eSardine: A General Purpose Platform with Autonomous AI and Explainable Outputs

Inês Rito Lima[1][a], Nuno Leite[1][b], Adriano Pinto[1][c], Pedro Pires[2][d], Carlos Martins[2][e] and Nuno V. Lopes[1][f]

[1]*DTx Digital Transformation Colab, 4800-058, Guimarães, Portugal*

[2]*Mobileum, 4705-319, Braga, Portugal*

Keywords: Automatic Machine Learning (AutoML), Explainable AI (XAI), General-purpose, Telecommunications, Anomaly and Fraud Detection.

Abstract: The combination of high computational power and data awareness triggered an increasing demand for business applications from industrial players. However, harnessing the knowledge from data requires expertise, usually being a time-consuming task. Additionally, the users' trust in the results obtained is commonly compromised due to the black box behavior of most Machine Learning models. This paper proposes a general-purpose platform, eSardine, that leverages automatic machine learning and explainability to produce fast, reliable, and interpretable results. The eSardine platform integrates forefront tools to enhance, and automate the data science process, with minimal human interaction. For any tabular supervised classification and regression problems, predicted outputs are given, as well as an explainability report of each prediction. The inclusion of AutoML tools, *i.e.*, automatic model tuning and selection, presented a strong baseline whose capabilities are amplified by built-in, yet customizable, autonomous processing mechanisms. The explainable reports aim to increase users' confidence in the models' quality and robustness. Furthermore, in the industrial context, understanding key factors unveiled in these reports is determinant to increase the business model's profitability. The platform was evaluated in two public datasets, where it outperformed state-of-the-art results.

1 INTRODUCTION

In the past few years, the increase of computational power associated with data awareness and database access led to a growth of Machine Learning (ML) methods and Artificial Intelligence (AI) capabilities (Hutter et al., 2019; He et al., 2021; Došilović et al., 2018). These have recently reached remarkable performances, fulfilling the needs of different areas, from business to medicine, and all technological fields in between. In this sense, this progress can be seen, as well as have repercussions, in two main perspectives:

- The growth in algorithm complexity provides both more accurate and complex results for human to understand and interpret;
- Its application context is widening with an increase in the number of non-ML experts needing to pace up with these models.

In light of recent technological and computational developments, both these items have isolated solutions. The lack of interpretation of the results can be addressed by Explainable Artificial Intelligence (XAI). The need for facilitating and accelerating the ML processes, for both non-experts and experts, can be addressed by the automation of these processes, commonly known as Automatic Machine Learning (AutoML).

Usually, using machine learning on any given scenario follows the generic architecture of: i) input data; ii) pre-process it, including data cleaning, feature engineering and selection; iii) model selection and tune; and iv) output results for either classification or regression problems. This is an iterative process, trying to optimize performance, supported on expert judgment and knowledge in the area. Therefore, it is a

[a] https://orcid.org/0000-0002-9681-4740
[b] https://orcid.org/0000-0002-8558-2956
[c] https://orcid.org/0000-0001-9397-3722
[d] https://orcid.org/0000-0002-9928-620X
[e] https://orcid.org/0000-0002-0678-4868
[f] https://orcid.org/0000-0003-0169-0616

Lima, I., Leite, N., Pinto, A., Pires, P., Martins, C. and Lopes, N.
eSardine: A General Purpose Platform with Autonomous AI and Explainable Outputs.
DOI: 10.5220/0010970000003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 2*, pages 612-625
ISBN: 978-989-758-547-0; ISSN: 2184-433X

very time-consuming task (Hutter et al., 2019). In a business oriented perspective of ML usage, AutoML arises as a great solution to surpass the time and human resources invested in ML processes.

AutoML can automate both data processing and model tuning and selection tasks, always envisioning the best learning performances. Overall, one of the most challenging tasks of AutoML is the model hyper-parameters optimization in the tuning phase. This can be very computationally demanding and time-consuming, according to the problem and data inputted (Hutter et al., 2019; He et al., 2021).

However, as AutoML eases the need for human intervention, to some extent widens the gap between model operation and human understanding. Additionally, AutoML might return highly complex models, not tuned by human experimentation, which on one hand represents less time consumed in building a model, but on the other hand, less understanding of its results.

Embedded in this context emerged the concept of XAI (Miller, 2019). As problems become more complex, less intuitive are the algorithms solving them, where most work as black boxes. Due to the low explainability, the lack of transparency on their operating mechanisms, and on the outputted results, these ML methods are nowadays a liability. *Trust* is a commonly raised problem, as users have increasingly more difficulties understanding models, and therefore resistance to rely on a black box (Došilović et al., 2018; Miller, 2019). With AI spreading rapidly and reaching so many areas of business, trust is crucial for the success of using ML in that area of application. In this context, trust has two different meanings: the trust in the result given by the model and the trust in the model itself (Ribeiro et al., 2016). Consequently, having the capacity to understand the model's operation mechanism or results given, resorting to XAI methods, leads to confidence in the prediction obtained, trust in the model performance, or to the identification of any model's inconsistencies. Regardless of which, is undeniable the safety and reliability promoted by Explainable AI (Gilpin et al., 2018; Došilović et al., 2018).

Additionally, with recent misuse of data and more restrictive processes for data collection and use, the necessity of understanding the models and results has been addressed by international organizations such as the World Economic Forum and the European Union (EU) as a matter of safety and risk. *Performance*, *security*, *control*, *economical*, *ethical* and *societal* risks are the main concerning fields when it comes to AI (Combes et al., 2018). The European Union points out the advantages and challenges associated with AI, underlining the importance of digital trust obtained from XAI. It also acknowledges the risk associated with the increasing power on AI, having released the *European Union regulations on algorithmic decision-making and a 'right to explanation'* (Goodman and Flaxman, 2017). It is established by these organizations the need for further research on the topic, to surpass bias and improve explainability (Combes et al., 2018; Goodman and Flaxman, 2017).

Within this context, the present work explores the development of the eSardine platform. A tool with Explainable, Scalable and Adaptive Risk Discovery in Networked Ecosystems capacities.

This work proposes an advanced artificial intelligence platform able to automatically learn, detect and predict new types of fraud, providing at the same time explainable outputs. Considering its future deployment in industrial environments, this platform was built for real-time detection with the capability of handling continuously large amounts of data received via streaming or batch. Also, it supports model versioning and updates. To widen the range of business applications the platform was developed with the capability of being General Purpose (GP), *i.e.*, applicable to any type of supervised learning problems, characterized by tabular data.

There is already work that has focused on developing novel AutoML frameworks (Erickson et al., 2020; Wang et al., 2021; Wang et al., 2020) with the goal of improving some of its operating modules. Nonetheless, evolving to a business analytics platform, that integrates with industrial solutions and services, is still an open issue. Furthermore, there is still a gap regarding the seamless combination of AutoML with Explaianble outputs in an end-to-end analytics platform, which is an attractive and valuable functionality in any business context. These open issues are the main goals of this work.

In this sense, the main contributions, in line with the platform's requirements, are the following:

- Integration of AutoML system with XAI capabilities;
- A general-purpose platform that is capable of handling a variety of datasets without previous data or business knowledge;
- Update and versioning of both ML and XAI models;
- Distributed and parallel strategies for data processing and model deployment in big data oriented contexts.

Whereas the current section frames the conducted work, its motivation and contributions, Section 2

presents the current state of the art of key methodologies encompassed in the platform and available libraries and packages that support its implementation. Section 3 describes the key components of the eSardine platform, detailing as well their configurations. Section 4 presents the platform's architecture and describes in detail its operation life cycle. To validate eSardine's capabilities in terms of model tuning and selection, performance results on benchmark datasets, in distinct approaches, are given in Section 5. Aligned with eSardine's explainable functionality, Section 6 delves on XAI output examples, highlighting its advantages. Section 7 provides an overview of the conducted work and highlights future lines of research.

2 RELATED WORK

The development of the eSardine platform as an integrated forefront solution for general-purpose AutoML with explainable outputs relies on the awareness of the current state of the art of such technologies and how they operate.

2.1 AutoML

AutoML can be described as a Combined Algorithm Selection and Hyperparameter optimization (CASH) problem, which has two main difficulties: i) no single machine learning method performs best on all datasets; and ii) some machine learning methods rely on hyperparameter optimization (Feurer et al., 2015a).

Considering that AutoML targets both pre-processing stages and model tuning and selection, it is important to highlight techniques that can be used for each of these challenges.

Datasets usually have inherent problems, such as sparsity, missing values, categorical variables, or irrelevant features (Guyon et al., 2019). These can be partially tackled by pre-processing techniques that might help increase the model's performance. Some considered techniques to overcome these data constraints include (Feurer et al., 2015a; Guyon et al., 2019; Guyon et al., 2016; Feurer et al., 2015b): i) Variable Encoding – such as ordinal or one-hot encoding, which convert categories to numbers; ii) Matrix decomposition – such as Principal Component Analysis (PCA) that decompose data into maximally descriptive components, translating to a dimensionality reduction; iii) Univariate feature selection – by selecting features based on univariate statistical tests; iv) Classification based feature selection – leading to feature selection after an ML determining which are the determinative features for the task; v) Feature clustering – capable of merging highly correlated features; vi) Kernel approximation – without costly functions of kernel approximation to all points; vii) Feature embedding – projecting features on a feature space with embedding methods (*e.g.* random forests); and viii) Polynomial feature expansion – that expands a set of features by computing polynomial combinations. In the eSardine platform some of these pre-processing techniques will be implemented separately from the AutoML tool, in order to tailor and achieve the GP goal.

Regarding the problem of tuning an ML model, which is at the core of AutoML capacities, there are several strategies known to address the Hyperparameters Optimization (HPO) problem, namely (Claesen and Moor, 2015): i) grid search (Bergstra and Bengio, 2012; Snoek et al., 2012); ii) random grid search; iii) Bayesian optimization (Snoek et al., 2012); iv) genetic programming (Olson and Moore, 2019); v) particle swarm optimization; vi) coupled simulated annealing; and vii) racing algorithms. Each algorithm poses a replacement of intrusive manual search, which relies on previous domain knowledge and experience.

Regardless of the HPO technique used, a good complementary strategy is to conduct Cross-Validation (CV) as it helps avoid over-fitting of the hyperparameters to the training set (Bachoc, 2013).

Various open-source tools are already available and implement the mentioned strategies. In particular, during the research and implementation of the eSardine, three were considered: Auto-SKlearn (Feurer et al., 2015a), TPOT (Olson et al., 2016) and H2O (LeDell and Poirier, 2020).

AutoSklearn is based on the scikit-learn package, applying Bayesian optimization to identify the best pipeline from tested combinations. TPOT (Tree-based Pipeline Optimization Tool) relies on genetic programming to automatically design and optimized ML pipelines. H2O's AutoML is an open source tool developed within the H2O.ai company. It can automatically select the best-suited ML model, resorting to a randomized grid search.

Other market options such as Auto-Weka or TransmogrifAI were not considered due to its lower performance (Gijsbers et al., 2019) and lack of integration with other programming languages.

2.2 XAI

Usually, more detailed and tailored models lead to better performances. These are, however, more complex and, therefore, less explainable. XAI aims at pro-

viding an explanation for either the overall model behavior, or the reason for a particular output (Guidotti et al., 2018; Molnar, 2018; Lipton, 2018).

In this sense, the most characterizing classification of XAI models falls in two major classes: local or global interpreters, *i.e.*, if the XAI models provide global insights into the model learning process, or if it explains a single prediction obtained at some instance, respectively (Guidotti et al., 2018; Molnar, 2018).

Additionally, interpretable methods can also be categorized with regard to the type of machine learning method subdue to interpretation, as well as, its complexity.

XAI models classified according to ML models' complexity can have (Molnar, 2018): i) an intrinsic interpretation, if the ML algorithm is interpretable by nature (*e.g.* linear models, tree based models); or ii) post-hoc interpretation, if it will be applied to a black box ML algorithm (*e.g.*: neural networks, ensemble methods) to reverse engineer it. The core of the explanation is also connected to the previous criteria, as a: i) model-specific interpretation relies only on features such as p-values or decision rules, adequate for more interpretable ML models; whereas a ii) model-agnostic interpretation is more useful for post-hoc models, analyzing the input-output data interdependencies.

Despite the distinction between types of XAI models, all of them rely on interpretable explanators. As described in (Guidotti et al., 2018), these can be selected based on the problem faced or the type of ML algorithm used, from a range of: i) decision trees; ii) decision rules; iii) features importance; iv) saliency masks; v) sensitivity analysis; vi) partial dependence plots; vii) prototype selection; or viii) activation maximization.

From a wide range of current XAI solutions presented in (Guidotti et al., 2018; Molnar, 2018), two open-source XAI tools were considered to integrate the eSardine platform, based on their capability to provide local explanations whilst being agnostic to the ML model and data inputted: LIME (Ribeiro et al., 2016) [1] and SHAP (Lundberg and Lee, 2017) [2].

LIME (Local Interpretable Model-Agnostic Explanations) creates a new dataset with protuberances in data, which is given to the black-box model, to assess its new prediction. In this sense, it evaluates the impact of each feature in determining the output (Ribeiro et al., 2016).

SHAP (SHapley Additive exPlanations), based on Shapley Values and Game Coalition Theory, assesses for each feature the weight of all alliances that can be made, *i.e.*, all combinations with and without the feature, extracting its value. It gathers the marginal contributions across all possible coalitions to indicate the importance of that feature for the ML model output (Lundberg and Lee, 2017).

2.3 Big Data Engine

Whereas the AutoML and Explainable requirements have each specific associated technologies, the general-purpose and big data requirements both rely on a single software engineering technology. As will be further discussed, the general-purpose capability is accomplished by the data handling and processing strategies developed for the eSardine platform. These methods were built resorting to a big data engine, to achieve both data generalization and scalability.

Big data engines are usually characterized by mechanisms that distribute tasks over various devices within a cluster (Rao et al., 2019). Additionally, parallelization techniques are usually also applied in each device to achieve higher computational performances.

Ultimately, the goal of providing big data analytical properties is to be capable of handling and processing a variety of high volume data, at high computational speed (Rao et al., 2019).

Considering the industrial applicability desired for the eSardine platform, technologies and mechanisms beyond code optimization have to be applied in order to make it scalable, both at data processing tasks and at handling various heterogeneous workloads at the same time.

Scaling the platform, in terms of its data processing tasks, to the petabyte order can be done by using frameworks and engines already designed for that effect. While there has been a lot of developments in this area, resulting in the existence of various big data oriented frameworks, two standout: 1) **Apache Spark**, defined as a "unified analytics engine for large-scale data processing" (Spa, 2020) is, arguably, the most acknowledge framework for large-scale data processing; 2) **Dask**, known to be a flexible library for parallel and distributed computing in the Python ecosystem (Das, 2020), has grown in active users due to its simple integration with the enormous python data science ecosystem.

The deep understanding of state-of-the-art key technologies available and their functioning is paramount for selecting compatible components, whose combination foster the optimal performance envisioned for the eSardine platform.

[1] https://github.com/marcotcr/lime

[2] https://github.com/slundberg/shap

3 eSardine PLATFORM ARCHITECTURE AND KEY COMPONENTS

As previously mentioned the eSardine platform aims to provide AutoML capacities associated with explainable outputs, easing the processes of data-science and ML in the context of tabular data, regardless of its size. The platform's general-purpose goal is achieved by the data processing pipeline, itself developed upon a distributed framework to enable big data capacity.

3.1 Design Goals

Combining and extending the design goals highlighted by (Erickson et al., 2020; Wang et al., 2020), the eSardine platform was envisioned and developed to fulfill the following design goals:

- Interoperability – between each key component in the platform, maximizing benefit from their joint action;
- Integrability – of the proposed platform within the business solution and service provided;
- Scalability – to be able to handle in the petabyte order a day, according to business requirements;
- Versatility – to adjust to a variety of structured datasets that may be provided, fulfilling the general-purpose business requirement;
- Reactivity – to provide near real-time predictions when operating in the business context;
- Simplicity – in its usage, requiring few interventions for both experts and non experts users;
- Configurability – enabling users to incorporate business knowledge and maximizing performance based on experience;
- Flexibility to pluggable operations - enabling users to develop custom processing transformers and filters, if not within the available ones;
- Versionability – complying with business requirements of retraining and improving models based on previous ones, preserving previous knowledge while integrating new data variability;
- Robustness – providing strong results regardless of the input conditions or the dataset;
- Timing predictability – by being able to define and limit training time and/or ML models' family in accordance with the experiments carried out by the user.

3.2 Global Architecture

In order to fulfil the presented design goals, Figure 1 illustrates the global architecture of the eSardine platform.

First, eSardine platform performs an analysis of the input data, to extract the dimension of its feature space and the different data types present. Then, the Describer module computes a predefined set of information retrieval functions to overall describe the input data. At the same time, this module has the capability to propose a set of transformations ought to be applied to the input data, by the Transformer module. Based on either the Describer recommendations or the user predefined transformations, the eSardine platform instantiates a set of pre-processing transformations in the Pre-processor module. The Transformer module then applies these transformations to the data. During the learning phase, the output of the Transformer module, combined with the target labels, sets the classification task and validation schema. These modules are further described in detail in Section 4.1. The final stage of eSardine considers h2o.autoML, to learn and obtain estimations for each input sample, followed by an interpretability stage, detailing the motivation for each outputted prediction. Section 3.3.1 and Section 3.3.2 detail the specifications of H2O AutoML and LIME, respectively, being both components evaluated on Section 5 and Section 6, respectively.

3.3 Key Components

The core of eSardine relies on the interoperability among three key components: AutoML, XAI, and big data components. The selection of these components was based on their characteristics, compatibility and capacity to fulfill the eSardine requirements.

In this work, we present a fully automatic data processing and data modeling pipeline, built upon the open-source key players H2O AutoML, LIME, and Apache Spark tools, as shown in Table 1.

Table 1: Key functionalities considered in the eSardine platform and respective selected packages (*i.e.* components).

Functionalities	Pacakges
Automatic Machine Learning	H2O AutoML
Explainable AI	LIME
Big Data Engine	Apache Spark

Among the tools considered for AutoML, discussed in Section 2, H2O AutoML stands out as the most suitable package, given its capabilities of: i) being made available by an industry leader, which implies reliable and stable versions, with a strong community, for both use and support; ii) independent and

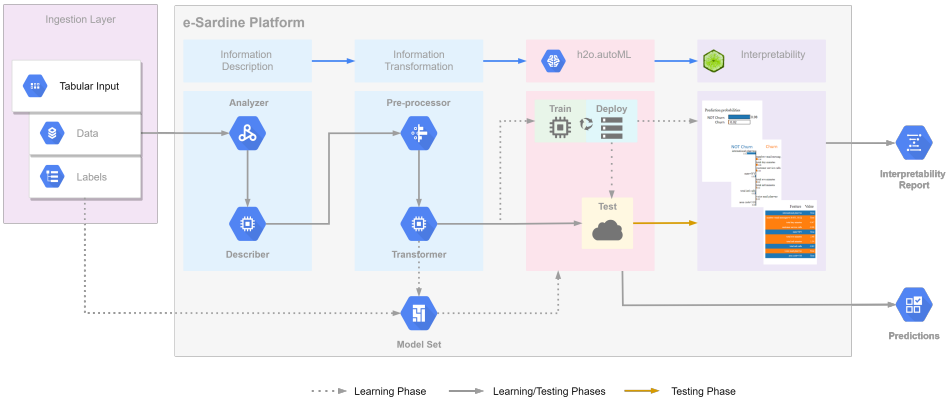

Figure 1: Overview of the eSardine modules and its operation mode during the distinct phases of application.

automatic control over the life cycle of machine learning models, as opposed to pipelines, enabling the integration with the platform capability of tailoring the dataset preparation steps; iii) model checkpointing[3], easing the versioning process; iv) distributed computing, enabled by H2O Clusters, aligned with platform's big data oriented requirement. Additionally, the easy integration, based on H2O's light system requirements, and the capacity of tailoring the models selection and parameters optimization were also considered. The model selection being conducted on cross-validation metrics also fosters its robustness and increases trust in models' performance.

AutoSklearn, in the context of big data applications, was discarded given that it was developed having as target small to medium-sized datasets (Feurer et al., 2015a). Also, its pipeline optimization strategy explores a fixed set of pipelines, which only include one data processor and model (Olson et al., 2016). This interferes with GP capacities and may compromise performance, integration, and tailoring of the pre-processing steps in the eSardine platform. The AutoSklearn is also less interoperable due to its system requirements.

TPOT, despite based on scikit-learn package, allows multiple operator combinations and distributed training, which is essential for big data analysis. Also, model selection is based on cross-validation performances, and model saving is optional. These would make TPOT the next best choice. However, its pipeline-based approach, instead of model optimization, the required computational time (Olson et al., 2016), complex model retrain, and lower established community favored the choice of the H2O AutoML tool.

Additionally, a benchmark study, comparing AutoML tools, has identified the H2O AutoML tool as the best performing across various datasets (Ferreira et al., 2020), which highlight its generalization capacities, useful for integration in the eSardine platform. H2O AutoML is reported to be the most widely used framework in the field, outperforming other available solutions (Erickson et al., 2020).

Simultaneously, it was assessed and verified the integration capacity of H2O AutoML with available XAI tools. Comparing LIME and SHAP models, it was considered the former to be more integrable with H2O AutoML. However, the platform was built so that future versions can easily integrate another XAI tool.

Being the eSardine built to be a general-purpose AutoML tool with explainable output for small and big data analysis, despite requirements addressing tabular data, components were selected based on their agnostic relation to data. In this sense, the platform can evolve and be scaled to other data characteristics, without future versions being limited by its components' limitations.

3.3.1 H2O AutoML Specifications

H2O AutoML solution (LeDell and Poirier, 2020), automatically trains, tunes, and selects the best performing model, for a given dataset, from a leaderboard of trained and tuned models. It resorts to a randomized grid search for the HPO problem, training and cross-validating both classification and regression models. The leader models are obtained from cross-validation results, being the performance metrics adjusted to the problem considered, which increases reliability in the overall performance of the model to be deployed. The number of models trained, the metrics by which they are assessed, the time available to tune them, and which models should be included/excluded are all configurable parameters. This allows a more problem-specific adjustment of the tool. The models

[3]H2O's method that allows building a new model on top of a previously trained one as new data is considered

available include a default Random Forest (DRF), an Extremely Randomized Forest (XRT), pre-specified H2O Gradient Boosting Machines (GBMs), a random grid of XGBoost GBMs and H2O GMBs, a random grid of Deep Neural Nets, and a fixed grid of GLMs'. H2O AutoML can also train two Stacked Ensemble models, one with all models and another with the best performing from each algorithm family. Considering the architecture presented previously, the checkpointing capacity of the H2O models is crucial, responsible for enabling model retrain based on new additional data. Due to this necessity, H2O AutoML ensemble algorithms are, by default, discarded, being a configurable number of models in the leaderboard saved for retraining and posterior ensemble within the eSardine platform. Stacked ensemble models are built inside the eSardine platform but out of the H2O, allowing a continuous update of the models and their versions.

3.3.2 LIME Specifications

Linked with each H2O leader model, there is a XAI model, in this case, a LIME model. LIME (Ribeiro et al., 2016) focuses on training local surrogate models to explain individual predictions of the black-box model. These surrogate models are trained in order to approximate the predictions of the associated ML model that they try to explain (Molnar, 2018; Ribeiro et al., 2016). To do so, LIME creates a new set by perturbing the original samples, from the training set provided to H2o AutoML, and assesses the corresponding predictions using the leader model, in order to test its behavior according to those data variations or perturbations. LIME uses feature importance as an interpretable explanator, thus varying them and identifying separately which features lead to which result.

While capable of providing explanations for any ML model and any type of data (tabular, text and images) (Molnar, 2018; Ribeiro et al., 2016), the main drawback of LIME is the fact that it relies on data perturbations, which might be unstable or prone to sampling issues (Molnar, 2018).

Considering the possible retrain and versioning of H2O leaders and leaderboards, every new update to the leader model requires a newly trained LIME model, which is equally versioned and coupled with its associated ML model.

3.3.3 Platform's Big Data Oriented Requirements

Selecting a big data oriented engine capable of handling up to the petabyte order a day in near real-time requires thorough assessment between the considered frameworks: Apache Spark and Dask. Whereas Spark is known to be a strong choice for scalable analytics, Dask enables easy integration with various python packages. Therefore, a benchmark analysis was conducted to identify the optimal big data engine to operate at the core of the eSardine platform.

This analysis resorted to the New York City Yellow Taxi Trip (YTT-NYC) Record Data, years 2011, 2012 and 2013[4], whose column's type heterogeneity (including numeric, Boolean and categorical variables) and volume are representative of the challenges that the eSardine platform might face. Each year's data is accountable for 32 GB, which were organized into 4 distinct datasets: 1) YTT-NYC'11, comprising the year 2011 of YTT-NYC; 2) YTT-NYC'11 and YTT-NYC'12; 3) YTT-NYC'11, YTT-NYC'12 and YTT-NYC'13; 4) In the union of the 2011, 2012 and 2013 datasets, followed by the union with itself which, in terms of size, made this last dataset have around 190 GB of size.

For each of these sets of data, the benchmark process followed the following guidelines strictly:

- Every execution was performed three times and the value considered was the average of those three executions. The goal was to rule out one-hit wonders.
- Any failure and recovery that happened was not disregarded and was made part of the comparison, providing information regarding the consistency and availability of the engine.
- The frameworks were test-stressed at 10%-20%, *i.e.*, the underlying hardware in terms of RAM (Random Access Memory) should only cover 10 to 20% of the size of the dataset in, at least, two different end-to-end executions.
- Every end-to-end execution should go through and measure execution times of four main steps:
  1. build the dataset (reading and appending) and persisting it in the Dask or Spark cluster;
  2. calculate the average trip distance grouped by the passenger count;
  3. calculate the average fare amount grouped by the payment type; and
  4. fit and transform the data with a pre-processing pipeline. The pre-processing pipeline applies 10 operations: eight standard scaling normalization operations and two ordinal encoding operations.

The computational capability enabled by the hardware used in both Dask and Spark runs was equivalent

[4]Amazon S3 bucket: https://registry.opendata.aws/nyc-tlc-trip-records-pds/

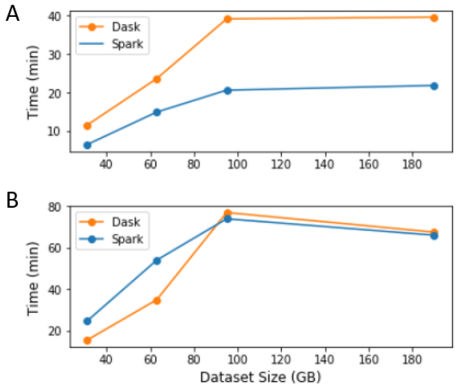

Figure 2: Frameworks comparison as data volume increases: A) when applying the pre-processing pipeline (step 4); and B) when reading data and applying the pre-processing pipeline (steps 1 and 4).

for comparison purposes. The first 3 datasets (with 31 GB, 63 GB and 95 GB) were executed (end-to-end) on a cluster with two workers [5], having these resources been doubled for the last dataset (190 GB).

The stress test evaluated the frameworks' robustness by increasingly diminishing computational capacity as data volume increased [6].

This big data oriented assessment compared both frameworks' performances, at each processing step, as data volume increases (Figure 2).

It was found that, while Dask surpassed Spark performances during step 1 (reading and appending), Spark strongly outperformed Dask in both grouping operations (steps 2 and 3). Please see the supplementary material for additional information. [7]. Considering the background of the eSardine platform and the challenges it might face, grouping operations are likely to be more recurrent and demanding than step 1 operations. The performance results obtained for step 4 are presented in Figure 2 A), showing that Spark is consistently and increasingly faster than Dask in applying a pre-processing pipeline. Figure 2 B) illustrates the summed time required for reading and applying the given pre-processing pipeline, to better perceive how both frameworks would behave in a real use case.

Although Dask handled the 2 lower-sized datasets faster, mainly due to its higher efficiency in data ingestion tasks, Spark overthrown Dask as data volume increased, establishing itself as the overall best framework to include in the eSardine platform. Additionally, apart from the high performance in applying a pre-processing module, Spark's efficiency when handling grouping, aggregation, and filtering tasks make it the optimal framework to implement customized operations with.

[5]Each worker included four cores and 8GB of RAM

[6]Computational capacity determined by RAM/Size, with 51.6%, 25.4%, 16.8% and 16.8% ratios for each dataset respectively

[7]Besides being less efficient, one of Dask's workers ran out of memory and was not able to automatically reboot, requiring human interaction

4 PLATFORM LIFE CYCLE

The eSardine platform architecture was designed in order to cohesively connect its key components in a distributed strategy, to ensure that all requirements are fulfilled. The platform architecture integrates four main stages: preparation, operation, results and maintenance, as illustrate in Figure 3. These stages comprehend the entire cycle of data and models through time.

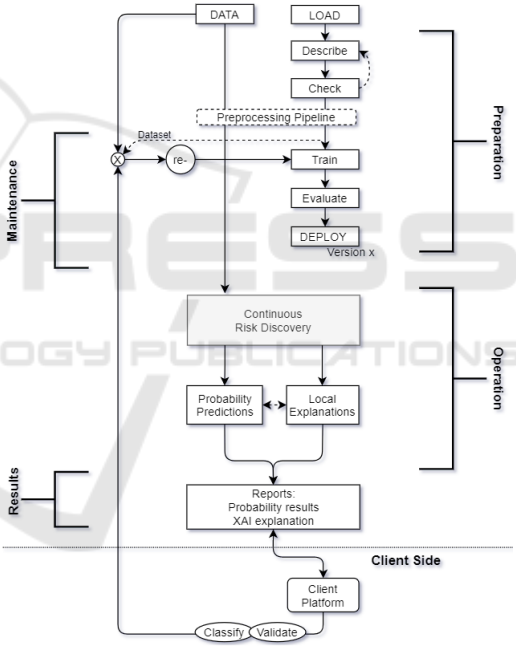

Figure 3: eSardine platform data flowchart.

4.1 Preparation

Preparation is a key stage of the eSardine platform, built in accordance with the GP requirement, to allow full control over data pre-processing actions. This stage is responsible for setting the platform ready for operation on demand.

This stage comprehends actions ranging from the inputted dataset to a trained ML and XAI models for deployment. After loading the dataset that will initially be used to train and test the models, the first provided output from the platform is a description

and analysis of the dataset. This description phase allows the user to understand any characteristics and highlight concerns regarding the dataset, such as the type of features included, the nature of their correlation, the occurrence of missing values, the features relevance and their contribution to the dataset global information, class imbalance, among others. From this analysis, the describer module also suggests a schema, which globally characterizes the type of data of each feature, and recommends applicable filters and transformations, to foster an increase of models' performance. The criteria for inclusion/exclusion of data is configurable, but default parameters are set to ensure the flow in case of no intervention from the user. Explanations on the actions to be taken are given, to increase users' awareness and foster intervention and customization. This process occurs iteratively, since after checking which actions are suggested, the user can customize this schema as many times as necessary before training. If no user input is given, the suggested schema is used, automating the pipeline and flow of data in the eSardine platform. The pre-processing pipeline established by this schema is preserved, since new data must undertake the same processing actions before being assessed by the model or used for retraining.

Once data is pre-processed, the training module resorts to AutoML to identify the best performing model from a leaderboard of tuned models. The number of models, training time, metrics applied, among other criteria, can be configured by the user. As discussed, the leader model selection is based on cross-validation performance to increase model reliability for deployment. Using a podium of configurable number of models from the leaderboard, a stacked ensemble is computed for each combination of the models from that podium. The ensembles' performance is also assessed in cross-validation and the best ensemble is compared to the leader, in order to identify which model should be deployed in the initial version of this pipeline.

Once the ML model is selected, the XAI model can be trained. In each deployed version are released both an ML model and its own XAI model Therefore, each ML model has its own XAI model, being both deployed in each version. Along with the leader and its associated XAI model, the configured podium is saved, as it will be necessary for the maintenance stage.

This stage ends with the deployment of a fully trained and tuned H2O model and its respective LIME model.

4.2 Operation

With the deployment of a stable version of an ML model and its respective LIME explanator, the eSardine platform operates continuously in order to predict and explain each sample that arrives by streaming or batch. This data has previously been processed by the pre-processing pipeline approved during the preparation stage.

The output of continuous risk discovery has two parcels: the ML model deployed is responsible for providing the prediction's probability, and LIME for the explanation of that prediction.

4.3 Results

The results correspond to the outputs of the operation stage which are made available to the user. These can be accessed in a report, that includes, for each sample, both the probabilities given by the ML model, and the explanation provided by LIME. The classification is conducted at the user's end, having control over the best suited threshold for the task, which in turn allows the possibility of defining a gray zone threshold, *i.e.*, a range of probabilities that require additional attention from the user. The samples in the gray zone can be targeted for a new manual classification useful for future models' retrain.

4.4 Maintenance

Maintenance is the stage responsible for keeping the models up to date and reactive to changing data trends. The eSardine platform allows models versioning, *i.e.*, replacing the deployed models with new ones trained on new or corrected data, guaranteeing the best performance, adjusted to subtle differences in data through time. New versions will be released as a result of the maintenance stage, and this process can be configured to occur automatically after a given time or number of samples processed.

The new data, that is continuously assessed by the currently deployed version, can be used for posterior model retrain and version update, allowing this model to absorb more information and update its previously tuned parameters.

Also, as shown in Figure 3, the platform interacts with the client through users, whose expert judgment can be useful to increase models' performance. The user's feedback to the obtained results grants model validation and might unlock further performance levels. Apart from the classifications given that do not attract attention, the samples that have a grey area probability, which by nature underline bias scenarios,

might be manually classified by an experienced user, and fed for model retraining.

Model's maintenance occurs not only on the leader of the current version, but also to the podium saved during preparation. In fact, new data may lead one of these models to outperform the retrained past leader, especially considering ML models that require large amounts of data to adequately tune their parameters. Besides retraining the podium and assessing its new performances on CV, the stacked ensemble of each combination of these retrained models is computed, and its performance analyzed. From this process, a new leader and podium are saved. In case the best performing algorithm is the one from the previous version, it shall remain the operating model. However, the retrained podium is saved and used in the next retrain iteration, as it contains the last overall data information.

Once defined the leader, the LIME model is computed, and that version is deployed for operation. In this sense, for each version, it is preserved the leader, the podium and the associated XAI model.

5 PLATFORM'S PERFORMANCE

To assess the eSardine platform capabilities, two datasets were considered: the CrowdAnalytix Churn prediction dataset and the Wine Quality prediction dataset. The selection of these datasets, which require different processing techniques and are handled as distinct ML tasks, aimed to demonstrate the general-purpose capabilities of the platform. Having this platform been built in partnership with a player in the telecommunication industry, the Churn dataset exemplifies a use case for this sector, handled as a binary classification problem. The Wine Quality dataset is handled as a regression problem, with the particularity of the model's performance being evaluated as a multi-class classification problem.

Also, addressing the open issue highlighted by (Hanussek et al., 2020) stating the lack of comparison between expert human tuning and AutoML, this work evaluates the performance of the eSardine platform with the interaction of a human data scientist.

To emphasize the advantages of resorting to the eSardine platform, model's performances were obtained for 3 independent analysis scenarios: i) using purely the H2O AutoML tool; ii) using the eSardine platform set by its default parameters and with the automatic schema; and iii) customizing parameters in the eSardine platform, both in terms of processing schema and training settings. The presented performance metrics were obtained with CV, averaged among 3 isolated training runs. Methodology [8] was defined to be equivalent to the previous work on these datasets (Churn -(Umayaparvathi and Iyakutti, 2016) , Wine Quality - (Cortez et al., 2009)), having the Churn dataset been evaluated on 10 fold cross-validation metrics and the Wine Quality dataset assessed resorting to 5 fold cross-validation metrics.

Churn Prediction

Churn is a major problem in the telecommunication industry, which reflects the will of a customer to unsubscribe a given service. Since the costs of obtaining a new costumer are higher than retaining an existing one (Umayaparvathi and Iyakutti, 2016), the task of predicting churn costumers can represent significant savings. The dataset considered has 3333 samples, with 20 features to predict the binary target variable of *Churn*, whose occurrence accounts for 14.5% of all data. H2O AutoML evaluated the performance on this dataset resorting to the accuracy as ranking score to select the top classification model. In addition, it also computed commonly used metrics for classification tasks, namely the F1-Score, Precision and Recall.

Following the described methodology, Table 2 summarizes the performance results obtained for the Churn dataset[9][10].

The results obtained outperform current the State-of-the-Art (SoA) (Umayaparvathi and Iyakutti, 2016), with incremental performance between the three evaluated scenarios. The best results are obtained from slight tweaking the automatically proposed processing phase of the eSardine platform. As shown, H2O tool provides a solid baseline in comparison with SoA performance, whose results are only enhanced by the processing architecture of the eSardine platform. This automatic processing architecture emerges based on the inherent characteristics of the dataset and its feature types. It included: the exclusion of 4 features due to the high correlation value with other features, the exclusion of another feature based on the percentage of the unique values, the encoding of 4 categorical features (including the target) having one of these been segmented into an *others* class due to its sparsity. Upon these strong processing steps suggested by the platform, the customization added a standard scaling of data, binning and encoding of features based on business context. Additional class balancing (ratio of

[8]Number of cross-validation folds and performance metrics

[9]Performance metrics focus the class *Churn*, *i.e.*, computed using the occurrence of *Churn* as the positive class

[10]The AutoML tunes as selects leader models, being the most common models in the H2O methodology the GBM, and in both eSardine methodologies the XGBoost

Table 2: Performance results obtained for the CrowdAnalytix Churn dataset.

	H2O	eSardine	eSardine custom	SoA
Accuracy	0.9492 ± 0.0148	0.9558 ± 0.0124	**0.9601** ± 0.0123	0.9520
Precision	0.8561 ± 0.0603	0.8849 ± 0.0590	**0.9166** ± 0.0656	0.9088
Recall	0.7872 ± 0.0376	0.8016 ± 0.0492	**0.8029** ± 0.0387	0.7433
F1-Score	0.8195 ± 0.0405	0.8399 ± 0.0423	**0.8544** ± 0.0362	0.8178

0.9 and 1.5 for the under and over-sampled classes, respectively) was implemented without significant performance increase.

Wine Quality Prediction

Considering the wine quality prediction task, the dataset used was originally developed by (Cortez et al., 2009), and it is split between white and red wine. The analysis conducted resorted to the white wine dataset, which includes 4898 samples of distinct white wines and 11 characteristics to characterize the quality. Conceptually, the quality can range from 0 (very bad) to 10 (excellent), yet the present dataset only contains ranges between 3 and 9. Although there are multiple quality classes, this dataset was handled as a regression problem, evaluated with Mean Average Error (MAE) and then cross-validation predictions accuracy computed within different tolerances, promoting adequate comparison with the SoA results from (Cortez et al., 2009). Tolerance of 0.25, 0.5, and 1 (T=0.25, T=0.5, T=1) were considered, meaning a class is considered to be well predicted if within these ranges (above and below). The results of these 3 different scenarios are shown in Table 3, averaged across 3 independent runs, each with 5 fold cross-validation metrics[11].

Overall, results show that every scenario considered outperformed current SoA results, either in deviance from the original value, and in each computed accuracy. As the White Wine dataset used is already very clean, the automatic pre-processing schema provided by the eSardine platform does not add any step. Therefore, the results obtained between H2O and automatic eSardine do not present significant variance. For this type of dataset, since data do not require processing for performance enhancement, the pre-processing stage focused on fostering the understanding of dataset's characteristics. For the customized runs of the eSardine platform, a standard normalization of zero mean and unit deviation was included. Due to the unbalanced nature of data [12], a class balancing with under and oversampling was also considered [13]. The combination of these two additional processing stages led to an increase of performances across all metrics, achieving the top rank in this analysis.

The obtained results on both datasets demonstrate that eSardine has the capability to outperform the current SoA. Furthermore, it has been already evaluated in a real-life confidential business dataset attaining robust classification performances. Thus, in an era where the increasing use and demand of AutoML highlights the need for more robust solutions (Wang et al., 2021), eSardine platform presents itself as ready to fulfill the promise of increasing business value in demanding industrial contexts.

6 EXPLAINABLE OUTPTS

As previously mentioned, explainable outputs improve user experience with ML, since these promote trust in both the quality and reliability of the models and in the accuracy of the results. In business contexts, this is fundamental so that decisions are made based on solid knowledge of *What* is happening and *Why* is happening.

To exemplify LIME's behavior within the eSardine platform, a set of examples on the CrowdAnalytix Churn dataset is given. The explainer was built on top of an ML model, in this case, a model obtained during a customized eSardine platform run. Therefore, the pre-processing stages considered in the previous section are built-in the model and expressed in the explanations given.

In this analysis, three distinct scenarios are covered: i) an instance is correctly predicted as not churn; ii) an instance is predicted as churn; iii) an instance is incorrectly predicted as not churn. Figure 4 presents the results obtained for these scenarios, respectively.

In this figure, the color blue is associated with characteristics and probabilities of not incurring in churn, whereas orange is linked to churn. To the left are the probabilities computed by the ML model, which in this case is an XGBoost classifier, following the customized eSardine pre-processing, with 117

[11]The AutoML tuning and selection elected as leaders mostly XGBoost models across all three methodologies

[12]Class relative frequency: 3: <1%; 4: 3%; 5: 30%; 6: 45%; 7: 18%; 8: 4%; 9: <1%

[13]Sampling ratio per class, in ascendant order: 1.3; 1.2; 1; 0.8; 1; 1.2; 1.3

Table 3: Performance results obtained for the White Wine Quality dataset. Bold values comprehend the highest scores per each evaluated metric.

	H2O	eSardine	eSardine custom	SoA
MAE	0.3878 ± 0.0094	0.3872 ± 0.0090	0.3830 ± 0.0083	**0.4500**
Accuracy$_{T=0.25}$	0.5284 ± 0.0109	0.5295 ± 0.0096	**0.5317** ± 0.0149	0.5030 ± 0.0110
Accuracy$_{T=0.50}$	0.6967 ± 0.0112	0.6969 ± 0.0113	**0.6997** ± 0.0094	0.6460 ± 0.0040
Accuracy$_{T=1.00}$	0.9034 ± 0.0079	0.9028 ± 0.0073	**0.9048** ± 0.0057	0.8680 ± 0.0020

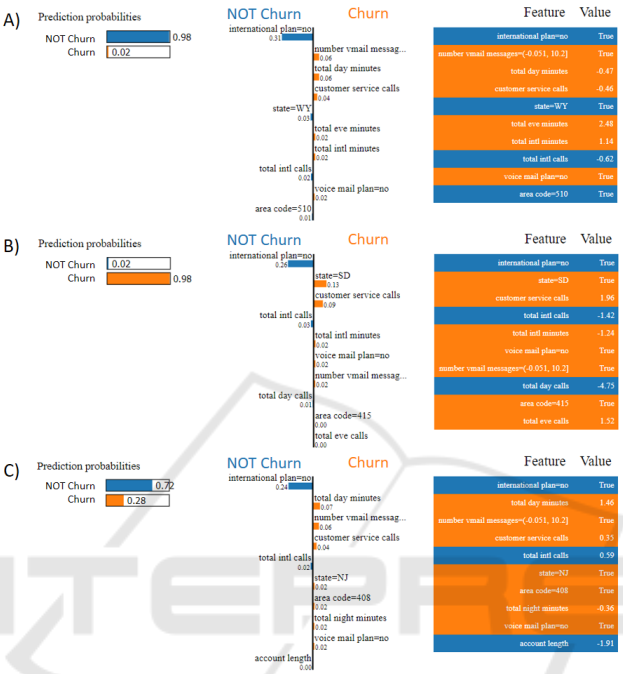

Figure 4: LIME Explanations: A) correctly predicted not churn; B) correctly predicted churn; and C) incorrectly predicted as not churn.

trees, 10 fold cross-validation, and trained with 90% of data. The remaining 10% were considered as testing set so that new data could be impartially predicted and explained by both the model and its LIME explanator, respectively. Due to the mentioned processing, only 15 features are valuable for predicting churn, being presented in the explanations only the top-10 contributing ones. The tables on the right side of this figure indicate in descending ranking the features which contributed to the prediction/explanation and their true value, shaded in the color of the binary class for which they have contributed. The center column provides a visual explanation of the individual contribution of the key features towards a specific class. These features, displayed against the central black line of zero weight for the prediction, are as long as their contribution, and in the color and side of the class they promote. Analysing this information increases reliability in the model, as features with the same value are consistently placed towards the same class. For example, the *international plan: no* targets the no *not Churn* class or the *number vmail messages* in the bin *[-0.051, 10.2]*, *i.e.*, below 10 calls fosters a *Churn* classification. These are criteria guidelines towards each class, internally computed by the XAI, which the user can validate if aligned with previous business knowledge and limitations (it may be acknowledged that a given *state* is more likely to churn, or that the subscription of a specific plan encourages the user to not churn).

By observing the features contributions and their consistency across different instances, scenarios A) and B) of Figure 4 are easily understood by their adequate prediction of not churn and churn, respectively. However, scenario C) where a churn case was missed and predicted as non-churn demands a more in depth analysis. While the classification confidence (*i.e.* the output probability) is not as certain as in the previous scenarios, the *international plan: no* has a high contribution to this prediction. In fact, this feature contributes with 0.24 to the 0.72 certainty of not churn, for which, without this specific characteristic,

this scenario would only be predicted to 0.48 as not churn, and therefore accurately predicted. Although misclassified, this instance also increases trust in the robustness of this model, since this specific feature is the one with the biggest weight on the intrinsic criteria, and even though this costumer has churned, its behavior was similar to a non-churn client.

Overall, the concept of AutoML increases the lack of trust in models, since there is little control over which model and tuned parameters are selected. The inclusion of LIME explanations in the eSardine platform is a step forward to help increase reliability and intelligibility in the models obtained and to help mitigate the concept of black boxes in ML. Despite extremely helpful, future work still needs to be conducted to increase XAI results' interpretability to users not familiar with data analytics processes.

7 CONCLUSION

The eSardine is a general-purpose platform, big data oriented, capable of providing new insights into data and extract powerful models, while accessible and understandable by a wide range of users in various sectors. This platform provides a fully functional and independent operating system, with adequate preprocessing steps, visual feedback on the data distribution and its properties, AutoML capacities associated with explainable outputs, and the possibility to retrain and update. In this way, the models can be updated, for a continuous life cycle of operation, which is a key functionally in the business sector.

Although automatic and autonomous, the user can tailor each step in the process to improve performance, from how to handle missing values, to which meta-learner is used in the ensemble, or when should the model be retrained. For new users or new data, the platform can provide, without intervention and knowledge on the data, the best customization of its GP processes based on the inherent characteristics of the dataset.

Ultimately, this is both a versatile and highly efficient platform, allowing an automatic functioning for fast and reliable results, with the tailoring functionalities that a user may require to enhance performance for a given task. Nonetheless, there is still room for improvement, where we envision the inclusion of NLP and image recognition processing pipelines, which will be unified with the AutoML and XAI modules, promoting a fully data-agnostic GP platform. Being explainability a recent area of research, efforts will also be conducted to ensure that reports built on XAI outputs are understandable outside the data science line of business.

The general-purpose function enabled by the integration of GP tools and the tailoring of each processing pipeline and data description widens the range of applications. The concern for the use of distributed computing tools ensures top and fast performances while handling big data. The integration of AutoML and XAI models enables fast results without compromising performance and ensuring user's trust in both the model and results given. The understanding is also fundamental in business to understand weak spots and liabilities and target them with preventive and corrective actions. Retrain enables the use of this tool by companies, where data is collected ceaselessly.

The unification of these functionalities in the eSardine platform results in a powerful analytics tool that can foster the growth of several businesses while unveiling the power of ML in an understandable manner.

ACKNOWLEDGEMENTS

This work has been supported by NORTE-06-3559-FSE-000018, integrated in the invitation NORTE-59-2018-41, aiming the Hiring of Highly Qualified Human Resources, co-financed by the Regional Operational Programme of the North 2020, thematic area of Competitiveness and Employment, through the European Social Fund (ESF).

REFERENCES

(2020). Apache Spark - Unified analytics engine. https://spark.apache.org/.

(2020). Dask - Scalable analytics in Python. https://dask.org/.

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Claesen, M. and Moor, B. D. (2015). Hyperparameter Search in Machine Learning. *XI Metaheuristics International Conference*, pages 10–14.

Combes, B., Herweijer, C., Ramchandani, P., and Sidhu, J. (2018). Fourth Industrial Revolution for the Earth Harnessing Artificial Intelligence for the Earth. *World Economic Forum*.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553.

Došilović, F. K., Brčić, M., and Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. (2020). Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*.

Ferreira, L., Pilastri, A., Martins, C., Santos, P., and Cortez, P. (2020). An automated and distributed machine learning framework for telecommunications risk management. In *ICAART (2)*, pages 99–107.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., and Hutter, F. (2015a). Efficient and robust automated machine learning. In *Advances in neural information processing systems*, pages 2962–2970.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., and Hutter, F. (2015b). Supplementary material for efficient and robust automated machine learning. *Advances in Neural Information Processing Systems*, pages 1–13.

Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., and Vanschoren, J. (2019). An open source automl benchmark. *arXiv preprint: 1907.00909*.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE.

Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

Guyon, I., Chaabane, I., Escalante, H. J., Escalera, S., Jajetic, D., Lloyd, J. R., Macià, N., Ray, B., Romaszko, L., Sebag, M., et al. (2016). A brief review of the chalearn automl challenge: any-time any-dataset learning without human intervention. In *Workshop on Automatic Machine Learning*, pages 21–30.

Guyon, I., Sun-Hosoya, L., Boullé, M., Escalante, H. J., Escalera, S., Liu, Z., Jajetic, D., Ray, B., Saeed, M., Sebag, M., et al. (2019). Analysis of the automl challenge series. *Automated Machine Learning*, page 177.

Hanussek, M., Blohm, M., and Kintz, M. (2020). Can automl outperform humans? an evaluation on popular openml datasets using automl benchmark. *arXiv preprint arXiv:2009.01564*.

He, X., Zhao, K., and Chu, X. (2021). Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622.

Hutter, F., Kotthoff, L., and Vanschoren, J. (2019). *Automated machine learning: methods, systems, challenges*. Springer Nature.

LeDell, E. and Poirier, S. (2020). H2o automl: Scalable automatic machine learning. In *7th ICML workshop on automated machine learning*.

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

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

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

Molnar, C. (2018). *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*.

Olson, R. and Moore, J. (2019). *Automated Machine Learning*, chapter TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning, pages 151–160. Springer.

Olson, R. S., Bartley, N., Urbanowicz, R. J., and Moore, J. H. (2016). Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 485–492, New York, NY, USA. ACM.

Rao, T. R., Mitra, P., Bhatt, R., and Goswami, A. (2019). The big data system, components, tools, and technologies: a survey. *Knowledge and Information Systems*, pages 1–81.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1–9.

Umayaparvathi, V. and Iyakutti, K. (2016). Attribute selection and customer churn prediction in telecom industry. In *2016 international conference on data mining and advanced computing (sapience)*, pages 84–90. IEEE.

Wang, B., Xu, H., Zhang, J., Chen, C., Fang, X., Kang, N., Hong, L., Zhang, W., Li, Y., Liu, Z., et al. (2020). Vega: towards an end-to-end configurable automl pipeline. *arXiv preprint arXiv:2011.01507*.

Wang, X., Li, B., Zhang, Y., Kailkhura, B., and Nahrstedt, K. (2021). Robusta: Robust automl for feature selection via reinforcement learning. *arXiv preprint arXiv:2101.05950*.