the impact of the feature selection on the training time.
- **Storage**: The storage concerns with the size of the dataset. In this case, the dataset size is often loaded into memory, that is, large datasets may require a lot of computational power.
- **Performance**: According to Bellmann (BELLMAN; KALABA, 1959), for some models of feature selection it can improve the degree of assertiveness of the model, that is, Accuracy, Precision, Recall, and F1 Score.

1 Background

1.1 Black Boxes

Recent advances of AI allow us to realize the provided achievements of accurate machine learning models (SAMEK; WIEGAND; MÜLLER, 2017). Major keys of this development were earlier improvements in support vector machines and more recent improvements in deep learning methodology (LECUN et al., 2012).

These results allowed many advances in different knowledge areas. A research developed by Samek, Wiegand and Muller (SAMEK; WIEGAND; MÜLLER, 2017), defines this improves in AI as: *"With the availability of large databases and recent improvements in deep learning methodology, the performance of AI systems is reaching or even exceeding the human level on an increasing number of complex tasks. Impressive examples of this development can be found in domains such as image classification, sentiment analysis, speech understanding or strategic game playing."*

Regardless of this advance, models are usually complex and have comprehension gaps, impairing people's understanding. For this reason, these models are known as Black boxes and its representation includes an input which is submitted into a complex and hierarchical level of interaction, followed by an output (Figure 1).

Black box models provide an opaque and non-intuitive accuracy value, impairing the understanding (GUNNING, 2017). This raises questions to the final user, as follows:

1. Why did you do that?
2. When can I trust you?
3. When do you succeed?
4. When do you fail?
5. How do I correct an error?

The next subsections are going to approach some methods and XAI tools.

1.2 Explainable Artificial Intelligence

Explainable AI (XAI) is an area of artificial intelligence research related to the ability in which humans can understand AI solutions. It contrasts with the concept of "black

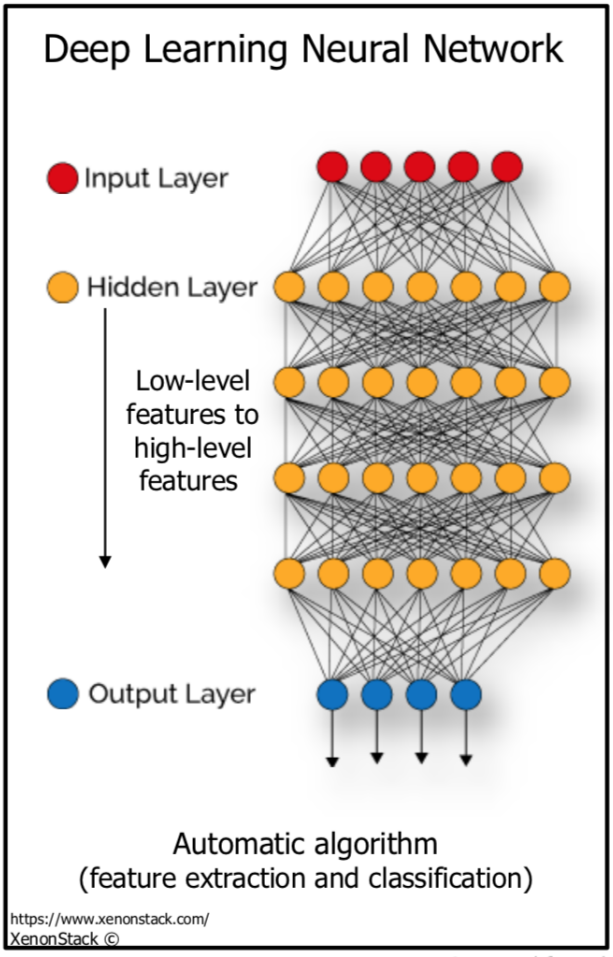

Figure 1 – E.g Black Box - Source: XenonStack

box", presented in the last subsection. Thus, Explainability is required in many systems where prediction should not fail: Transportation; Finances, Security; Legal; Medicine and Military. (SAMEK; WIEGAND; MÜLLER, 2017).

In this sense, Herlocker, Jonathan, Konstan, Joseph, Riedl, and John in their research introduce how relevant trust is for the final user. That research presented that when the user knows how the movie's recommendation works (How the prediction model works) they accept more that advice, as consequence, they watch more movies (HERLOCKER; KONSTAN; RIEDL, 2000).

Interpretability and Accuracy are considered one of the major factors of a successful predictive model (CHOI et al., 2016). However, many applications present the impossibility of understanding and validating the decisions of an AI model, and that is a strong penalty (SAMEK; WIEGAND; MÜLLER, 2017).

A research proposed by Samek, Wiegand and Muller (SAMEK; WIEGAND; MÜLLER, 2017), defines this trust as: *"If the users do not trust a model or a prediction, they will not use it. It is important to differentiate between two different (but related) definitions*

of trust: (1) trusting a prediction, i.e. whether a user trusts an individual prediction sufficiently to take some action based on it, and (2) trusting a model, i.e. whether the user trusts a model to behave in reasonable ways if deployed."

Nevertheless, for Samek, explainable artificial intelligence presents other relevant characteristics (SAMEK; WIEGAND; MÜLLER, 2017):

- **Verification of the system:** Provide information that justify the output of a model. As mentioned before a user should not trust in a black box by default.
- **Improvement of the System:** Understand its weakness. For instance, it is more difficult to analyze black box models than interpretable models. Detecting biases on data sets and models making it easier to understand the results provided from a specific input.
- **Learning from the system:** Many AI Systems are trained with millions of examples, for this reason it can recognize patterns intangible for people. However, When using an explainable AI system it is easier to access information about the model, thus acquiring new insights about the model.
- **Compliance to legislation:** Recently, countries have increased attention in legislation that approach AI Systems, for example autonomous car laws. Many times, those models are black boxes, which makes it quite complicated to explain their results. Withal, it's difficult to assign responsibility for wrong predictions. According to Martins (MARTINS,), cites the implications of Brazilian law 13.709 / 2018 on AI models. One of the implications is that the IA models must be transparent, understandable and explainable

As mentioned before, there was a common belief that a trade off must be done in favour of interpretability or accuracy (CHOI et al., 2016). the image 2 below represents this concept, where represents models and its accuracy related with interpretability.

However, recent studies have shown that these tools can improve the accuracy of the models, reduce the computational costs of model training and most of all add interpretability to those systems.

This work covers post-hoc explainability techniques, that is models do not meet any of the criteria imposed to declare them transparent, a separate method must be devised and applied to the model to explain its decisions. These techniques aim to present how an already developed model produces its predictions for any given input. Model-agnostic techniques for post-hoc explainability could be classified:

- **Explanation by simplification:** These techniques consist of fragmenting the explanation of a model or prediction into parts or fragments. Among the most known

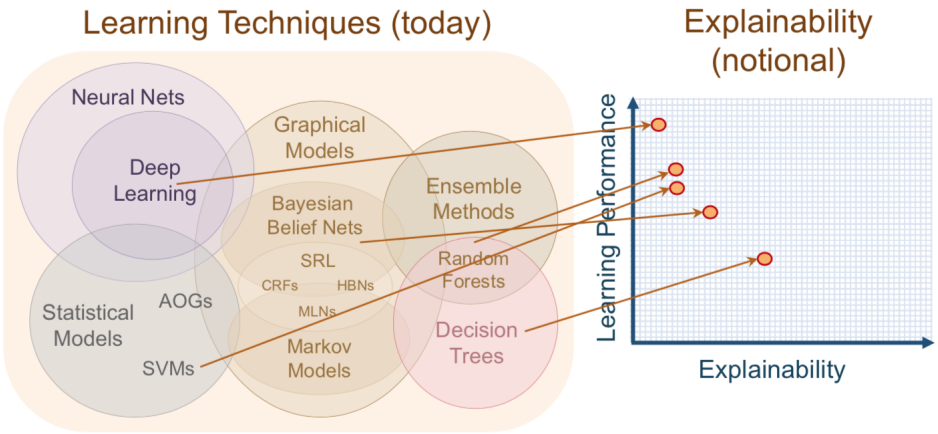

Figure 2 – XAI Initial Concept - Source: DARPA XAI

contributions to this approach we encounter the technique of Local Interpretable Model-Agnostic Explanations (LIME) (RIBEIRO; SINGH; GUESTRIN, 2016a) and all its variations (MISHRA; STURM; DIXON, 2017), (RIBEIRO; SINGH; GUESTRIN, 2016b). LIME builds locally linear models around the predictions of an opaque model to explain it.

- **Feature relevance explanation:** These techniques aim to describe the functioning of an opaque model by measuring or ranking the influence, relevance or importance each feature has in the prediction output by the model to be explained. One of the most known contributions to this approach we encounter the SHAP technique of (LUNDBERG; LEE, 2017).

- **Visual explanation techniques:** These techniques consist in presenting a portfolio of visualization techniques to help in the explanation of a black-box ML. That are important works in this research area such as (CORTEZ; EMBRECHTS, 2011) and (CORTEZ; EMBRECHTS, 2013).

1.2.1 SHAP

SHAP is a tool that allows understanding the output of any machine learning model based on a game theoretic approach. This technique connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions. Shap is defined as post hoc and model agnostic XAI, that is, it can be applied to any model and its applied post modelling training step (MUELLER et al., 2019)

In this work, SHAP was used as a feature selection tool due to the following reasons (SHAP,):

- SHAP is a model agnostic tool
- SHAP is Based on mathematical concepts, i.e, Shapley values
- SHAP is a Global approach, it's not exclusively a local approach, i.e, LIME
- SHAP is post-hoc technique

1.2.1.1 General Idea

Suppose you had to explain a machine learning model that calculates the value of an apartment. There are several attributes that can set your price, for example, covered parking, swimming pool, pets friendly, size, location, etc.

For a linear model it is not such a complicated task, as each resource has a weight that defines its importance in the final prediction. However, for other models, it can be a complex task.

There is a solution based on the game theor. In this context, the Shapley value is a method for allocating payments to players depending on their contribution to the total payment. Players cooperate in a coalition and receive a certain profit from that cooperation. In machine learning, it is understood that players are the features of a model, and the gain is the final prediction of the model. Thus, The Shapley value represents the average marginal contribution of a resource value across all possible coalitions.

For example, to calculate the weight of a feature F, the value of that feature is replaced by a value from another instance of the model and all possible possibilities are taken in order to compare the original prediction with this new prediction. The avarage value between the random value and the original represents the weight of this features to the final prediction.

This calculation is repeated for all possible coalitions. Since this theory is based on comparisons, the computation time increases exponentially with the number of resources.

1.2.1.2 Definition

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j \tag{1.1}$$

where g is the explanation model, $z' \in \{0,1\}^M$ is the coalition vector, M is the maximum coalition size and $\phi_j \in R$ is the feature attribution for a feature j, the Shapley values

1.2.1.3 Library

SHAP is an open Source project maintained by Scott Lundberg, a microsoft researcher, and it's available under MIT License. Today, this tool is available as a framework and could be installed using PIP (SHAP,). The project repo has accomplished some results in XAI community, such as:

- More **than twelve thousand** of stars on Github;
- More than **120 contributors**;
- More than 45 releases;

The image 3 below presents the SHAP open source repository:

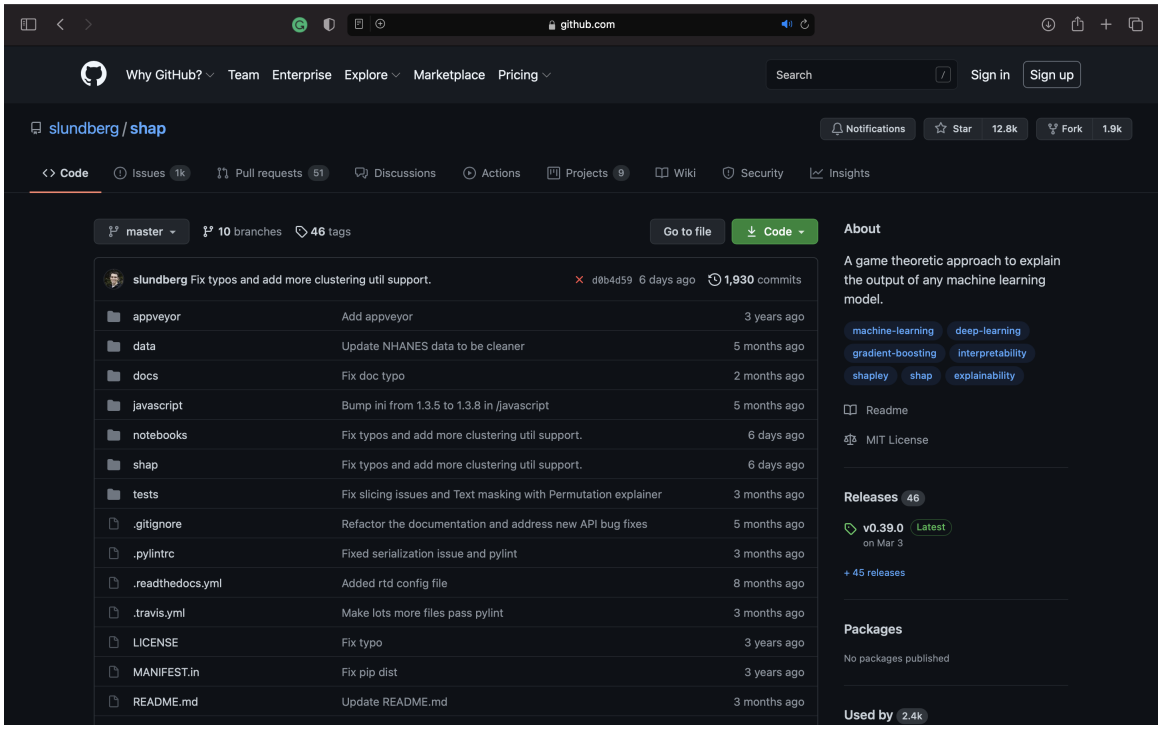

Figure 3 – SHAP Repository - Source: SHAP

In addition, the library provides a clear explanation of the relevance of each feature in the final prediction. In the image below 4, the pink part defines the one with the greatest impact, while the blue part presents the one with the least impact. The example was taken from the Boston Houses dataset, a very popular dataset.

1.3 Feature Selection

In the paper, A Survey on Feature Selection methods, the authors relate the dimensionality of the dataset to redundancy problems (CHANDRASHEKAR; SAHIN, 2014).

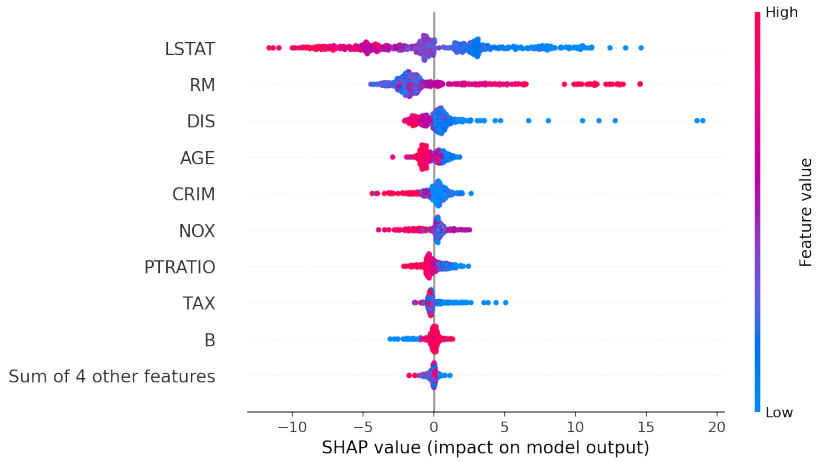

Figure 4 – Example SHAP Output - Source: SHAP

This problem occurs because each data point appears to be equidistant from each other (MARCÍLIO; ELER, 2020). One of the ways to solve or reduce this problem is through the selection of data, that is, remove data that does not directly impact the machine learning solution.

This concept of selecting features that are relevant to an AI model is called feature selection. The features selection algorithms can be classified as:

- **Filter:** This consists of selecting a subset of the most important features of a dataset. This procedure is performed as a pre-processing step. (CHANDRASHEKAR; SAHIN, 2014)
- **Wrapper:** This method consists of removing and / or replacing a set of features that least impact the predictive power of a model, i.e, the model's output. (BELLMAN, 2015)
- **Embedded:** his method occurs during training, so they are specific to each type of model. E.g, one of the most common models is the removal of neurons in which the weights approach zero. (BOLÓN-CANEDO et al., 2014)

In summary, feature selection provides great benefits, such as:

- **Reduces Overfitting:** By decreasing the amount of redundant data decreasing the likelihood of making decisions based on noise.
- **Improves Accuracy:** By decreasing the amount of misleading data it could benefit the accuracy of a model.
- **Reduces Training Time:** By decreasing the amount of data points the training time is shorter, since the algorithm's complexity is decreased.

Recent studies demonstrated that XAI tools can be used as a feature selection mechanism. In this context, SHAP obtained better results as a feature selection tool than other quite common techniques, such as Mutual Information, RFE and ANOVA (MARCÍLIO; ELER, 2020). As mentioned before, this work aims to apply XAI technique as a feature selection tool, measuring and analyzing in different contexts.

1.4 Models

As mentioned in the introduction, this work aims to understand feature selection using SHAP through different perspectives. One of the most important perspectives is to understand how it works by multiple models. Thus, this section is going to present each used model.

1.4.1 Catboost

CatBoost is an algorithm for gradient boosting on decision trees. It is developed by Yandex researchers and engineers, and is used for search, recommendation systems, personal assistant, self-driving cars, weather prediction and many other tasks at Yandex and in other companies, etc. It is in open-source. (PROKHORENKOVA et al., 2017)

1.4.2 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes. (SCIKIT-LEARN, 2021) . Although the Random Forest model presents tools that allow us to understand the relevance of each feature, this model was used as a way to understand how TREE SHAP works, an explainer made for this type of model (SHAP,).

1.4.3 LightGBM

LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient. One of the main characteristics is the support to distributed of parallel. It was purposed by microsft. (LIGHTGBM,)

1.4.4 XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework (XGBOOST,).

2 Material and Methods

2.1 Methods

In order to meet the purposed objectives. A sequence of key activities was created. As already mentioned, SHAP is a post-hoc technique, that is, it is always applied after training a model.

In the experiments of this work, in the initial interaction the model is executed with all the features and the SHAP is applied on it in order to identify the relevance of each feature.

After this first interaction, the next ones consist of performing the training of the model with a set of features in their order of priority according to the data obtained by SHAP in the initial stage. This process happens as follows: In the first interaction, the model is executed with only the most relevant feature, in the second interaction, the model is executed with the first and second most relevant features, and so on. The image 5 represents represents the experiments activities. Throughout this section, each of these activities will be presented in detail.

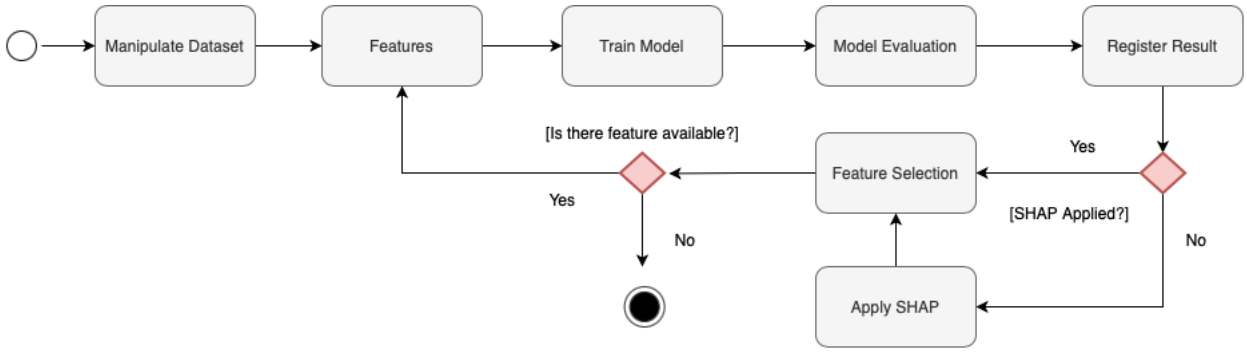

Figure 5 – Experiments Methodology

2.1.1 Manipulate Dataset - Data Preparation

In many cases, most of the data used in data mining are collected and used for other purposes, and consequently, they are of some kind of refinement before being modeled. In this activity, two very important tasks are performed. (SHEARER, 2000)

The first is known as data cleaning, this task consists in make changes, perhaps tracking down sources to make specific data corrections, excluding some cases or items of data, or replacing some items of data with default values or replacements selected by a more sophisticated modeling technique.

The second is known as data formatting, it consists in to manipulate the data type to another format that fits better to modeling purposes.

2.1.2 Select Features

One of the most important aspects of a model is selecting the features that fit better into a particular domain. As previously presented, all features were selected in order to achieve their importance in the model's output using Shapley values.

In the first interaction, no feature removal is performed, in relation to the data preparation step. However, from the second interaction, the first and second most relevant features are selected, and so on.

2.1.3 Train Model

In this step, the model is trained with the selected features, the dataset was split in 70 percent to training and 30 percent to data validation. As presented before, each experiment is based on a specific model. The next subsections are going to present the models approached in this research and their configurations.

2.1.3.1 XGBoost

In classification experiments was used the XGBClassifier and its default values, in the package version 1.4.1 for python (XGBOOST, 2021).

The table 1 presents the value for each XGBoost property.

Table 1 – XGBoost Classifier - Model's Properties Values

Property	**Value**
base_score	0.5
booster	gbtree
colsample_bynode.	1
colsample_bytree	1
gamma	0
learning_rate	0.1
max_delta_step	0
max_depth	3
min_child_weight	1
n_estimators	100
n_jobs	1
objective	multi:softprob
random_state	0
reg_lambda	1
scale_pos_weight	1
Verbosity	1

2.1.3.2 Random Forest

In classification experiments was used the RandomForestClassifier and its defaults values, in the version 0.24.1 (SCIKIT-LEARN, 2021).

The table 2 below presents those values.

Table 2 – Random Forest Classifier - Model's Properties Values

Property	**Value**
n_estimators	gbdt
criterion	squared_error
min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0
max_features	auto
min_impurity_decrease	0.
bootstrap	True
oob_score	False
verbose	0
warm_start	False
ccp_alpha	0.0

2.1.3.3 LightGBM

In classification experiments was used the LGBMClassifier and its defaults values, in the version 3.2.1. (MICROSOFT, 2021).

The table 3 below presents those values.

2.1.3.4 CatBoost

In classification experiments was used the CatBoostClassifier and its defaults values, in the version 0.25.1.

In Catboost, the defaults values are None for every property, in this case. Please check the documentation for more details (CATBOOST, 2021).

2.1.4 Model Evaluation

In this step, the results found in the training of a model (M) with resources f are measured. As was presented at the beginning of the work, this work aims to measure different perspectives of the training process of a model, not just looking at the precision or accuracy of the models.

For classification problems, the concept of confusion matrix is used as a basis for different metrics that aim to identify different perspectives of the assertiveness of a machine learning model.

Table 3 – LightGBM Classifier - Model's Properties Values

Property	**Value**
boosting_type	gbdt
num_leaves	31
max_depth	-1
learning_rate	0.1
n_estimators	100
subsample_for_bin	200000
min_split_gain	0.0
min_child_weight	0.001
min_child_samples	20
subsample	1.0
ubsample_freq	0
colsample_bytree	1.0
reg_alpha	0.0
reg_lambda	0.0
n_jobs	-1
silent	True
importance_type	split

In this project, we are based on the following metrics: Storage (bytes); Training time (ms); Precision (%), Precision (%), Reccall (%), and F1 Score (%). In the metrics section, we'll go into detail for each one.

2.1.5 Apply Shap

As previously presented, the shap is a post-hoc technique, that is, it is performed after the training phase of a model. From the output vector of the trained model it is possible to use the SHAP explainers, that is, an interface that allows to calculate the shap values in an optimized way. From these shap values, we can interpret the data in different ways (graphs and manipulations).

In this experiment, a csv file was created with the shap values of each feature in descending order. From the shap values, we can understand the importance of each feature. This step that provides the sequence of the applied feature selection. In addition, this phase is only performed in the first interaction of the experiment, the other interactions are based on the csv with the priorities created in the first interaction.

2.1.6 Feature Selection

As mentioned in the beginning of this section, the features are selected and grouped according to shap values.

In this step, the table is with features relevance (shap values) and the features are grouped in a interactive way.

In the first interaction, the model is executed with only the most relevant feature, in the second interaction, the model is executed with the first and second most relevant features, and so on. Mathematically, this could be understood as:

1. First Interaction: $I_1 = f_1$
2. Second Interaction: $I_2 = f_1, f_3$
3. Third Interaction: $I_3 = f_1, f_2, f_3$
4. N Interaction: $I_n = f_1, f_2, f_3, ..., f_n$

2.2 Metrics

2.2.1 Performance

As previously mentioned, machine learning is used in different applications in our daily lives, some of them in scenarios where an unexpected result can generate major impacts. In this context, it is essential to understand performance metrics that seek to explain the assertiveness of a model from different perspectives.

In this work, classification problems were used, which means that the output is the segmentation into two or more classes. In this type of problem, the following metrics are used: Accuracy; Precision; Recall; and F1-Score. These metrics are based on the concepts of True Positives (TP), True Negative (TN), False Positive (FP), and False Negatives (FN) which can be understood as:

- True Positives (TP): True positives are the cases when the current class of the data point was 1 (True) and the predicted is also 1 (True).
- True Negatives (TN): True negatives are the cases when the current class of the data point was 0 (False) and the predicted is also 0 (False)
- False Positives (FP): False positives are the cases when the current class of the data point was 0 (False) and the predicted is 1 (True). False is because the model has predicted incorrectly and positive because the class predicted was a positive one (1)/
- False Negatives (FN): False negatives are the cases when the current class of the data point was 1 (True) and the predicted is 0 (False). False is because the model has predicted incorrectly and negative because the class predicted was a negative one (0).

Thus, the Confusion Matrix is used as the basis for these metrics, since its values represent TP; TN, FP, FN. Thus, from the confusion matrix, we were able to understand these metrics.

In this section, we are going yo explain how these metrics were calculated and what they mean.

2.2.1.1 Accuracy

In classification problems, accuracy is the number of correct predictions made by the model over all kinds of predictions made. it is calculated as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2.2.1.2 Precision

Precision is the fraction of the correctly classified instances from the total classified instances. Precision is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

2.2.1.3 Recall

Recall is the fraction of the correctly classified instances from the total classified instances. The following equation represents recall:

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

2.2.1.4 F1 Score

F1 score represents both Precision(P) and Recall(R). Its aim is to balance between two other metrics: recall and precision. The F1-Score is calculated as:

$$F1 - Score = 2 \cdot \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{2.4}$$

2.2.2 Training Time

According to Marcílio and Eler (MARCÍLIO; ELER, 2020), one of the biggest benefits of the feature selection is the training time in which the model is executed in the dataset. In this way, feature selection was performed in different conditions in order to understand how they work according with the amount of features. Those measure were obtained by the time library in python.

2.2.3 Storage

The storage concerns about the size of the dataset. In this case, the dataset's size often is loaded into memory, that is, large datasets may require a lot of computational power. In this case, this metric was measured using the python standard library. The filter was applied according to the proposed methodology.

2.3 Materials

2.3.1 Datasets

In order to understand SHAP as feature selection tool, it was used different datasets in the experiment. This subsection is going to present each dataset and its properties.

2.3.1.1 Breast Cancer Dataset

The Breast Cancer dataset is very common in the AI community. This dataset is associated with health, more specifically breast cancer, the features were computed from digitalized images of of a fine needle aspirate (FNA) of a breast mass. They represent characteristics of the cell nuclei present in the image (DUA; GRAFF, 2017). This dataset is one of the most used to test and apply concepts. It's available in scikit learn and also known as toy dataset.

The table 4 presents some dataset's properties.

Table 4 – Cancer Breast Dataset Properties

Data Set Characteristics	Multivariate
Attribute Characteristics	Real
Associated Tasks	Classification
Number of Instances	569
Number of Attributes	32
Missing Values	No
Area	Life
Date Donated	01-11-1195
Number of Web Hits	1485620

2.3.1.1.1 Breast Cancer - Dataset Classes

The Breast Cancer Dataset is used to detect breast cancer, classifying tumors between malign and benign. In this case, from the total of 569 instances, 357 were classified as benign tumors and 212 as malignant tumors (breast cancer).

The following table 5 presents the dataset segmentation according to classes (Benign and Malign):

Table 5 – Cancer Breast Dataset - Number of Instances x Classes

Class	Number of Instances
Benign	357
Malign	212

2.3.1.2 Credit Card Fraud Dataset

The Credit Card Fraud dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset is related with finance, more specifically credit card fraud, it has 31 features, most of them are opaque, that is, their names and meaning are not readable due to confidentiality issues (MLG, 2013). This dataset has been collected during a research collaboration of Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.

The table 6 presents some dataset's properties.

Table 6 – Credit Card Fraud Dataset Properties

Data Set Characteristics	Multivariate
Attribute Characteristics	Real
Associated Tasks	Classification
Number of Instances	284807
Number of Attributes	31
Missing Values	No
Area	Finance
Date Donated	12-09-2013
Number of Web Hits	N.A.

2.3.1.2.1 Credit Card Fraud - Dataset Classes

The Credit Fraud Dataset is used to detect credit card fraud. As already mentioned, this dataset is based on real data, which means, only a small sample of the dataset's instances are expected to be classified as fraud. In this case, from the total of 284807, 492 were classified as fraud and 284315 as normal transactions (non-fraud).

The following table 7 presents the dataset segmentation according to classes (Fraud and Non-Fraud):

2.3.2 Hardware

The experiments were performed with a laptop with the following configurations:

Table 7 – Cancer Breast Dataset - Number of Instances x Classes

Class	Number of Instances
Fraud	492
Non Fraud	284315

- **Processor:** Intel Core i5 (10th generation), 4 cores and 2.0 GHz, Turbo Boost up to 3.8 GHz, with 6 MB shared L3 cache
- **RAM memory:** 16GB LPDDR4X integrated memory with 3733 MHz
- **Graphics Chip**: Intel Iris Plus Graphics
- **Storage:** 512 GB SSD
- **Operating System:** macOS Big Sur 11.2.3

3 Results

In this section, we will cover the experiments developed according to the methodology and techniques presented in the previous section. In this way, this chapter can be divided into two subsections. The first subsection presents the results for the experiment with the breast cancer dataset. The second subsection addresses the experiment carried out in the credit card fraud detection dataset.

3.1 Experiment 1 - Breast Cancer

3.1.1 Feature Relevance

One of the most interesting results of the experiment was the understanding of the relevance of each feature in each model. It was possible to see that each model works in a different way and this is reflected in different weights for each of the features.

Some features, for example, *concave points*, *perimeter_worst*, *concave points_mean* are very relevant in all models, alternating between the top 5 most relevant. However, others such as *area_se* proved to be relevant for a specific model, in this case, the fourth more relevant in the XGBoost, but the thirteenth in the Catboost model.

The Appendix A presents in detail the value and weight of each of the features for each model.

3.1.2 Performance

In order to evaluate the model's performance, it was used the following metrics: Accuracy; Precision; Recall; and f1 score. For this experiment, the accuracy values followed the other performance metrics, for example, In the Catboost model, the highest accuracy was obtained with 29 features, but also with this feature number, the highest precision, recall, and F1 score were obtained.

Using the Catboost model it was possible to obtain the highest performance with 29 features, presenting accuracy, precision, recall, and f1 score of approximately 99.41. Using one feature as the base, 85.9% were obtained for the same metrics. Using two features as the base, the model presented a great performance gain, about 6% for all performance metrics. Between 5 and 15 features, the results obtained varied between 94 to 98%.

In the LightGBM Model, it was possible to obtain the highest performance with 29 features, the metrics of accuracy, precision, recall, and f1 score presented a result

of approximately 97.6%. Using only one feature as a base, approximately 91.8% was obtained for the same metrics. From 5 features, the model found many different results to performance metrics.

For Random Forest it was possible to obtain the highest performance with 25 features, presenting accuracy, precision, recall, and F1 Score with approximately 98.2%. With only one feature, 88.8% were obtained for the same metrics. For two features the model showed a great performance gain, about 5%. From 4 to 25 features, the results varied considerably for performance metrics, between 92% and 98%.

The XGBoost Model, on the other hand, did not need many features to achieve its greatest metrics of accuracy, precision, recall, and f1 score. Only the 7 most relevant features were needed to achieve accuracy, precision, recall, and f1-score of approximately 98.8%. Using only the most relevant feature, the model obtained 89.4%

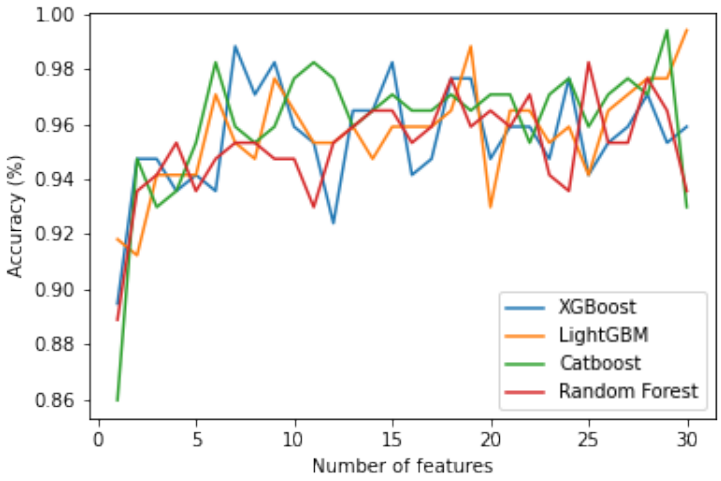

Figure 6 – Cancer Breast Experiment - Accuracy x Number of Features

3.1.3 Training Time

The breast cancer dataset is not a very extensive dataset, as it has just over 500 instances and 30 features. Thus, the training time of the model for the hardware used is relatively fast. There is a hypothesis that due to the short training time on all models, the results are more sensitive to any change in computer performance, for example, thread management, queues, etc. For XGBoost, LightGBM and Random Forest the training time values remained constant for different amounts of features. The Catboost presented a interesting result, even with a small dataset, the time was longer than the other models and grew progressively with the increase in features.

The image 10 below present those results.

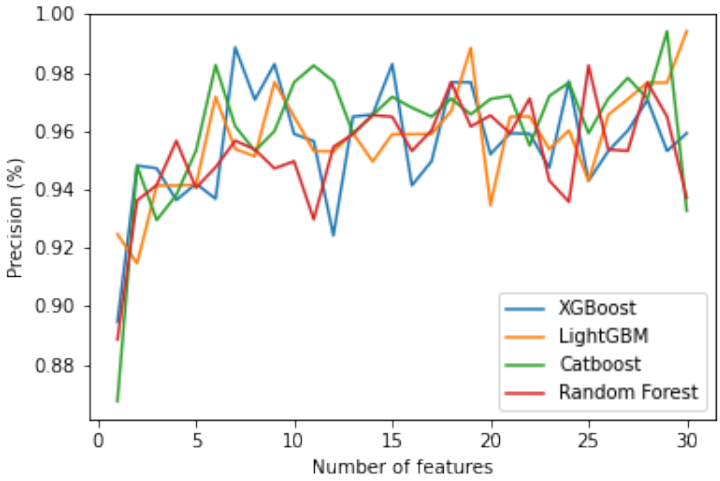

Figure 7 – Cancer Breast Experiment - Precision x Number of Features

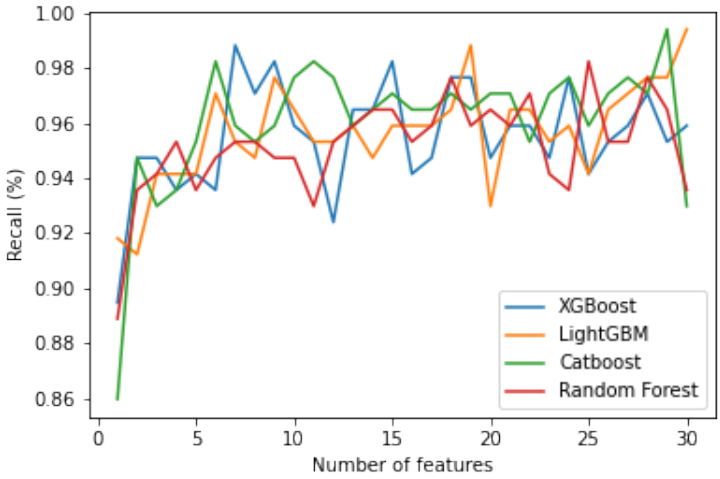

Figure 8 – Cancer Breast Experiment - Recall x Number of Features

3.1.4 Storage

As all instances of the model have the same number of features, it was expected that the storage would exhibit a linear behavior, which was possible to report. For all models, storage has shown an identical behavior, since this topic is much more related to the number of features than a specific property of any of the models.

For the dataset with only one feature selected, 11509 bytes were stored, a big difference in relation to the dataset with all the features that presented 128323 bytes. The image 11 presents those results.

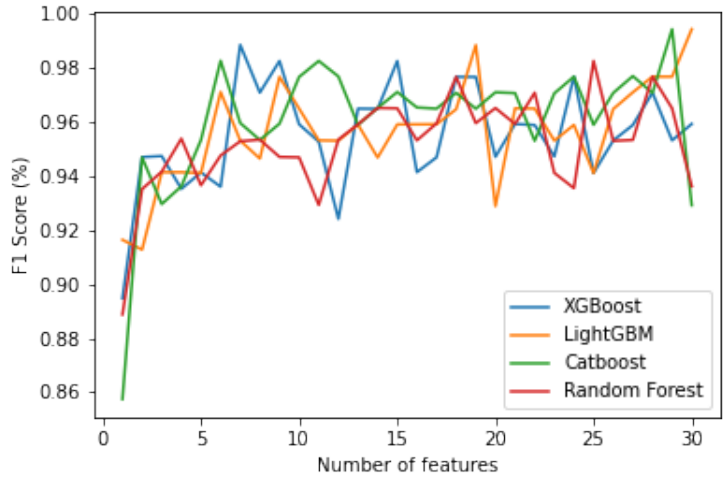

Figure 9 – Cancer Breast Experiment - F1 Score x Number of Features

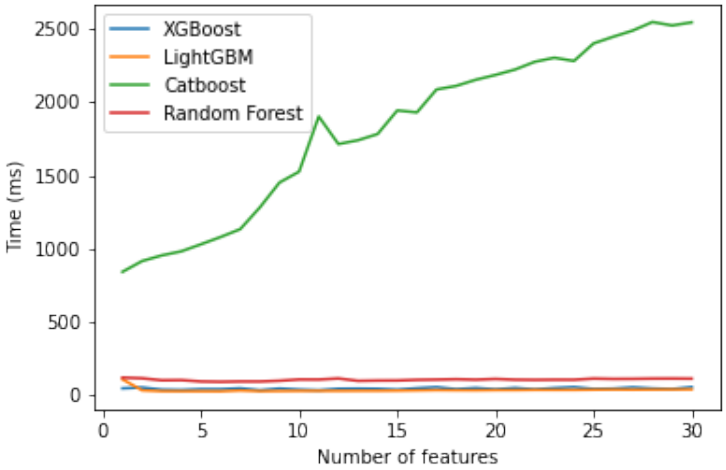

Figure 10 – Cancer Breast Experiment - Training Time x Number of Features

3.2 Experiment 2 - Credit Card Fraud

3.2.1 Feature Relevance

One of the objectives of this research is to understand the relevance of features in different models. In this sense, the order of priority varied across all models. For some models there was not so much variation, for example, the most relevant feature for the Random Forest, XGBoost, and Catboost models was the same, V14. In other features, such as the V1 feature, the relevance for different models varied considerably. In this case, V1 was the most relevant for the LigthGBM model, but it was the twentieth most relevant for XGBoost.

From this we can understand how the models work, making it easier to understand

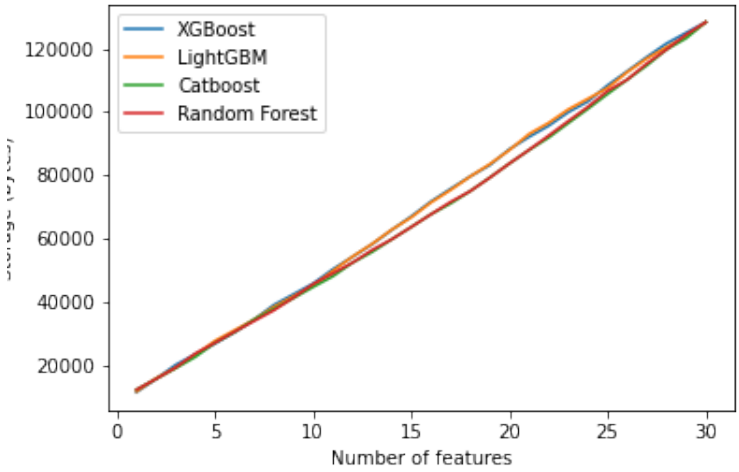

Figure 11 – Cancer Breast Experiment - Storage x Number of Features

what is taken into account in the final prediction.

In appendix B, there are details on the value and weight of each of the features for each model.

3.2.2 Performance

In order to evaluate the performance of the model, the following metrics were used: Accuracy; Precision; Recall; and f1 score. In this case, they varied in the same proportion, presenting very similar results between the metrics, this is clear when analyzing the charts. Each model obtained the highest values of its metrics with a different set and number of features.

For Catboost, using only the most relevant feature as a basis, approximately 99.8% accuracy, precision, recall and f1 score were obtained. Although there were not many performance gains with the increase of the feature, the highest values of accuracy, precision, recall, and f1 score were obtained using the 19 most relevant features, in this scenario approximately 99.9%.

Using the LigthGBM Model, it was possible to find an interesting result, the metrics of recall, accuracy, f1 score, and precision did not vary with the same proportion. For this model, the highest accuracy and recall were obtained using only the most relevant feature, approximately 99.8% were obtained for these metrics. However, 5 features were necessary to obtain the highest values of f1-score and precision, in this case, approximately 99.85% was obtained.

The Random Forest model obtained the highest accuracy using the 8 most relevant features, for this scenario 99.95% was obtained for the metrics of accuracy, precision,

recall and f1-score. Using only the most relevant feature, 99.85% was obtained for the same metric.

The XGBoost, obtained the highest accuracy using the 30 most relevant features, a little different in relation to the other models. In this scenario, approximately 99.9% were obtained for these metrics. Using only the most relevant feature, 99.8% were obtained for the same metrics.

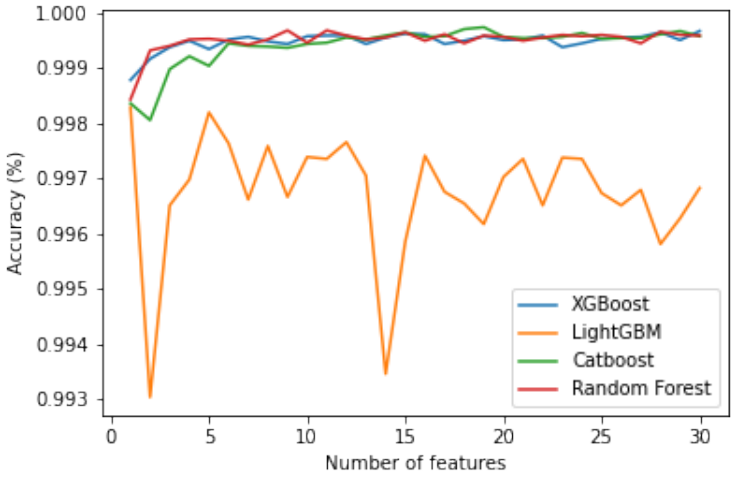

Figure 12 – Credit Card Fraud Experiment - Accuracy x Number of Features

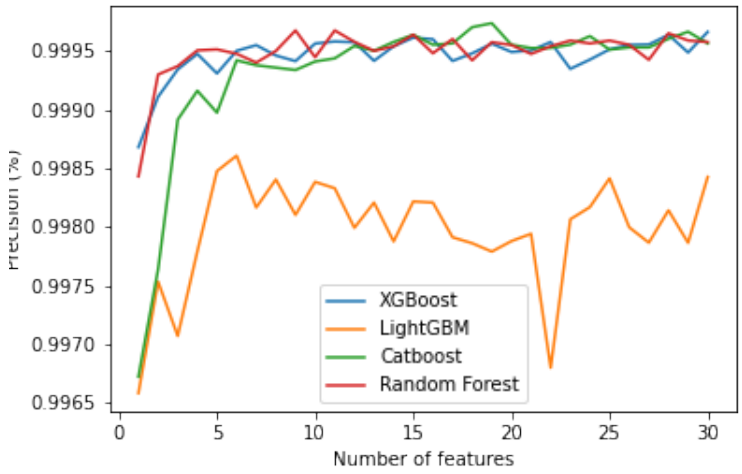

Figure 13 – Credit Card Fraud Experiment - Precision x Number of Features

3.2.3 Training Time

The training time presented specific characteristics for each of the models. For XGboost there was a small progressive increase in the training time according to the

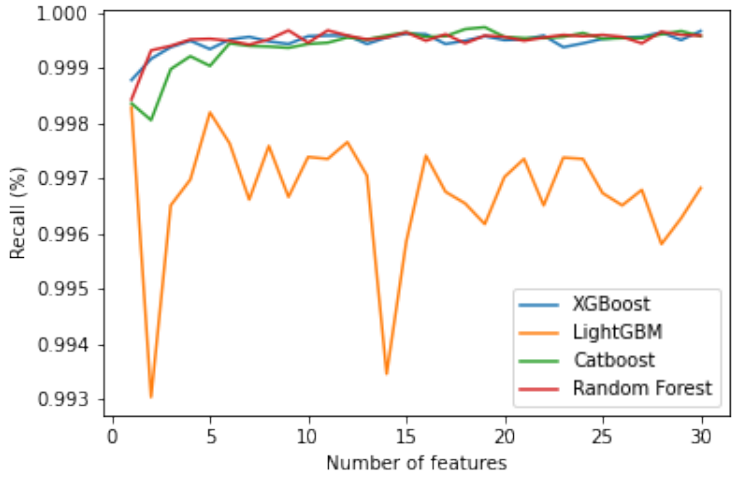

Figure 14 – Credit Card Fraud Experiment - Recall x Number of Features

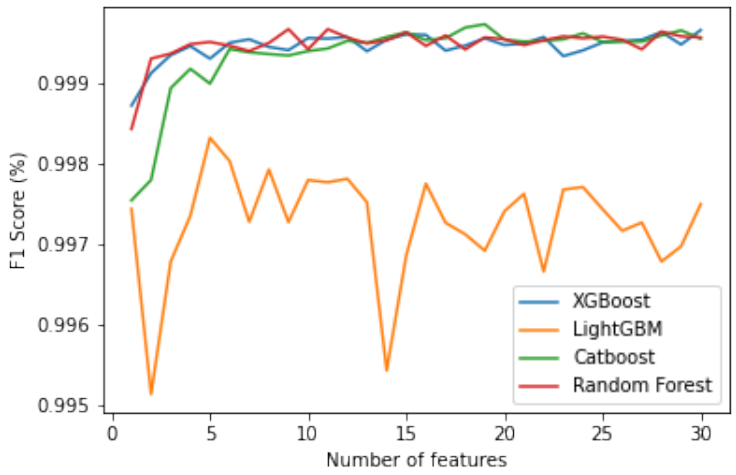

Figure 15 – Credit Card Fraud Experiment - F1 Score x Number of Features

number of features. For the ligthGBM model, the training time remained practically constant, regardless of the number of features. On the other hand, at Random Forest there was a considerable and progressive increase according to the number of features. In Catboost, the model was not very sensitive in relation to the variation in the number of features.

The following image 16 shows the relationship between the training time and the number of features:

3.2.4 Storage

The Credit Card Fraud dataset is a dataset with thousands of instances, and all of them presenting the same number of features. The increase in the storage of the dataset

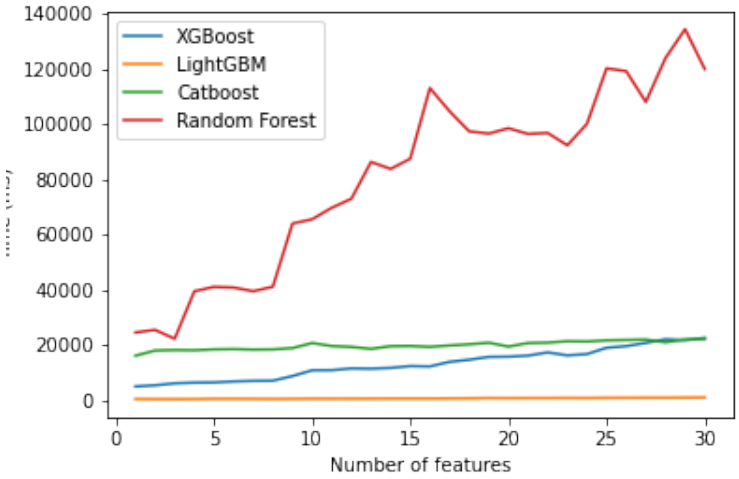

Figure 16 – Credit Card Fraud Experiment - Training Time x Number of Features

was supposed to vary linearly, which has been proven. However, it is interesting to see the variation. For a feature, the data set had approximately 7.2 mb of storage, almost 150 mb more than the dataset with all the features, a huge difference.

The following image 17 shows the relationship between storage and the number of features:

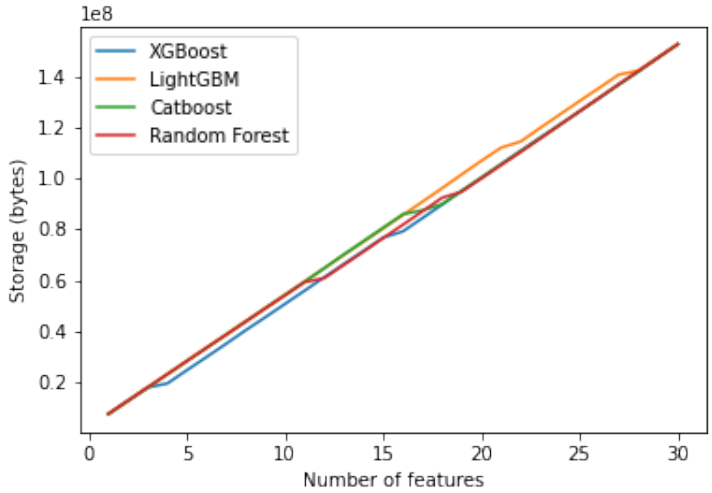

Figure 17 – Cancer Breast Experiment - Storage x Number of Features

3.3 Summary

From the experiments carried out it was possible to find some interesting points about the use of SHAP as a feature selection tool.

The first point concerns the coherence of the features, in the two experiments, although each model works differently, and as a consequence of changing the order of importance of the features, it can be observed that the group of the most relevant features has changed little.

Another interesting aspect was that in some models, the relevance of a feature or group of features was so high for the model that the highest performance metrics were obtained with only a part of the dataset, it means, the amount of data is not necessarily implies performance gains. In experiments with the Credit Card Fraud dataset, the dataset with hundreds of thousands of instances, it became clear that it is possible to save large hardware processing from the selection of features using SHAP.

Initially, one of the hypotheses of the work was that decreasing the number of features would imply a shorter training time. However, this hypothesis has not been proven. For the Credit Card Fraud dataset, this hypothesis was confirmed for the .Random Forest model.

Finally, one of the biggest gains, is to understand the functioning of the model, that is, the explicability. From the understanding of how a model works, it is possible to identify points for improvement, validate with specialists in a knowledge area the functioning, and even assist in homologation processes. In the same vein, As already mentioned, stricter rules regarding the use of AI in different contexts have now come into force, the use of tools such as SHAP, which is based on scientific theories, can greatly assist in the approval and auditing of Machine Learning models.

4 Conclusion

In the last few years, some XAI tools have emerged in order to explain Machine Learning models. Recently, some studies have started using these tools for different purposes.

In this work, we use SHAP as a feature selection tool. As a result, using one or a small group of features it was possible to obtain excellent results that associate performance (accuracy, precision, recall and F1 Score) with storage. In other words, in some contexts we are able to understand what is really important for the model, and it is not necessary to use the entire dataset. For some models, using few features can mean great savings in computational resources.

Despite having each model use its own approach, almost all models presented a similar group of more important features, presenting coherence on the part of SHAP.

Nevertheless, we can still use the shap as an explanability tool, its real purpose. Thus, it is possible to use it as a tool that allows us to explain how the model works for stakeholders of an ML project, company, or even in audits.

Therefore, we were able to conclude that SHAP, in addition to bringing explanability, can bring performance gains in a machine learning model.

4.0.1 Future Work

Today, there are several methodologies that are used in mining and developing machine learning models. However, there is no methodology that uses XAI tools as the basis, such as SHAP.

Using this work as a basis, in future works, studies about how we can development machine learning models based on SHAP could be performed, since SHAP can be used in different ways. Thus, it's possible to understand how SHAP can bring gains in the development of AI models, even proposing a methodology centered on XAI.

References

BELLMAN, R.; KALABA, R. On adaptive control processes. *IRE Transactions on Automatic Control*, IEEE, v. 4, n. 2, p. 1–9, 1959. Cited on page 20.

BELLMAN, R. E. *Adaptive control processes: a guided tour*. [S.l.]: Princeton university press, 2015. Cited on page 27.

BOLÓN-CANEDO, V. et al. A review of microarray datasets and applied feature selection methods. *Information Sciences*, Elsevier, v. 282, p. 111–135, 2014. Cited on page 27.

CATBOOST. *catboost.CatBoostClassifier*. [S.l.], 2021. 0.25.1. Cited on page 31.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. *Computers & Electrical Engineering*, Elsevier, v. 40, n. 1, p. 16–28, 2014. Cited 2 times on pages 26 and 27.

CHOI, E. et al. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *arXiv preprint arXiv:1608.05745*, 2016. Cited 2 times on pages 22 and 23.

CORTEZ, P.; EMBRECHTS, M. J. Opening black box data mining models using sensitivity analysis. In: IEEE. *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. [S.l.], 2011. p. 341–348. Cited on page 24.

CORTEZ, P.; EMBRECHTS, M. J. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, Elsevier, v. 225, p. 1–17, 2013. Cited on page 24.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Available at: <http://archive.ics.uci.edu/ml>. Cited on page 35.

GUNNING, D. *Explainable Artificial Intelligence (XAI). DARPA*. 2017. Cited on page 21.

HERLOCKER, J. L.; KONSTAN, J. A.; RIEDL, J. Explaining collaborative filtering recommendations. In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. [S.l.: s.n.], 2000. p. 241–250. Cited on page 22.

LECUN, Y. A. et al. Efficient backprop. In: *Neural networks: Tricks of the trade*. [S.l.]: Springer, 2012. p. 9–48. Cited on page 21.

LIGHTGBM. <https://lightgbm.readthedocs.io/en/latest/>. Accessed: 2021-05-31. Cited on page 28.

LUNDBERG, S.; LEE, S.-I. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017. Cited on page 24.

MARCÍLIO, W. E.; ELER, D. M. From explanations to feature selection: assessing shap values as feature selection mechanism. In: IEEE. *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2020. p. 340–347. Cited 4 times on pages 20, 27, 28, and 34.

MARTINS, A. M. Lgpd–lgpd, ia, decisões automatizadas impactos e perspectivas. Cited on page 23.

MICROSOFT. *lightgbm.LGBMClassifier*. [S.l.], 2021. 3.2.1. Cited on page 31.

MISHRA, S.; STURM, B. L.; DIXON, S. Local interpretable model-agnostic explanations for music content analysis. In: *ISMIR*. [S.l.: s.n.], 2017. p. 537–543. Cited on page 24.

MLG, U. *Credit card fraud detection dataset*. 2013. Available at: <https://www.kaggle.com/mlg-ulb/creditcardfraud>. Cited on page 36.

MUELLER, S. T. et al. Explanation in human-ai systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable ai. *arXiv preprint arXiv:1902.01876*, 2019. Cited on page 24.

PROKHORENKOVA, L. et al. Catboost: unbiased boosting with categorical features. *arXiv preprint arXiv:1706.09516*, 2017. Cited on page 28.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. " 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*. [S.l.: s.n.], 2016. p. 1135–1144. Cited on page 24.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. Nothing else matters: model-agnostic explanations by identifying prediction invariance. *arXiv preprint arXiv:1611.05817*, 2016. Cited on page 24.

SAMEK, W.; WIEGAND, T.; MÜLLER, K.-R. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017. Cited 3 times on pages 21, 22, and 23.

SCIKIT-LEARN. *sklearn.ensemble.RandomForestClassifier*. [S.l.], 2021. 0.24.1. Cited 2 times on pages 28 and 31.

SHAP. <https://github.com/slundberg/shap>. Accessed: 2021-05-31. Cited 3 times on pages 24, 26, and 28.

SHEARER, C. The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, THE DATA WAREHOUSE INSTITUTE, v. 5, n. 4, p. 13–22, 2000. Cited 2 times on pages 20 and 29.

XGBOOST. <https://xgboost.readthedocs.io/en/latest/>. Accessed: 2021-05-31. Cited on page 28.

XGBOOST. *xgboost.XGBClassifier*. [S.l.], 2021. 1.4.1. Cited on page 30.

Appendix

APPENDIX A – Cancer Breast Experiment

A.1 Random Forest Tree

Table 8 – Feature Relevance - SHAP Values - Cancer's Breast Dataset - Random Forest

	Unnamed: 0	feature_name	importance_value
0	27	concave points_worst	26.566186
1	22	perimeter_worst	24.922345
2	23	area_worst	22.120229
3	7	concave points_mean	19.327713
4	20	radius_worst	18.165788
5	3	area_mean	11.098553
6	2	perimeter_mean	10.244782
7	0	radius_mean	9.617430
8	26	concavity_worst	8.732641
9	6	concavity_mean	7.201295
10	13	area_se	7.070465
11	21	texture_worst	4.364284
12	25	compactness_worst	4.340140
13	1	texture_mean	3.977649
14	10	radius_se	3.863066
15	24	smoothness_worst	3.463269
16	28	symmetry_worst	2.941480
17	12	perimeter_se	2.334336
18	5	compactness_mean	2.078270
19	16	concavity_se	2.043887
20	4	smoothness_mean	1.266888
21	18	symmetry_se	0.935642
22	15	compactness_se	0.832027
23	9	fractal_dimension_mean	0.771670
24	14	smoothness_se	0.718923
25	11	texture_se	0.695754
26	19	fractal_dimension_se	0.677671
27	29	fractal_dimension_worst	0.642576
28	17	concave points_se	0.561824
29	8	symmetry_mean	0.518114

A.2 XGBoost

Table 9 – Feature Relevance - SHAP Values - Cancer's Breast Dataset - XGBoost

	Unnamed: 0	feature_name	importance_value
0	22	perimeter_worst	1.101188
1	27	concave points_worst	0.974443
2	7	concave points_mean	0.939440
3	13	area_se	0.833870
4	21	texture_worst	0.817257
5	23	area_worst	0.723298
6	26	concavity_worst	0.694400
7	15	compactness_se	0.486687
8	20	radius_worst	0.448749
9	1	texture_mean	0.382153
10	18	symmetry_se	0.367454
11	28	symmetry_worst	0.366362
12	24	smoothness_worst	0.360021
13	16	concavity_se	0.272285
14	4	smoothness_mean	0.199590
15	17	concave points_se	0.194344
16	25	compactness_worst	0.150426
17	8	symmetry_mean	0.102595
18	11	texture_se	0.056790
19	14	smoothness_se	0.049512
20	10	radius_se	0.048950
21	3	area_mean	0.036825
22	6	concavity_mean	0.034707
23	12	perimeter_se	0.034166
24	19	fractal_dimension_se	0.029044
25	29	fractal_dimension_worst	0.028442
26	9	fractal_dimension_mean	0.000000
27	5	compactness_mean	0.000000
28	2	perimeter_mean	0.000000
29	0	radius_mean	0.000000

A.3 LightGBM

Table 10 – Feature Relevance - SHAP Values - Cancer's Breast Dataset - LightGBM

	Unnamed: 0	feature_name	importance_value
0	23	area_worst	693.576415
1	27	concave points_worst	665.362136
2	22	perimeter_worst	609.189908
3	26	concavity_worst	572.558205
4	7	concave points_mean	386.386723
5	21	texture_worst	244.280990
6	20	radius_worst	180.097799
7	24	smoothness_worst	162.563704
8	1	texture_mean	145.043682
9	13	area_se	131.548509
10	15	compactness_se	116.738775
11	9	fractal_dimension_mean	54.882406
12	25	compactness_worst	52.690447
13	18	symmetry_se	46.297387
14	10	radius_se	44.728723
15	17	concave points_se	40.917433
16	8	symmetry_mean	35.757749
17	6	concavity_mean	32.645883
18	28	symmetry_worst	27.367866
19	29	fractal_dimension_worst	24.207242
20	19	fractal_dimension_se	18.185502
21	3	area_mean	15.923575
22	5	compactness_mean	15.462100
23	0	radius_mean	10.687702
24	2	perimeter_mean	9.954175
25	14	smoothness_se	7.776548
26	4	smoothness_mean	5.529139
27	11	texture_se	4.130595
28	12	perimeter_se	3.564711
29	16	concavity_se	0.208021

A.4 Catboost

Table 11 – Feature Relevance - SHAP Values - Cancer's Breast Dataset - Catboost

	Unnamed: 0	feature_name	importance_value
0	27	concave points_worst	0.740244
1	22	perimeter_worst	0.669129
2	20	radius_worst	0.658063
3	23	area_worst	0.610135
4	7	concave points_mean	0.590276
5	21	texture_worst	0.352823
6	26	concavity_worst	0.337185
7	1	texture_mean	0.307245
8	13	area_se	0.303341
9	3	area_mean	0.285331
10	0	radius_mean	0.255972
11	6	concavity_mean	0.243867
12	2	perimeter_mean	0.211958
13	24	smoothness_worst	0.182216
14	28	symmetry_worst	0.139773
15	25	compactness_worst	0.114145
16	12	perimeter_se	0.110660
17	10	radius_se	0.107240
18	4	smoothness_mean	0.063675
19	29	fractal_dimension_worst	0.059282
20	5	compactness_mean	0.055388
21	8	symmetry_mean	0.048001
22	15	compactness_se	0.047225
23	16	concavity_se	0.046898
24	17	concave points_se	0.039364
25	9	fractal_dimension_mean	0.028292
26	18	symmetry_se	0.028099
27	14	smoothness_se	0.027675
28	11	texture_se	0.022075
29	19	fractal_dimension_se	0.021249

APPENDIX B – Credit Card Fraud Experiment

B.1 Random Forest Tree

Table 12 – Feature Relevance - SHAP Values - Credit Card Fraud Dataset - Random Forest

	Unnamed: 0	feature_name	importance_value
0	14	V14	178.540740
1	17	V17	138.214021
2	12	V12	116.735307
3	4	V4	80.975493
4	10	V10	73.854558
5	1	V1	69.409760
6	11	V11	64.872522
7	2	V2	53.458866
8	16	V16	47.702313
9	3	V3	31.618076
10	7	V7	28.742319
11	29	Amount	26.028112
12	19	V19	23.369754
13	9	V9	22.087690
14	20	V20	19.178026
15	18	V18	18.984821
16	21	V21	18.598131
17	26	V26	17.896018
18	0	Time	16.489262
19	13	V13	15.010573
20	6	V6	14.088459
21	15	V15	13.587823
22	8	V8	13.001889
23	23	V23	12.677521
24	28	V28	10.965329
25	25	V25	10.779406
26	27	V27	10.497737
27	5	V5	10.375139
28	24	V24	10.231140
29	22	V22	9.036302

B.2 XGBoost

Table 13 – Feature Relevance - SHAP Values - Credit Card Fraud Dataset - XGBoost

	Unnamed: 0	feature_name	importance_value
0	14	V14	1.114405
1	4	V4	0.966266
2	12	V12	0.555716
3	29	Amount	0.387997
4	10	V10	0.311767
5	11	V11	0.299570
6	19	V19	0.271064
7	20	V20	0.247851
8	3	V3	0.223351
9	22	V22	0.223130
10	16	V16	0.222293
11	8	V8	0.215675
12	7	V7	0.210741
13	5	V5	0.209147
14	13	V13	0.205142
15	0	Time	0.182314
16	25	V25	0.181492
17	26	V26	0.172901
18	28	V28	0.165343
19	2	V2	0.157619
20	21	V21	0.148308
21	24	V24	0.132269
22	6	V6	0.126853
23	1	V1	0.123816
24	15	V15	0.120010
25	9	V9	0.105844
26	23	V23	0.103325
27	18	V18	0.092280
28	17	V17	0.089707
29	27	V27	0.088435

B.3 LightGBM

Table 14 – Feature Relevance - SHAP Values - Credit Card Fraud Dataset - LightGBM

	Unnamed: 0	feature_name	importance_value
0	1	V1	7.434911e+09
1	12	V12	1.810319e+09
2	26	V26	7.792542e+08
3	14	V14	7.395059e+08
4	16	V16	7.129183e+08
5	20	V20	7.027402e+08
6	5	V5	6.751558e+08
7	15	V15	6.458956e+08
8	3	V3	6.069165e+08
9	7	V7	5.344268e+08
10	13	V13	5.257495e+08
11	22	V22	5.226141e+08
12	10	V10	5.100766e+08
13	18	V18	5.091284e+08
14	9	V9	5.089845e+08
15	8	V8	3.873433e+08
16	17	V17	3.493250e+08
17	21	V21	3.475735e+08
18	28	V28	3.198095e+08
19	23	V23	2.826331e+08
20	4	V4	2.263139e+08
21	0	Time	2.166006e+08
22	27	V27	1.724407e+08
23	24	V24	1.308633e+08
24	25	V25	1.101273e+08
25	11	V11	8.126164e+07
26	19	V19	6.437499e+07
27	29	Amount	5.708457e+07
28	2	V2	5.237114e+07
29	6	V6	3.519177e+07

B.4 Catboost

Table 15 – Feature Relevance - SHAP Values - Credit Card Fraud Dataset - Catboost

	Unnamed: 0	feature_name	importance_value
0	1	V1	0.610696
1	14	V14	0.575734
2	4	V4	0.512862
3	8	V8	0.416447
4	13	V13	0.370410
5	10	V10	0.368462
6	11	V11	0.336685
7	19	V19	0.328266
8	25	V25	0.280274
9	12	V12	0.275883
10	18	V18	0.246081
11	26	V26	0.216451
12	28	V28	0.199943
13	6	V6	0.197276
14	24	V24	0.185735
15	17	V17	0.182155
16	29	Amount	0.180362
17	0	Time	0.177253
18	22	V22	0.171437
19	7	V7	0.138809
20	16	V16	0.136536
21	3	V3	0.135688
22	2	V2	0.132959
23	20	V20	0.118468
24	15	V15	0.114412
25	23	V23	0.109098
26	9	V9	0.106141
27	5	V5	0.089899
28	27	V27	0.074293
29	21	V21	0.071447

APPENDIX C – Source Code

The source code of each experiments are available in the public repository: <https://github.com/miguelpimentel/shap_feature_selection>

The experiments were developed from python notebooks. For each experiment, a notebook was created for each of the used models, that is, for the Cancer Breast dataset, four python notebooks were created, one for each model. The same process was introduced in the Credit Card Dataset.

When running each notebook the results are saved in the result folder for each of the experiments. A CSV file is created with the result of the metrics for each scenario of the experiment (number of features), and the charts are created for each of the metrics.

For each experiment, there is a notebook with the objective of gathering the results obtained for each model and creating reports and charts about the experiment as a whole.

Last but not least important, for any questions, read the README.md file and feel free to open an issue.