systems

MDPI

Article

Prediction and Visualisation of SICONV Project Profiles Using Machine Learning

Adriano de Oliveira Andrade [1,*], **Leonardo Garcia Marques** [2], **Osvaldo Resende** [3], **Geraldo Andrade de Oliveira** [3], **Leandro Rodrigues da Silva Souza** [3] **and Adriano Alves Pereira** [1]

[1] Centre for Innovation and Technology Assessment in Health, Postgraduate Program in Electrical and Biomedical Engineering, Federal University of Uberlândia, Uberlândia 38408-100, Brazil
[2] Instituto Federal de Educação, Ciência e Tecnologia, Campus Itumbiara, Itumbiara 75524-245, Brazil
[3] Instituto Federal Goiano, Campus Rio Verde, Rio Verde 75901-970, Brazil
* Correspondence: adriano@ufu.br; Tel.: +55-(34)-3239-4046

Abstract: Background: Inefficient use of public funds can have a negative impact on the lives of citizens. The development of machine learning-based technologies for data visualisation and prediction has opened the possibility of evaluating the accountability of publicly funded projects. Methods: This study describes the conception and evaluation of the architecture of a system that can be utilised for project profile definition and prediction. The system was used to analyse data from 20,942 System of Management of Agreements and Transfer Contracts (SICONV) projects in Brazil, which are government-funded projects. SICONV is a Brazilian Government initiative that records the entire life cycle of agreements, transfer contracts, and partnership terms, from proposal formalisation to final accountability. The projects were represented by seven variables, all of which were related to the timeline and budget of the project. Data statistics and clustering in a lower-dimensional space calculated using t-SNE were used to generate project profiles. Performance measures were used to test and compare several project-profile prediction models based on classifiers. Results: Data clustering was achieved, and ten project profiles were defined as a result. Among 25 prediction models, k-Nearest-Neighbor (*kknn*) was the one that yielded the highest accuracy (0.991 ± 0.002). Conclusions: The system predicted SICONV project profiles accurately. This system can help auditors and citizens evaluate new and ongoing project profiles, identifying inappropriate public funding.

Keywords: accountability; machine learning; t-SNE; PCA; BPMN; SICONV; MLR3

check for updates

Citation: de Oliveira Andrade, A.; Marques, L.G.; Resende, O.; Andrade de Oliveira, G.; Rodrigues da Silva Souza, L.; Alves Pereira, A. Prediction and Visualisation of SICONV Project Profiles Using Machine Learning. *Systems* **2022**, *10*, 252. https://doi.org/10.3390/systems10060252

Academic Editor: William T. Scherer

Received: 2 November 2022
Accepted: 7 December 2022
Published: 10 December 2022

1. Introduction

Accountability is defined as the process of demonstrating the use of funds received and transferred to a person (physical or legal) over a specified period and for a specified purpose. In other words, anyone who receives money to perform a particular work project, service, etc., must prove the expenditure of these funds in a document called Accountability, which attests to the fulfilment of the civil obligation [1].

The quality of such accountability influences the decision made when allocating resources to entities [2]. The perceived quality of accountability is determined by several factors, including efficiency (the actions taken by the entity when using the resources provided), stability (the possibility of the activities performed by the entity continuing), reputation, and the amount of information provided by the beneficiary entity [3]. The use of such dimensions can aid in the definition of variables used in the construction of models or systems for the prediction and classification of accountability quality [3].

Recently, Rana et al. [4] presented a review study that identifies the primary needs in the field of financial auditing in light of technological advances, economic constraints, along with social and political contexts. The low number of studies from Asia, Africa, Latin America, and Southern and Eastern Europe is highlighted. According to the authors, financial accountability is a tool that aids in public and private governance, as well as the improvement of public service transparency and delivery. Furthermore, there is a need to employ systems based on machine learning and artificial intelligence to assist in decision-making and assessing public-sector performance [5].

This justifies the appearance of several studies in the area. For instance, Sun and Sales [6] proposed the use of Artificial Neural Networks (ANN) for predicting public procurement irregularities based on the characteristics of the contractor. Typical irregularities are (i) transparency irregularities; (ii) professional standard irregularities; (iii) fairness irregularities; (iv) contract monitoring and regulation irregularities; (v) procedural irregularities. The authors used several data sets, including SICONV (Portuguese: Sistema de Convênios; English: Federal Government Agreements Management System). In terms of prediction of irregularities an overall accuracy of 80.87% was reported. In another recent study, Zhang [7] proposed the evaluation of a financial audit model based on a Convolutional Neural Network (CNN). The study compared classification results of financial data based on CNN (accuracy of 93.4%) and ANN (accuracy of 90.0%). Table 1 summarizes the purpose and results of other relevant studies that illustrate the use of machine learning in accountability.

Table 1. Relevant studies reporting the use of machine learning in accountability.

Study	Purpose	Type of Irregularity	Main Method	Maximum Accuracy
Mongwe et al. (2021) [8]	Fraud detection	Fairness irregularities	Bayesian logistic regression	75.3%
Khan et al. (2022) [9]	Fraud detection	Fairness irregularities	Beetle Antennae Search (BAS)	84.9%
Jiang and Jones (2018) [10]	Financial distress detection	Stability	Gradient Boosting Model (TreeNet)	94.9%
Zhang (2021) [7]	Financial audit	Procedural irregularities	Convolutional Neural Network	93.4%
Abbasi et al. (2012) [11]	Fraud detection	Fairness irregularities	Meta-learning	80%
Hamal and Senvar (2021) [12]	Fraud detection	Fairness irregularities	Random Forest	93.7%
Bertomeu et al. (2020) [13]	Misstatements	Financial data	Random Under-Sampling Boost (RUSBoost)	76.3%
Yang Bao et al. (2020) [14]	Fraud detection	Fairness irregularities	Random Under-Sampling Boost (RUSBoost)	71.7%
Zhang (2021) [15]	Management accounting information	Decision-making	Artificial Neural Network (ANN)	100%
Song et al. (2014) [16]	Fraud detection	Fairness irregularities	Ensemble of classifiers	84.5%
Papík and Papíkova (2022) [17]	Fraud detection	Fairness irregularities	Neural Network (NN)	90.8%
Chen and Zhang (2022) [18]	Financial crisis	Irregular accounting information	Artificial Neural Network (ANN)	90.0%
Li (2022) [19]	Parallel bookkeeping	Connection of Financial Accounting and Budget Accounting	Deep Neural Network	87.7%

Table 1. *Cont.*

Study	Purpose	Type of Irregularity	Main Method	Maximum Accuracy
Liu (2022) [20]	Financial Accounting Quality	Financial quality indicators	Dynamic Neuron Model	98%
Mongwe et al. (2021) [8]	Financial audit	Fraud and weak corporate governance	Bayesian logistic regression with automatic relevance determination (BLR-ARD)	73%
Cecchini et al. (2010) [21]	Fraud detection	Fairness irregularities	Support vector machines using the financial kernel (SVM-FK)	87.8%
Kuzey et al. (2019) [22]	Factors influencing cost system functionality	Cost data management process	Decision tree algorithm C5.0 (DT-C5.0)	91.5%

Transparency is fundamental in accountability processes. The use of machine learning methods has helped automate this process, primarily with regard to the identification of failures in the monitoring or evaluation of processes. From the perspective of accountability in the public sector, it is necessary that entities provide data so that any interested agent, whether a non-governmental organization or a citizen, can monitor the good use of public resources. Currently, there is a dearth in the provision of public data that are easy to interpret and access, in addition to systems that enable the interpretation of such data in order that society can effectively monitor the quality of accountability [23].

Although there are several studies addressing the use of machine learning to identify fraud (e.g., [24]) and risk assessment (e.g., [25]), there is a lack of studies addressing the application of these methods for assessing the quality of accountability, particularly in the governmental sector [6,26–29]. This study proposes the architecture and organisation of a system to predict SICONV project profiles. To this end, t-Stochastic Neighbor Embedding is used to visualise data in a two-dimensional lower space. Several models based on different classifiers are compared and evaluated using the Machine Learning in R framework (mlr3).

The identification of variables that can assist in the process of accountability is fundamental for the monitoring of projects that receive financial support from private or government entities. In this sense, this study proposes a set of descriptors based on fundamental factors for the evaluation of accountability (i.e., efficiency, regularity and predictability [3]). The descriptors were characterized and quantified from the assessment of 20,942 projects that received financial resources from the Brazilian Government. The similarity between the assessed projects was evaluated by the projection of the descriptors in a space of reduced dimension, from which it was possible to identify groups of projects that have similar profiles. These groups were used to perform the labelling of projects for the construction of classification models that can be used in the monitoring of ongoing projects and in the verification of the quality of finalized accounts.

2. Materials and Methods

Figure 1 depicts the proposed architecture of a system for project profile prediction. Standard Business Process Model and Notation (BPMN, v2.0) was utilised to represent the architecture. According to Tomaskova and Kopecky [30], BPMN can be thought of either as a language for creating business process models or as a standard for modelling business processes. By utilising BPMN, one may consider the unique processes and entities of a system, as well as their interactions. Each macro-process (e.g., a broad and general task) or entity (e.g., a user or a sub-system) is placed in a region (called a lane) delimited by a rectangle, and it is labelled by a vertical text to the left of the rectangle. The beginning and end of each process are denoted by a circle and a thicker circle, respectively. The sequence flow of activities or tasks is represented by solid lines with a filled arrow at the end, whereas data used or produced by the tasks are represented by dotted lines with an unfilled arrow at the end. The formal abstract specification of the system is a major advantage of this type of representation, as it facilitates its reproduction, extension, comprehension, and maintenance. The full specification of BPMN is given by the Object Management Group (OMG) at https://www.bpmn.org/ (accessed on 6 December 2022). All the processes of the system and statistical analysis were carried out in R, which is a language and environment for statistical computing [31].

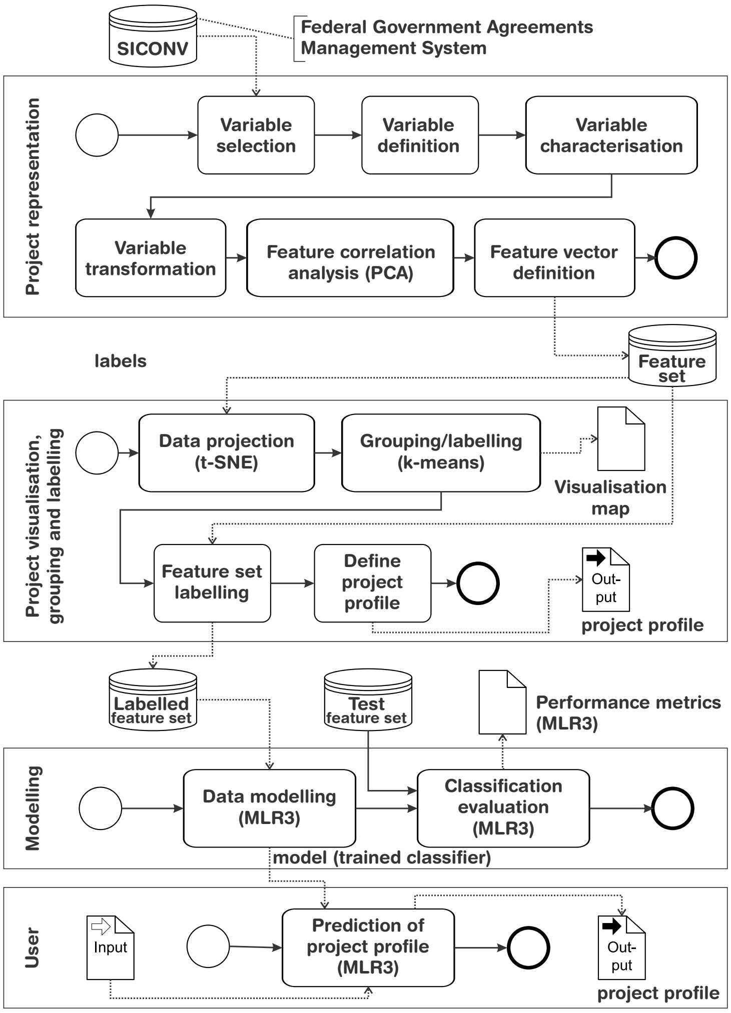
SICONV
Federal Government Agreements Management System
Project representation
Variable selection
Variable definition
Variable characterisation
Variable transformation
Feature correlation analysis (PCA)
Feature vector definition
Feature set
labels
Project visualisation, grouping and labelling
Data projection (t-SNE)
Grouping/labelling (k-means)
Visualisation map
Feature set labelling
Define project profile
Out-put
project profile
Labelled feature set
Test feature set
Performance metrics (MLR3)
Modelling
Data modelling (MLR3)
Classification evaluation (MLR3)
model (trained classifier)
User
Input
Prediction of project profile (MLR3)
Out-put
project profile

Figure 1. System architecture depicting the processes, task, user, and data. The diagram follows the BMPN standard.

2.1. The Database SICONV

The System of Management of Agreements and Transfer Contracts (SICONV) is a Brazilian Government initiative that records the entire life cycle of agreements, transfer contracts, and partnership terms, from proposal formalisation to final accountability [32]. The system prioritises transfer transparency and ensures proper use of scarce public resources. It also demonstrates to society the return on public investments. Although society may have access to several indicators showing how governmental resources are spent, these are publicly presented to the public in a summarised manner. Therefore, to gain access to the data set (SICONV) used in this study, it was necessary to establish a formal agreement with the Ministry of Agriculture, Cattle and Supplying (MAPA, in Portuguese, Ministério da Agricultura, Pecuária e Abastecimento).

2.2. Project Representation

The primary objective of this process is to represent the selected SICONV projects with a multidimensional feature vector that possesses variables associated with accountability-relevant factors, i.e., efficiency, regularity, and predictability. Efficiency can be defined as the capacity to complete a project without wasting resources and time. Variables that contrast planned actions with actual results can have an impact on efficiency. Regularity is the quality of having parts that are evenly or symmetrically distributed. From a temporal standpoint, regularity refers to events that occur frequently or with the same intervals of time between occurrences. Regularity can be affected by variables related to the planning of actions in a project. Predictability is the quality of possessing prior knowledge of an event or action. Variables that contribute to greater regularity may contribute to greater predictability, whereas the more irregular the behaviour of the variable the more it contributes to a decreased predictability.

2.2.1. Variable Selection

The variable selection task (Figure 1) was a qualitative task that required the collaborative analysis of two researchers in order to examine and select variables from SICONV that could be used in this study. This selection possessed the following practical requirements and restrictions: (i) the choice of the variables should be guided by accountability-relevant factors (i.e., efficiency, regularity, and predictability), meaning that only variables that could influence these factors could be selected; (ii) variables considered sensitive toward revealing personal project information were avoided; (iii) there was a preference for simplicity, meaning that the selected variables can be easily understood even by the lay person, and potentially can be public and open.

2.2.2. Variable Definition

The variables defined by the variable definition task are displayed in Table 2. The accountability factor that influenced the selection of each variable is indicated. Variables that favour positive regularity and predictability were associated with these factors, whereas variables sensitive to changes in the course of a project as opposed to predetermined actions were associated with efficiency. For the execution of this study, 20,942 projects that had approved accountability between 2008 and 2021 were included, that is, the data of the selected projects are those of projects that received a favourable opinion, which are considered projects that use public resources in accordance with the standards required by the evaluation agencies.

The project execution-planning period ($N1$) is a factor responsible for structuring the execution of the project, i.e., defining the tools used, execution costs, and step prioritisation over time. The proper scheduling of the project timeline is crucial to its success. The duration of the project's execution can directly affect its regularity and predictability. For instance, temporally well-defined project tasks contribute to the temporal regularity in the execution of the project; additionally, an increase in regularity facilitates the predictability of the temporal execution of specific project phases.

The actual duration of project execution ($N2$) is a determinant of whether the defined operating flow was sufficient to achieve the desired results within the allotted timeframe. In other words, this indicates whether the results were obtained within the specified time frame. This variable has a direct effect on efficiency, as it enables the detection of discrepancies between planned and actual actions over time.

The total value of the contract ($N3$) is a financial factor responsible for determining the contribution of all authors, government and applicant of the proposal into the budget of the project over the planned time. In addition, this value determines the expected cost and the amount of resources that will be allocated at each stage of the project. The financial planning of the proposal contributes to the regularity and predictability of the results of the project.

The value of the government contribution ($N4$) is a financial factor that represents the government's budget, in terms of costs, destined for the project. This amount corresponds to the quantity that the government estimates it will inject into the project over the planned time. The regularity and predictability of the results of the project can be attributed, in part, to the complete financial planning that was implemented for the proposal.

The amount returned ($N5$) at the conclusion of the contract represents the difference between the financial resources used and those provided at the beginning of the project. In regards to the use of resources in the public sector, the return of financial resources cannot always be viewed as positive, as the returned resources may be subject to restrictions on reuse under certain circumstances. This variable influences efficiency, as inefficient use of financial resources can lead to resource waste.

Legally, administrative contracts can be changed. These changes are formalised by the addendum term, or additive term ($N6$). The additive term can be used to make additions or deletions on the object, renegotiations, and other legal modifications that are considered contract changes. For miscellaneous contracts (provision of discontinued services and supplies), which may be delayed in their execution schedule and impose the need for their extension, there arises the need to formalize a specific additive term for extension ($N7$). All these variables influence the efficiency concerning the execution of the project.

Table 2. Definition of the variables selected from the SICONV database. The selection of the variables was guided by the presented accountability factors.

Variable	**Definition**	**Accountability Factor**
$N1$	Period (in days) planned for the execution of the project	Regularity and Predictability
$N2$	Period (in days) effectively used for the project execution	Efficiency
$N3$	Total amount of the agreement (R$) considering the amount of government contribution and the counterpart of the applicant	Regularity and Predictability
$N4$	Government contribution amount (R$)	Regularity and Predictability
$N5$	Amount returned (R$) at the end of the agreement	Efficiency
$N6$	Number of additive terms	Efficiency
$N7$	Number of extensions	Efficiency

2.2.3. Variable Characterisation

The variables under investigation were characterised statistically by their range, central tendency, and spread. The range was determined using minimum values (min), maximum values (max), the 25th percentile (1st quartile, q_{25}), and the 75th percentile (3rd quartile, q_{75}). The central tendency was determined by calculating the mean and median. The spread was measured using standard deviation (sd), interquartile range (IQR), and median absolute deviation (mad).

The probabilities of each variable were calculated. The function density was employed to estimate the probabilities of each variable, by using a 512-point Gaussion kernel function. The function *density* [31] was employed. The probabilities were calculated as the integral of the region bounded by the main peaks of the distribution. For peak detection, the function *findpeaks* [33] was used. A region around a peak was defined by where the peak begins and ends in the sense of where the pattern starts and ends.

2.2.4. Variable Transformation

The logarithmic transformation in base 10 was applied to each variable due to the fact that the variables were in different units and some variables had a large range. The transformed variables were characterised by their boxplot, histogram, density, and empirical cumulative density function (ecdf). Using the Kolmogorov–Smirnov test, pairs of ecdfs were statistically compared. A p-value level of 0.05 was adopted, indicating that test results with p-values below this threshold provide sufficient evidence to conclude that the sample data do not come from the same distribution. To implement logarithmic transformation, variables with values of zero were replaced with 0.1.

2.2.5. Feature Correlation Analysis

Principal Component Analysis (PCA) is a multivariate technique that can be used for dimension reduction, feature extraction, correlation analysis, and data filtering [34]. The primary objective of utilising PCA, in this study, was to estimate the directions of the studied variables in the principal component (PC) space, so that correlations between variables could be evaluated along distinct dimensions. This is depicted on the loadings plot, a graph of the direction vectors that define the model. The loadings plot illustrates how the original variables contribute to the creation of the principal component. Positively correlated variables will appear adjacent to one another on a loadings plot of, for example, p1 versus p2, whereas negatively correlated variables will appear diagonally opposite one another. PCA was estimated by using the function *prcomp* [31] and the variables were scaled to possess unit variance before analysis. The function *ffviz_pca_var* [35] was used to visualise the direction of the variables in the PC space. The scree plot, estimated with *fviz_eig* [35], was employed to verify the cumulative variance explained by each principal component and hence to define the number of components to be retained for the visualisation of loadings plots.

Hartmann and Waske [36] provided the following interpretations of the original variables as vectors, which are reproduced below for clarity and completeness:

- The more parallel a vector is to a PC axis, the more it contributes to that PC.
- The longer the vector, the more variability of this variable is represented by the two principal components displayed.
- Small angles between vectors indicate high positive correlation, right angles indicate no correlation, and opposite angles indicate high negative correlation.

2.2.6. Feature Vector Definition

Each project was represented by a seven-dimensional feature vector using logarithmic in base 10 transformed variables (from $N1$ to $N7$). The feature vectors were stored in an R data frame (feature set, Figure 1) for further analysis.

2.3. Project Visualisation, Grouping and Labelling

The objective of this process (i.e., project visualisation, grouping and labelling, Figure 1) is to visualise the relatively high-dimensional data in $\mathbb{R}^7$ in a lower-dimensional space ($\mathbb{R}^2$), cluster these data in the two-dimensional space and then label the projects according to the clustering results. There are several strategies for mapping high-dimensional data in a lower-dimensional space, including PCA; however, according to Maaten and Hinton [37], the use of t-Stochastic Neighbor Embedding (t-SNE) is capable of capturing much of the local structure of the high-dimensional data, while also revealing global structure such as the presence of clusters at several scales. In R, there is an efficient implementation of t-SNE (*Rtsne*) [38], which was used in this research. Prior to data projection, the high-dimensional data were standardized. Data projection with t-SNE requires the setting of cost function (perplexity, $Perp$) and optimization parameters (number of iterations, T, learning rate, η, and moment, $\alpha(t)$). The perplexity can be interpreted as a smooth measure of the effective number of neighbours. Typical values of $Perp$ are between 5 and 50 [37]. In this study, all default values of *Rtsne* were used, i.e., $Perp = 30$, $\eta = 200$ and α varied as a function of T from 0.5 to 0.8. The number of iterations T was set to 1000.

Hartigan and Wong [39] describe the k-means clustering algorithm in detail, which is implemented in R as the *kmeans* function. The k-means algorithm divides M points in N dimensions into k clusters with the goal of minimising the within-cluster sum of squares. The input data for the k-means was the two-dimensional data yielded by t-SNE, the number of centres was set to 10 and the maximum number of iterations to 10. The function *fviz_cluster* [35] was employed for the visualisation of the results of k-means. The labelling of the data points (cluster membership) was based on the minimum Euclidean distance between the cluster centres to the observation. For the characterisation of each cluster, the median of the values each variable (from $N1$ to $N7$), according to the distinct clusters, was estimated. The loadings plots based on the two PCs that captured the largest data variability (based on PCA, as previously described) was estimated for each cluster. The aim was to characterise the correlation of the variables according to the obtained clustering.

2.4. Definition of the Project Profile

Project profile was defined in terms of the clustering results. First, the median value for each variable (med_k^{var}), specific to a cluster k, was estimated to define the set $Q_{var} = \{med_k^{var} \mid 1 \leq k \leq 10 \in \mathbb{N}\}$, in which Q_{var} is the set of median (*med*) statistics for each variable $var = \{N1, N2, N3, N4, N5, N6, N7\}$. Secondly, specific quantiles (i.e., 0%, 25%, 50%, 75% and 100%) of Q_{var} were estimated, in which 0% and 100% are the minimum and maximum values of Q_{var}, respectively. Finally, each variable was qualified by an ordinal scale (i.e., low, medium and high) according to distinct clusters and quantiles.

2.5. Modelling

The goal of data modelling was to create a model that could predict SICONV project profiles based on the set of variables ($N1$, $N2$, $N3$, $N4$, $N5$, $N6$, and $N7$). The recently developed framework [40], called Machine Learning in R (*mlr3*), was used for this purpose. *mrl3* is a generic, object-oriented, and extensible framework for the R language that can be used for classification, regression, and other machine learning tasks. The data were organized in an R data frame with 20,942 observations (i.e., projects) and seven variables (i.e., $N1$, $N2$, $N3$, $N4$, $N5$, $N6$, and $N7$) and a label for each observation according to the clustering results. The variables were transformed by a logarithmic transformation in base 10.

mrl3 implements all necessary steps to create and test classification models. The first step for creating a classification model is to define a *Classification Task*, which encapsulates the data with meta-information, such as the name of the prediction target column.

A model should be created by a *Learner*, which is a classifier method, according to the *mlr3* terminology. Currently, there are 137 *Learners* available [41] (https://mlr3extralearners.mlr-org.com/articles/learners/listlearners.html, accessed on 6 December 2022), but not all of these are suitable for modelling multi-class problems, as in this study. As a result, only classifiers capable of dealing with multi-class problems were chosen. In total, the performance of 25 classifiers was evaluated. The classifier parameters used are the toolbox defaults, which are fully described in the manual [41]. A general description of these classifiers is given in Table A1.

The data set was split into training (80%) and test sets (20%), i.e., $ratio = 0.8$. The function *rsmp* employed the *subsampling* method to split data repeats ($repeats = 20$ times) into a training and test set with a ratio of 0.8. The model training was obtained through *resampling*. The training of all models based on distinct classifiers was obtained by applying the function *classif* to each classifier defined in *classifier.list*. This can be be performed without difficulty through use of a functional language such as R (see Appendix B).

The performance of the classifiers was evaluated by using the package *mlr3measures* [42], which implements several performance measures for supervised learning. The following performance metrics for multi-class tasks were employed:

- Classification Accuracy (*acc*): *acc* is defined in Equation (1), in which n is the number of observations, i is the i-th observation, w_i is the weight and $[P]$ is a function using the Iverson bracket notation (Equation (2)), which holds a value of one if the target (t_i) equals the response (r_i) and zero, otherwise. In this study, w is normalized such that the sum of its values is one, and w_i has the same value for all observations and assessment metrics.

$$acc = \frac{1}{n}\sum_{i=1}^{n} w_i[t_i = r_i],\ acc \in [0,1] \tag{1}$$

$$[P] = \begin{cases} 1, & \text{if } P \text{ is true,} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

- Balanced Accuracy (*bacc*): *bacc* computes the weighted balanced accuracy, suitable for imbalanced data sets. Equation (3) defines *bacc*, in which y_i is the class of the i-th observation and y_j is the class of the j-th observation, of a multi-class problem with k classes.

$$bacc = \frac{1}{\sum_{i=1}^{n}\hat{w}_i}\sum_{i=1}^{n}\hat{w}_i[t_i = r_i],\ bacc \in [0,1] \tag{3}$$

$$\hat{w}_i = \frac{w_i}{\sum_{j=1}^{n}[y_j = y_i]w_i} \tag{4}$$

- Classification Error (*ce*): *ce* compares true observed labels with predicted labels in multi-class classification tasks. Equation (5) defines *ce*.

$$ce = \frac{1}{n}\sum_{i=1}^{n} w_i[t_i \neq r_i],\ ce \in [0,1] \tag{5}$$

- Log Loss (*logloss*): *logloss* compares true observed labels with predicted probabilities in multi-class classification tasks. This is defined in Equation (6), in which p_i is the probability for the true class of observation i.

$$logloss = -\frac{1}{n}\sum_{i=1}^{n} w_i \log(p_i),\ logloss \in [0,\infty[\tag{6}$$

- Multi-class Brier Score (*mbrier*): *mbrier* compares true observed labels with predicted probabilities in multi-class classification tasks. *mbrier* is defined in Equation (7), in which I_{ij} is 1 if observation i possesses the true label j, and 0 otherwise.

$$mbrier = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}(I_{ij} - p_{ij})^2,\ mbrier \in [0,2] \tag{7}$$

In addition to the aforementioned multi-class metrics, the following binary metrics were estimated for each class of the classifier that yielded the best prediction performance:

- *recall*: it is also called true positive rate or sensitivity. This is defined in Equation (8), in which TP is the number of true positives and FN the number of false negatives.

$$recall = \frac{TP}{TP + FN},\ recall \in [0,1] \tag{8}$$

- *specificity*: it is also called true negative rate. This is defined in Equation (9), in which TN is the number of true negatives, FP the number of false positives and TN the number of true negatives.

$$specificity = \frac{TN}{FP + TN},\ specificity \in [0,1] \tag{9}$$

- F-beta Score (*fbeta*): *fbeta* compares true observed labels with predicted labels in binary classification tasks. *fbeta* is defined in Equation (10). In this study $\beta = 1$, which is a measure called the $F1$ score.

$$fbeta = \left(1 + \beta^2\right)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall},\ fbeta \in [0,1] \tag{10}$$

$$precision = \frac{TP}{TP + FP} \tag{11}$$

3. Results and Discussion

This study presented a system architecture for predicting SICONV project profiles (Figure 1). The system specification was developed using BMPN to facilitate the comprehension and execution of its architecture. As noted by Rana et al. [4], the examination of data such as the SICONV data set contributes to alleviating the lack of studies in this field in Latin America by analysing data originating from the public sector. The inclusion of data visualisation in a two-dimensional space, using t-SNE, and the labelling of the data through the use of a clustering method (i.e., *k*-means) is a significant difference between the proposed architecture and those reported in previous studies [9,24,25]. Data visualisation permits the visual examination of similarities between the profiles of diverse projects. Unsupervised labelling allows for the labelling of data in the original data space while taking into account the clustering results of the data analysis in the lower-dimensional space.

The variables of the SICONV database (Table 2) were chosen based on accountability criteria (i.e., regularity, predictability, and efficiency) and, as emphasised by Laat [23], the need to choose variables that can be made available to the general public. The fact that these variables are simple to comprehend and do not contain sensitive information contributes to the transparency of the project profiles. In this study, a large number of projects were reviewed (20,942), and as their accountability was approved by auditing agencies, these become a valuable resource for establishing typical monitoring profiles for new and current projects.

Table 3 and Figure 2 present the characterisation of the investigated variables in terms of basic statistics and probability. The statistical analysis of the data (Table 3) reveals that the majority of variables displayed a very high degree of variability, indicating variances between project profiles. The low variability of the variable $N7$ (number of project extensions) indicates that project extensions are not usual for SICONV projects. In practice, if a project has a high $N7$, the auditing agency should assess it carefully. Figure 2 depicts the probabilities associated with the values of each variable. This can be used to uncover errors in accountability as well as atypical projects.

Figures 3 and 4 depicts the boxplot, histogram, density and empirical cumulative distribution functions of the logarithmic transformed variables. The application of the Kolmogorov–Smirnov Test for pairs of variables yielded p-values less than 0.05 for all comparisons. As the primary objective of this research was to develop a prediction model for the SICONV projects, it was important to logarithmically transform the variables to limit the impact of their units and range on the model. Figure 3 depicts the boxplot, histogram, and density of the variables that have been logarithmically transformed. As expected, the transformed values of the variables fell inside a compatible range. Additionally, the density of each variable exhibits multiple peaks, indicating the existence of diverse project profiles. Each empirical cumulative distribution function (ecdf) of the variable is depicted in Figure 4. Although there was no statistical equivalence between pairs of ecdfs, the ecdfs of some variables have similar shapes, such as $N1$ and $N2$ (temporal variables) and $N3$ and $N4$ (financial variables). The interpretation of the ecdf is straightforward, for example, the ecdf of $N7$ indicates that 87.5% of projects do not request extension for variable $N7$ (values for which the logarithm, at base 10, are less than 0). An advantage of presenting ecdfs of various transformed variables in a single plot (i.e., Figure 4) is the ability to compare their behaviour in terms of value distribution, as well as to compare data from a specific project (not used in the estimations) against reference measurements (used to generate the ecdfs).

Table 3. Main statistics of the studied variables.

Type	Statistic	*N*1 (Days)	*N*2 (Days)	*N*3 (R$)	*N*4 (R$)	*N*5 (R$)	*N*6 (Quantity)	*N*7 (Quantity)
Range	min	14	24	15,306	15,000	0.1	0.1	0.1
	max	1968	4019	72,482,484	69,527,434	6,115,575	22	6
	q_{25}	377	565	127,909.2	107,250.0	0.10	1.0	0.1
	q_{75}	730	1,066	315,000.0	292,500.0	412.75	2.0	0.1
Centre	mean	599.189	865.338	330,704.1	293,912.6	3,173.833	1.839	0.264
	median	547	735	205,000	193,621.5	8	1	0.1
Spread	sd	270.908	445.233	1,091,178	971,204.6	60,592.85	1.9	0.465
	IQR	353	501	187,090.8	185,250	412.65	1	0
	mad	266.868	312.829	130,295.3	138,803.2	11.713	1.334	0

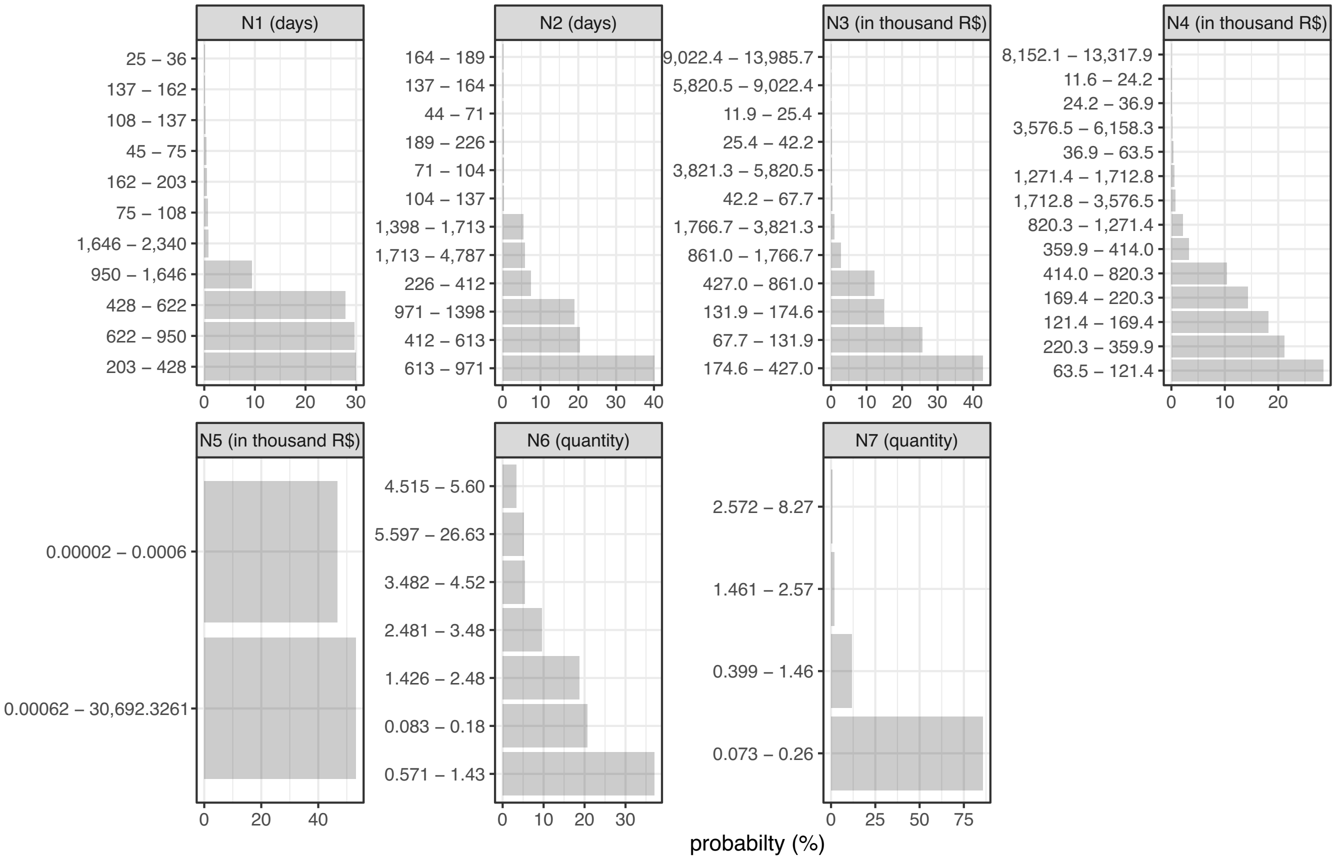

Figure 2. Probability of values for each variable.

The retained dimensions of PCA were selected according to the percentage of explained variances (Figure 5). PCA and the analysis of loadings plots were used to examine the correlation between variables (Figure 6). The capacity to find correlations between variables according to distinct principal component dimensions that account for the majority of data variability is an advantage of PCA. The scree plot depicted in Figure 5 demonstrates that the data variability was represented by the first five principal components; hence, it makes sense to assess the correlation between variables in each space formed by pairs of principal components. Several variables (e.g., *N*1 and *N*6) show a high correlation in the space described by dimensions 1 and 2 but are uncorrelated in other dimensions (e.g., *N*1 and *N*6 in the dimension 2 versus dimension 3 plot). This occurs due to the variety of project profiles contained in the SICONV database. The key conclusion drawn from the data depicted in Figure 6 is that none of the evaluated variables could be eliminated from the analysis since they display complementary information in the sense that they are not absolutely correlated in all assessed loadings plots.

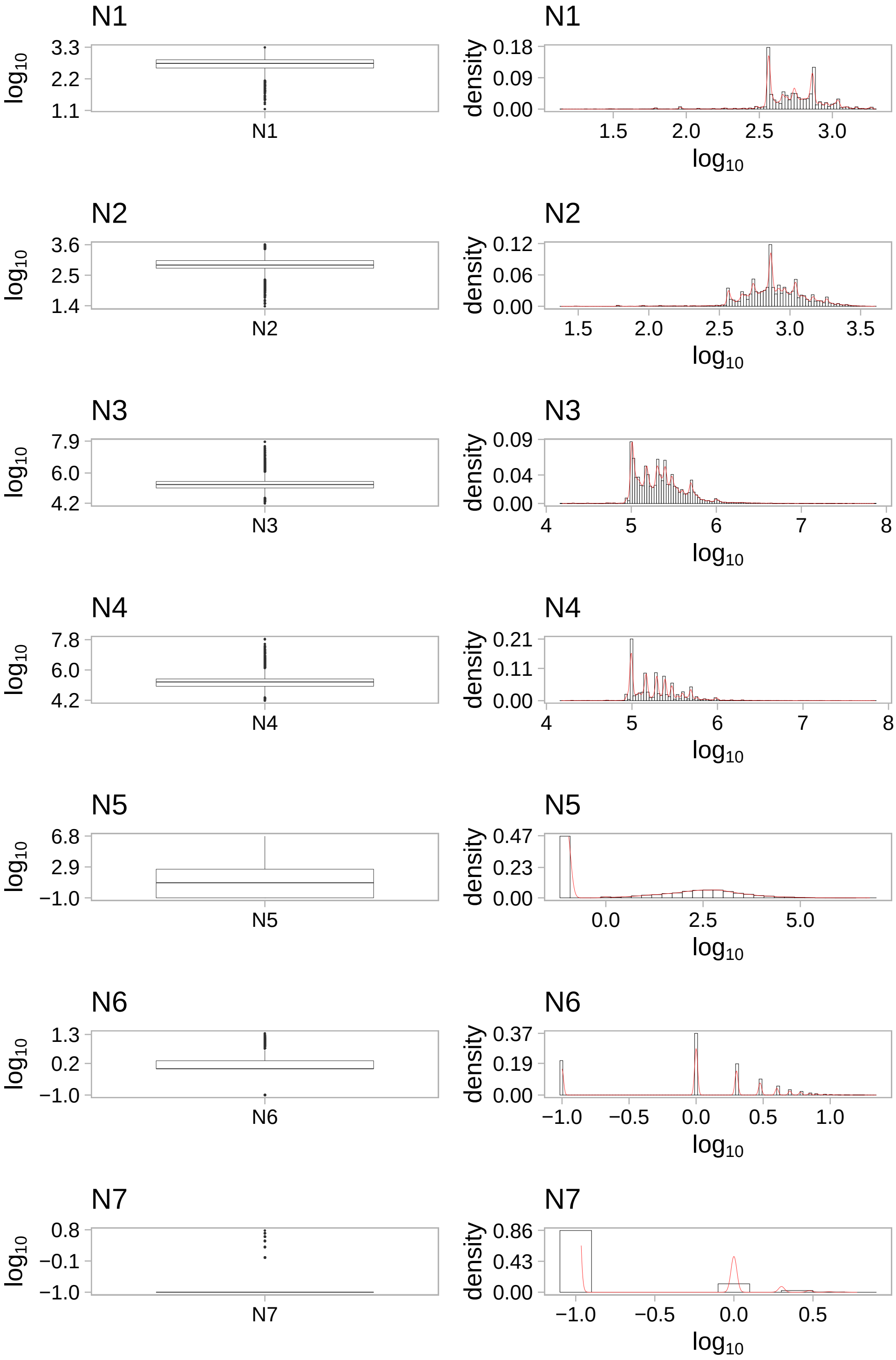

Figure 3. Boxplot of the transformed variables.

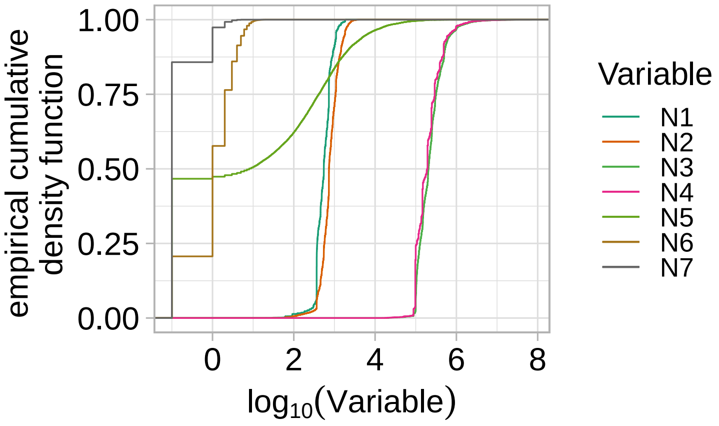

Figure 4. Empirical cumulative distribution functions of the logarithmic transformed variables.

The visualisation and clustering of SICONV projects are shown in Figure 7. Reviewing the literature, we found a lack of reporting on data visualisation in a lower-dimensional space (e.g., two-dimensional space). Data visualisation facilitates comprehension of the proximity between data points (e.g., projects represented in their feature space). This is why t-SNE (Figure 7) was utilised for data visualisation. We also attempted to visualise the data in a two-dimensional space using PCA, but the method could not produce a clear, non-overlapping distribution of data points. Taking into account the careful visualisation of the data points in the t-SNE space and the presence of many peaks in the distributions depicted in Figure 3, the data were grouped with the formation of 10 distinct clusters. Although a variety of approaches exist in the literature to estimate the optimum number of clusters in a data set, their results vary, and in the end, it is always practical to consider a meaningful definition of a cluster for a particular application.

Table 4 presents the median of each variable according to clustering results. The loadings plots for each cluster are shown in Figure 8. The direction of each variable for distinct clusters according to different quadrants of the loading plots are given in Table 5.

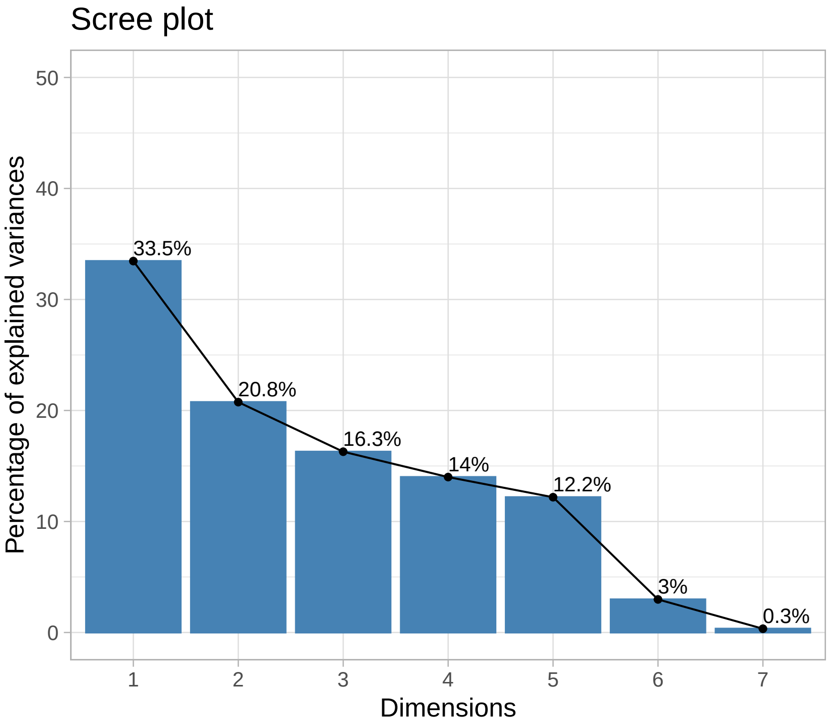

Figure 5. Scree plot.

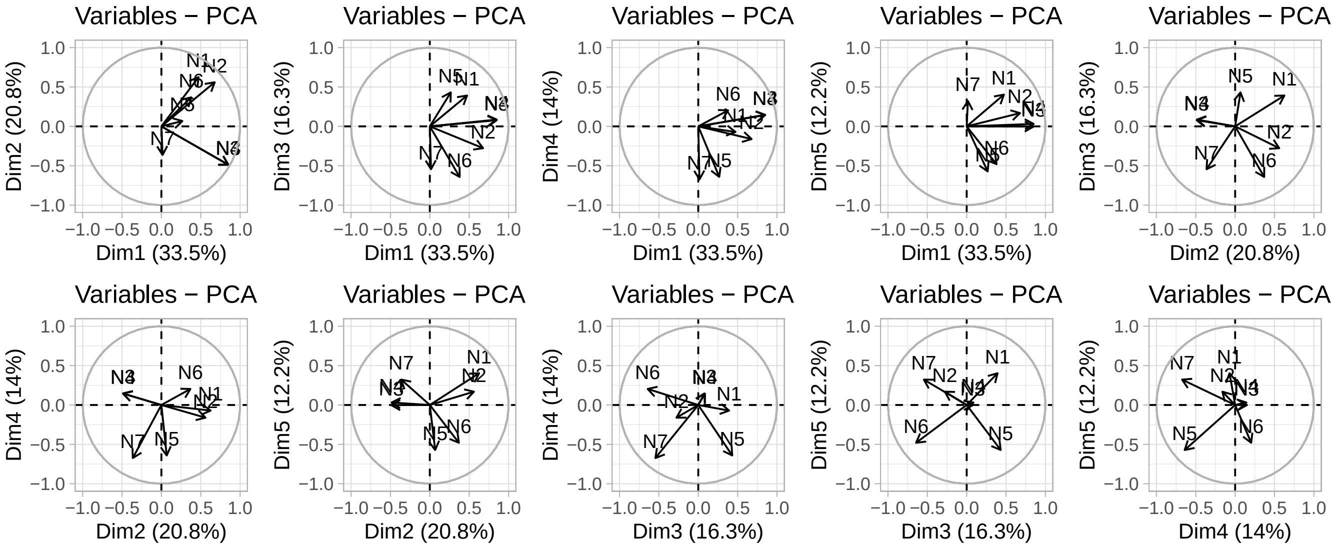

Figure 6. Loadings plots showing the correlation and directions of distinct variables according to PC dimensions.

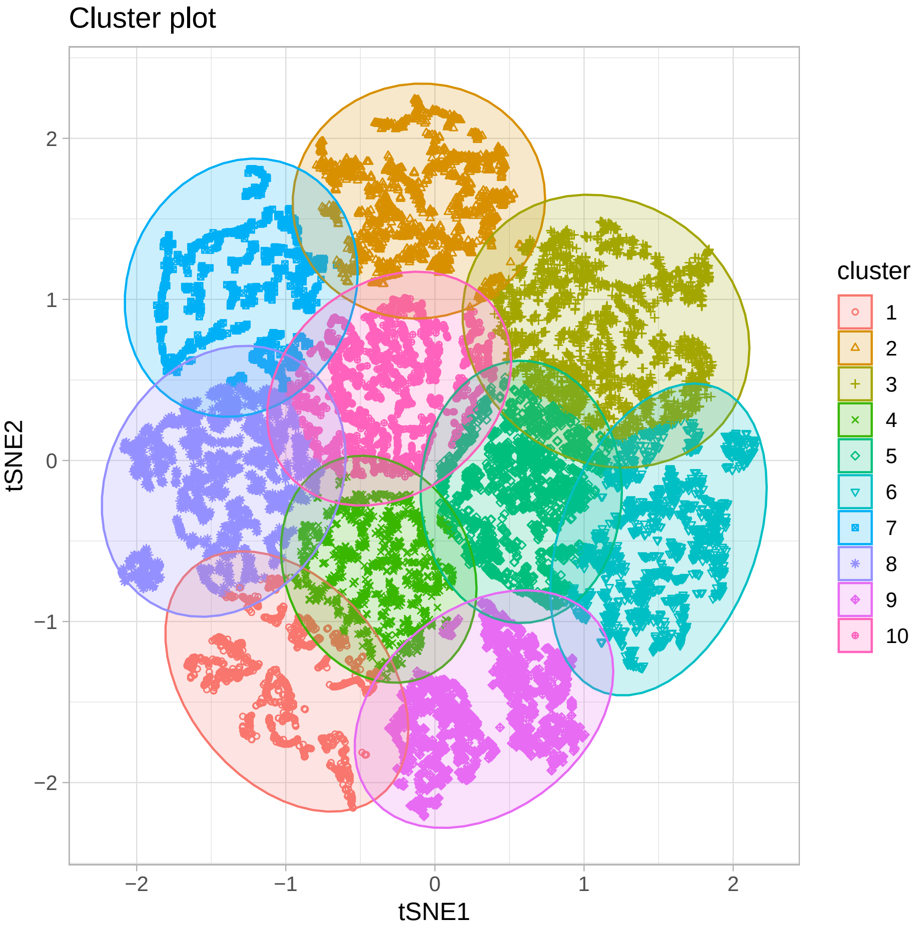

Figure 7. Results of the projection of the high-dimensional data into the lower bi-dimensional space given by t-SNE. Each observation represents a project (20,948 in total). The data points were clustered by k-means ($k = 10$) and the data points which belong to the same group are in the same ellipsoidal coloured region.

Table 4. Median of each variable according to distinct clusters shown in Figure 7. Values equal to zero are replaced by 0.1.

Cluster	***N1* (Days)**	***N2* (Days)**	***N3* (R$)**	***N4* (R$)**	***N5* (R$)**	***N6* (Quantity)**	***N7* (Quantity)**
1	366	395	180,000	146,250	0.1	0.1	1
2	716	718	202,000	195,000	227	0.1	0.1
3	546	749	333,000	292,500	753	2	0.1
4	669	730	153,000	146,250	0.1	1	0.1
5	534	707	118,000	100,000	241	2	0.1
6	940	1044	210,000	195,000	289	1	0.1
7	576	670	209,580	195,000	0.1	0.1	0.1
8	453	748	425,000	390,000	0.1	2	0.1
9	444	1002	242,406	200,000	2	2	1
10	456	815	144,837	117,000	0.1	2	0.1

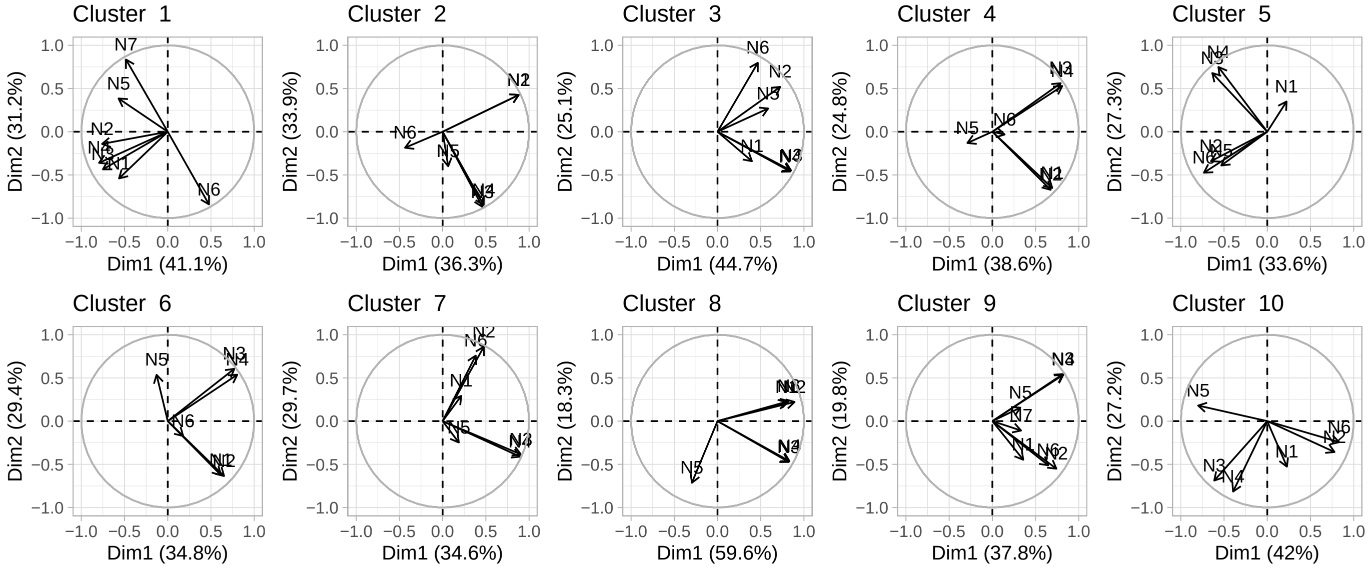

Figure 8. Loading plots for each cluster.

Table 5. Direction of each variable for distinct clusters, according to its positioning on the quadrant, q, is given in Figure 8. q_1, q_2, q_3, and q_4 are the first, second, third, and fourth quadrants, respectively.

Cluster	q_1	q_2	q_3	q_4
1		*N7*, *N5*	*N2*, *N4*, *N3*, *N1*	
2	*N2*, *N1*		*N6*	*N5*, *N3*, *N4*
3	*N5*, *N2*, *N6*			*N1*, *N4*, *N3*
4	*N4*, *N3*		*N5*	*N2*, *N1*, *N6*
5	*N1*	*N4*, *N3*	*N2*, *N6*, *N5*	
6	*N4*, *N3*	*N5*	*N1*, *N2*, *N6*	
7	*N1*, *N2*, *N6*			*N5*, *N4*, *N3*
8	*N2*, *N1*, *N6*		*N5*	*N3*, *N4*
9	*N5*, *N4*, *N3*			*N1*, *N6*, *N2*, *N7*
10		*N5*	*N3*, *N4*	*N1*, *N2*, *N6*

The project profiles (Figure 9) were obtained by the estimated quantiles (Table 6) and categorization (Table 7) of variables. Taking into account the clustering results (Figure 7), the median of each variable was calculated for each cluster (i.e., project profile). The outcomes are shown in Table 8. Table 5 displays the estimated direction of each variable based on clustering. This facilitates the comprehension of how distinct variables correlate with respect to unique project profiles. For the construction of project profiles in terms of qualifiers (e.g., low, medium, and high), the quantiles of each variable were estimated (Table 7) and the variable values were then categorised as indicated in Table 7. On this basis, the project profiles depicted in Figure 9 were defined.

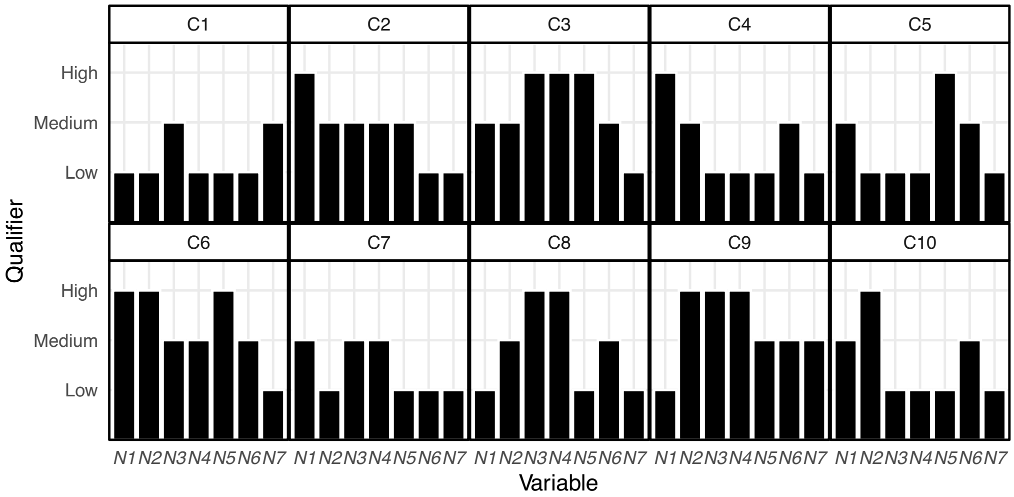

Figure 9. Definition of project profile according to the data presented in Tables 4 and 7.

Table 6. Quantiles for each variable estimated from Table 4.

Variable	0%	25%	50%	75%	100%
*N*1	366.00	453.75	540.00	645.75	940.00
*N*2	395.00	709.75	739.00	798.50	1,043.50
*N*3	118,000.0	159,750.0	205,790.0	234,304.1	425,000.0
*N*4	100,000	146,250	195,000	198,750	390,000
*N*5	0.10	0.10	1.05	237.50	753.00
*N*6	0.100	0.325	1.500	2.000	2.000
*N*7	0.1	0.1	0.1	0.1	1.0

Table 7. Categorization of the variable values based on the quantiles shown in Table 6. Reverse brackets indicate open intervals.

Variable	Low	Medium	High
*N*1	≤453.75	[453.75, 645.75]	>645.75
*N*2	≤709.75	[709.75, 798.50]	>798.50
*N*3	≤159,750	]159,750, 234,304.1]	>234,304.1
*N*4	≤146,250	[146,250, 198,750]	>198,750
*N*5	0.1	[0.1, 237.50]	>237.50
*N*6	≤0.325	[0.325, 2.00]	>2.00
*N*7	0.1	[0.1, 1.0]	>1.0

The prediction of project profiles for distinct classifiers and according to different metrics are provided in Table 8. The classification performance for the most accurate model (classif.kknn) is given in Table 9. In this study, the successful classification of the data (Tables 8 and 9) demonstrates that the assumption that the data could be grouped into ten categories (i.e., project profiles) was accurate. However, more approaches to data stratification could be assessed.

For data modelling and prediction, the *mlr3* framework was employed. As *mlr3* is a new and publicly available tool, it is vital that knowledge based on its use is spread. In addition to illustrating the level of tool abstraction, the code given in this work facilitates comprehension of the key phases involved in data modelling and prediction. This study employs classifiers that are capable of handling multi-class problems and are available in *mlr3*. Table A1 provided a summary of each applied classifier, in order that the reader may gain an understanding of their underlying assumptions. The results indicated that the *kknn* classifier produced the best results with the lowest variability when compared to the other classifiers that were examined using multi-class metrics. The accuracy of the models is comparable to that reported in various other studies (Table 1). As the *kknn* classifier exhibited the highest degree of accuracy, binary evaluation metrics for it were calculated (Table 9). The results showed outstanding performance across various project profiles (i.e., classes). Based on the set of input variables (from $N1$ to $N7$), these models can be used to predict project profiles. A further application of this study would be the creation of an online graphical interface, through which citizens and auditors could enter input variables and the system would go on to estimate the expected profile. Such information is required for the follow-up of publicly funded projects. Finally, for the classifiers that yielded a low performance, an additional study could be executed with the aim of trying to estimate optimal parameters for these classifiers that could result in the improvement of their performance.

Table 8. Classification performance according to distinct evaluation metrics. The mean and standard deviation are presented. The estimates were obtained from the subsampling method that split the data 20 times into training and test sets with a ratio of 0.8.

Classifier	***acc***	***bacc***	***ce***	***logloss***	***mbrier***
classif.AdaBoostM1	0.245 ± 0.006	0.199 ± 0.001	0.755 ± 0.006	1.774 ± 0.007	0.817 ± 0.002
classif.C50	0.97 ± 0.003	0.971 ± 0.002	0.03 ± 0.003	0.145 ± 0.013	0.054 ± 0.005
classif.catboost	0.1 ± 0.006	0.1 ± 0.006	0.9 ± 0.006	34.539 ± 0.001	1 ± 0.001
classif.ctree	0.96 ± 0.004	0.961 ± 0.004	0.04 ± 0.004	0.349 ± 0.043	0.063 ± 0.005
classif.cv_glmnet	0.871 ± 0.007	0.866 ± 0.007	0.129 ± 0.007	0.417 ± 0.017	0.192 ± 0.007
classif.featureless	0.138 ± 0.005	0.100 ± 0.001	0.862 ± 0.005	29.785 ± 0.169	1.725 ± 0.01
classif.gbm	0.918 ± 0.005	0.917 ± 0.005	0.082 ± 0.005	0.35 ± 0.011	0.146 ± 0.005
classif.glmnet	0.843 ± 0.005	0.83 ± 0.005	0.157 ± 0.005	0.659 ± 0.01	0.294 ± 0.004
classif.IBk	0.978 ± 0.002	0.979 ± 0.002	0.022 ± 0.002	0.211 ± 0.022	0.043 ± 0.004
classif.JRip	0.961 ± 0.004	0.962 ± 0.004	0.039 ± 0.004	0.32 ± 0.032	0.072 ± 0.008
classif.kknn	0.991 ± 0.002	0.991 ± 0.002	0.009 ± 0.002	0.035 ± 0.01	0.016 ± 0.002
classif.lda	0.849 ± 0.005	0.838 ± 0.006	0.151 ± 0.005	0.873 ± 0.053	0.256 ± 0.006
classif.liblinear	0.831 ± 0.006	0.819 ± 0.006	0.169 ± 0.006	0.698 ± 0.01	0.316 ± 0.004
classif.lightgbm	0.139 ± 0.125	0.134 ± 0.121	0.861 ± 0.125	12.976 ± 2.004	1.71 ± 0.25
classif.LMT	0.96 ± 0.003	0.961 ± 0.003	0.04 ± 0.003	0.193 ± 0.026	0.062 ± 0.005
classif.naive_bayes	0.849 ± 0.005	0.842 ± 0.005	0.151 ± 0.005	0.843 ± 0.042	0.227 ± 0.009
classif.nnet	0.593 ± 0.144	0.581 ± 0.151	0.407 ± 0.144	1.064 ± 0.359	0.512 ± 0.133
classif.OneR	0.243 ± 0.005	0.217 ± 0.004	0.757 ± 0.005	26.146 ± 0.16	1.514 ± 0.009
classif.PART	0.968 ± 0.004	0.969 ± 0.004	0.032 ± 0.004	0.562 ± 0.092	0.059 ± 0.007
classif.randomForest	0.976 ± 0.003	0.976 ± 0.002	0.024 ± 0.003	0.12 ± 0.005	0.047 ± 0.003
classif.ranger	0.974 ± 0.003	0.974 ± 0.003	0.026 ± 0.003	0.141 ± 0.004	0.052 ± 0.002
classif.rfsrc	0.977 ± 0.002	0.977 ± 0.002	0.023 ± 0.002	0.074 ± 0.006	0.036 ± 0.003
classif.rpart	0.871 ± 0.006	0.864 ± 0.006	0.129 ± 0.006	0.466 ± 0.017	0.223 ± 0.009
classif.svm	0.961 ± 0.002	0.962 ± 0.002	0.039 ± 0.002	0.108 ± 0.006	0.057 ± 0.003
classif.xgboost	0.928 ± 0.007	0.927 ± 0.007	0.072 ± 0.007	1.178 ± 0.005	0.522 ± 0.002

Table 9. Classification performance according to binary metrics for each class. The presented results are for the classifier *kknn*, which yielded the best prediction performance. The values are between 0 and 1.

Class	$fbeta$ ($\times 10^{-2}$)	$recall$ ($\times 10^{-2}$)	$specificity$ ($\times 10^{-2}$)
1	99.39	99.42	99.95
2	99.56	99.61	99.95
3	99.04	99.30	99.80
4	99.15	99.04	99.93
5	98.63	98.86	99.80
6	98.65	98.33	99.89
7	99.27	99.05	99.96
8	99.20	99.08	99.92
9	99.91	99.99	99.98
10	98.65	98.60	99.83

4. Study Limitations

The limitations of our study are related to the absence of uncertainty and reliability analyses, both of which can be optimized using machine learning algorithms. Machine learning has been used in several areas, including uncertainty and reliability estimates [43,44]. One way of estimating uncertainty was proposed in the study by Peng et al. [43], where the use of Machine Learning in health prognostics was evaluated. According to the authors, the structure of health predictions based on deep learning is composed of two main stages, namely, acquisition of condition monitoring (CM) data from execution to the failure and remaining useful life (RUL) based on deep learning. However, the authors claim that a neglected issue in methods based on deep learning is that the RUL prediction can be affected by various types of prognostic uncertainty and that, the outcome of prognostics without prior knowledge of uncertainties in a method based on deep learning may be questionable. The authors suggest incorporating uncertainty characterisation and inference into deep learning models. Therefore, to contemplate the suggestions, Peng et al. [43] developed a method based on Bayesian deep learning (based on BDL) for health prognostics for the quantification of uncertainty.

Regarding reliability, in the study by Zhang et al. [44], the reliability of a parallel system as a redundancy structure is estimated, in which one of the variables involved is stress. According to the authors, reliability can be determined by classical methods of statistical inference, but Bayesian methods are also widely used. Thus, the authors use Bayesian inference to infer the reliability of a system using the multicomponent stress-strength model under Marshall–Olkin Weibull distribution. However, one of the causes of the divergence in the results between different studies and the reliability of the presented results is the determination of the parameters used in the machine learning algorithms. Zhang et al. [44] proposed a method of data augmentation to determine the parameters used and at the same time improve the quality of the statistical inference. The method is evaluated through the simulation of a dataset, and the authors conclude that the proposed method has a good performance in estimating reliability.

Another limitation of our research is the use of the clustering method (i.e., k-means). The major limitations of k-means, as pointed out by Bandyopadhyay and Maulik [45], are that the method may become stuck at locally optimal values and that the method is dependent on the user specification of the number of clusters generated from the dataset. In this regard, there are several recent methods in the literature that attempt to identify natural clusters in datasets without any background information about the data objects. Nature-inspired metaheuristic optimization algorithms, in particular, have been used in recent times to overcome the challenges of the traditional clustering algorithm in dealing with automatic data clustering [46,47].

5. Conclusions

This investigation proposes the architecture of a system for predicting SICONV project profiles. The set of variables used in this study is relevant for accountability, which can be made available to the public, thus adding to transparency when it comes to the use of public resources. Using data statistics and clustering results of data mapped onto a two-dimensional space, project profiles were introduced. The study of the performance of several prediction models revealed a high degree of accuracy, indicating that the implemented system can be utilised for the monitoring of projects funded by private and government entities.

Since the profiles were calculated based on more than 20,000 projects, their practical application is straightforward, i.e., new and ongoing projects are expected to have one of the 10 project profiles. If this does not occur, an auditor should investigate the underlying cause. In addition, a system could be created to generate alerts whenever a project deviates from one of the potential project profiles. The basis of this system shows itself as being the most successful classifier tested in this investigative study. Considering the accuracy of the classifiers the most accurate methods were k-Nearest-Neighbor (classif.kknn: 0.991 ± 0.002), Random Forest SRC (classif.rfsrc: 0.977 ± 0.002) and Random Forest (classif.randomForest: 0.976 ± 0.003).

Author Contributions: Conceptualization, A.d.O.A. and A.A.P.; methodology, A.d.O.A. and A.A.P.; software, A.d.O.A. and A.A.P.; validation, A.d.O.A. and A.A.P.; formal analysis, A.d.O.A. and A.A.P.; investigation, A.d.O.A. and A.A.P.; resources, L.R.d.S.S.; data curation, A.d.O.A., A.A.P. and L.R.d.S.S.; writing—original draft preparation, A.d.O.A. and A.A.P.; writing—review and editing, A.d.O.A., A.A.P. and L.R.d.S.S.; visualization, A.d.O.A. and A.A.P.; supervision, A.d.O.A., A.A.P. and L.R.d.S.S.; project administration, L.R.d.S.S.; funding acquisition, L.R.d.S.S., L.G.M., O.R., G.A.d.O. and L.R.d.S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Agriculture, Livestock and Supply–MAPA within the scope of the Ministry of Agriculture and in partnership with the Instituto Federal Goiano–IF Goiano, grant number TED 05/2020-MAPA/IF Goiano. The results reported in this research are derived from the subproject "Determination of scores related to agreements processes for comparison of results investment by the Ministry of Agriculture, Livestock and Supply–MAPA". AOA is fellow of CNPq, Brazil (304818/2018-6 and 305223/2014-3). AAP is fellow of CNPq, Brazil (309525/2021-7).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available due to contract restrictions.

Acknowledgments: The authors would like to express their gratitude to Instituto Federal de Educação, Ciência e Tecnologia Goiano (IF Goiano), Centro de Excelência em Agricultura Exponencial (CEAGRE) and Ministério da Agricultura, Pecuária e Abastecimento (MAPA) for their financial support (Project TED 05/2020 - MAPA/IF Goiano), which was critical to the completion of this study.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SICONV	System of Management of Agreements and Contracts of Transfer
BPMN	Business Process Model and Notation
MAPA	Ministry of Agriculture, Cattle and Supply
PCA	Principal Component Analysis
t-SNE	t-Distributed Stochastic Neighbor Embedding
MLR3	Machine Learning in R
R$	the Brazilian real, which is the official currency of Brazil

Appendix A

Table A1. Set of classifiers tested in this study. All the classifiers are available in the *mlr3* framework.

Classifier	Description
classif.AdaBoostM1	AdaBoost generates a set of hypotheses and combines them using weighted majority voting. Training a weak classifier with iteratively updated training data generates hypotheses. This increases the likelihood that misclassified cases will be included in the training data of the classifier. Training data for successive classifiers focus on harder-to-classify cases [48].
classif.C50	The decision tree divides a dataset into smaller subsets. The leaf node represents a decision and each branch represents a value. Classification starts at the root node and is classified according to features. Algorithm C5.0 is derived from algorithm C4.5, which is derived from algorithm ID3. The C5.0 algorithm has the advantage over the ID3 and C4.5 algorithms: speed, better memory usage, smaller decision trees [49].
classif.catboost	Categorical Boosting (catBoost)—CatBoost handles categorical features using binary decision trees as base predictors and different permutations for different steps of gradient boosting. CatBoost is an implementation of gradient boosting. CatBoost is indicated for studies involving categorical and heterogeneous data [50].
classif.ctree	Conditional Inferences Trees (cTREE)—The CTree method recursively partitions the data by performing a univariate division on the dependent variable, just like traditional decision trees. However, the CTree method uses a classical statistical significance test, selecting a division point based on the minimum p-value of all independence tests, between the response variable and each explanatory variable [51].
classif.cv_glmnet	Cross validation Generalized Linear Models With Elastic Net Regularization (cglmnet)—GLMNET can use Lasso or Cyclical Coordinate Descent Algorithms, repeating the cycle to convergence, successively optimizing the objective function on each parameter with the others fixed. GLMNET is a package that fits into linear and similar models generalized by maximum penalized likelihood. It can be used for linear, logistics and multinomial regression. One of GLMNET's main tuning parameters is the regularization penalty, hence GLMNET has a set of values called regularization path. Path is specified by the argument called Lambda. Cvg_lmnet uses cross-validation to optimize the Lambda value [52].
classif.featureless	Featureless—The Featureless classifier uses the distance of objects and ignores all features. Objects are classified according to distances of a subset of training objects. The distances obtained are combined with classifiers that can be linear or non-linear [53].
classif.gbm	Gradient Boosting Machines (GBM)—GBM is used to solve regression and data classification problems. The learning model is based on consecutively fitting new models to provide a more accurate estimate of the response variable. GBM analyses the predictors and chooses the strongest predictors. GBM performance can be improved by using an additional classifier [54,55].
classif.glmnet	Generalized Linear Models with Elastic Net Regularization (glmnet)—Similar to cv_glmnet, but uses a cost-sensitive measure to optimize the lambda value.
classif.IBk	Instance-Bases Learning with parameter k (IBk)—IBk is a k-Nearest-Neighbour classifier and is in the Lazy classifier category. *k* is a value that determines the number of neighbours that are analysed and the outcome is determined by majority vote. The value of K can be selected based on cross-validation. The basic principle of this algorithm is that when the instance is given, the algorithm searches in the training dataset for its closest instance samples, through use, most commonly, of Euclidean distance, which is used to assign the class for the test sample [56].

Table A1. *Cont.*

Classifier	Description
classif.JRip	Repeated Incremental Pruning to Produce Error Reduction (RIPPER)—JRip is an optimized version of IREP. JRip uses propositional rules that can be executed to classify elements; the rules are created through sequential algorithms. The JRip algorithm creates rules for each dataset, considering the features of the evaluated class; subsequently the next class will also be evaluated and measured according to the previous class. This cycle is repeated until the last class is evaluated [57,58].
classif.kknn	k-Nearest-Neighbour—The kNN classifier classifies unlabelled observations, assigning these to the most similar labelled class. When a data point is provided, KNN searches the training dataset for its nearest K samples to the data point, commonly using the Euclidean distance. The parameter k determines how many neighbours will be chosen for the kNN algorithm [59,60].
classif.lda	Linear Discriminant Analysis (LDA)—LDA is used to distinguish two distinct classes through the linear combination of features. This combination can be used for classification or dimension reduction. Through this method it is possible to project a multidimensional data set in only one dimension, resulting in a single feature [61,62].
classif.liblinear	Library for Large-Scale Linear (liblinear)—LibLINEAR is an open source library and uses a coordinate descent algorithm. LibLINEAR supports logistic regression (LR) and linear support vector machines (SVM). LibLINEAR can classify data that can be linearly separated via a hyperplane [63,64].
classif.lightgbm	Light Gradient boosting algorithm (LightGBM)—This algorithm is based on decision tree algorithms. LightGBM is the implementation of Gradient Boosting with Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [65].
classif.LMT	Logistic Model Trees (LMT)—A logistic model tree (LMT) combines a decision tree and linear logistic regression. LMT uses a tree-growing approach called LogitBoost to refine logistic regression models along their corresponding paths. Additive logistic regression modelling by LogitBoost provides a way to build a leaf model from a partial linear model, which is inherited from its ancestor nodes as the tree grows [66].
classif.naive_bayes	Naive Bayesian (naive_bayes)—Naive Bayesian uses the construction of a Bayesian probabilistic model (based on Bayes' theorem). The Naive Bayesian classifier only needs the mean and variance parameters of the variables and assumes that the variables are independent [67].
classif.nnet	Neural Network (nnet)—Neural Network (NN) is a mathematical representation of the networks of neurons with input signals , while generating output constrained to propagate intheforwarddirection. Therefore, feed-forward optimization is necessaryamongseveralalgorithms,from these back propagation (BP) is the most commonly used [67–69].
classif.OneR	One Rule (OneR)—OneR creates a rule for each predictor in the dataset, then selects the rule with the lowest misclassification rate and assumes this rule as a “one rule". To create a rule for a predictor, it builds a frequency table for each predictor against the class. Accordingly, for each predictor the rule is made as follows: for each predictor count how often each class value appears, find the most frequent class, make the rule assign that class to that predictor value, calculate the total error of the rules of each predictor, choose the predictor with the smallest total error [70–72].

Table A1. *Cont.*

Classifier	Description
classif.PART	Regression Partition Tree (PART)— The PART classification method uses the divide and conquer approach. The PART algorithm comes under classification rules, building a partial C4.5 decision tree during each iteration, using the J4.8 classifier technique. The PART algorithm creates rules recursively, then deletes the instances affected by those rules and repeats the process until there are no more instances, the best leaf is turned into a rule [73–76].
classif.randomForest	Random Forest (randomForest)—The random forest is an algorithm that uses a combination of individual tree predictors, building multiple decision trees in the training stage. For each data point, each tree casts a vote for one class and the forest tries to predict the class based on the class that obtained the majority of votes [60,77].
classif.ranger	Random Classification Forest (ranger)—Fast implementation of Random Forest method [78].
classif.rfsrc	Random Forest for Survival, Regression, and Classification (rfsrc)—Random Forest SRC is an implementation of Random Forest for application in Survival, Regression, and Classification citepIshwaran2008.
classif.rpart	Recursive Partitioning (rpart)—RPART is the implementation of Classification and Regression Trees (CART). This is a method that uses a recursive partitioning regression tree. The algorithm creates a large tree and then prunes the tree to a size that has the lowest cross-validation error estimate by evaluating the values of a cost-complexity parameter [79].
classif.svm	Support Vector Machine (SVM)—A support vector machine tries to classify data by a separating hyperplane. In this form, SVM separates the input data into two classes, trying to maximize the distance between the optimal hyperplane and the nearest training pattern [60].
classif.xgboost	eXtreme Gradient Boosting classification (xgboost)—XGBoost is a decision tree set, which consists of a set of classification or regression trees, based on Gradient Boosting, which iteratively calculates the prediction of multiple trees. The process is repeated several times until the accuracy or error is satisfactory. After each iteration, the model learns and adds new information to the set. The final model is a linear combination of hundreds to thousands of trees forming a regression model where each term is a tree [80,81].

Appendix B

The snippet of code below illustrates how a *Classification Task* is created, in which *id* is the task identifier, *backend* is the data that will be used to create the task and *target* is the variable of data that has class identification. In the example, «XdataSet»is the string that identifies the task, XdataSet is the data frame, and «label»is the category.

```
task <- TaskClassif$new(id = "XdataSet", backend = XdataSet, target
    = "label")
```

The code below contains a list of the classifiers that were chosen. The easy access to documentation and the ability to replicate results are two advantages of using the *mlr3* framework.

```
classifier.list <- c("classif.AdaBoostM1",
"classif.C50",
"classif.catboost",
"classif.ctree",
"classif.cv_glmnet",
"classif.featureless",
"classif.gbm",
"classif.glmnet",
"classif.IBk",
"classif.JRip",
"classif.kknn",
"classif.lda",
"classif.liblinear",
"classif.lightgbm",
"classif.LMT",
"classif.naive_bayes",
"classif.nnet",
"classif.OneR",
"classif.PART",
"classif.randomForest",
"classif.ranger",
"classif.rfsrc",
"classif.rpart",
"classif.svm",
"classif.xgboost")
```

The snippet of code below illustrates how to create a *Learner* (*lrn*), for which the output are probabilities of each class of a multi-class problem.

```
# classifier.name is one of the classifiers in classifier.list

# classif is a function that returns a trained model

classif <- function(classifier.name, task)
{
learner <- lrn(classifier.name, predict_type = "prob")
subsampling <- rsmp("subsampling", repeats = 20, ratio = 0.8)
rr <- resample(task, learner, subsampling, store_models = TRUE)
return(rr)
}
```

The piece of code below uses *map* to apply *classif* to each element of *classifier.list*, given the data set defined in *task*.

```
# rr is the output variable that stores all the created models in a list.
rr <- map(classifier.list, classif, task)
```

References

1. de Lacerda, L.F.T. Analysis of the Quality of Accountability of Private Foundations in the Federal District to the Public Ministry of the Federal District and Territories. Bachelor Dissertation, Universidade de Brasília, Brasília, Brazil, 2017. Available online: https://bdm.unb.br/handle/10483/18432 (accessed on 1 August 2022).
2. Portulhak, H.; Vaz, P.V.C.; Delay, A.J.; Pacheco, V. The quality of third sector organizations' accountability: An analysis from its relationship with the behavior of individual donors. *Enfoque Reflexão Contábil* **2017**, *36*, 45–63. .
31273. [CrossRef]
3. Trussel, J.M.; Parsons, L.M. Financial reporting factors affecting donations to charitable organizations. *Adv. Account.* **2007**, *23*, 263–285. [CrossRef]

4. Rana, T.; Steccolini, I.; Bracci, E.; Mihret, D.G. Performance auditing in the public sector: A systematic literature review and future research avenues. *Financ. Account. Manag.* **2021**, *38*, 337–359. [CrossRef]
5. Otia, J.E.; Bracci, E. Digital transformation and the public sector auditing: The SAI's perspective. *Financ. Account. Manag.* **2022**, *38*, 252–280. [CrossRef]
6. Sun, T.; Sales, L.J. Predicting public procurement irregularity: An application of neural networks. *J. Emerg. Technol. Account.* **2018**, *15*, 141–154. [CrossRef]
7. Zhang, X. Construction and simulation of financial audit model based on convolutional neural network. *Comput. Intell. Neurosci.* **2021**, *2021*, 1–11. [CrossRef]
8. Mongwe, W.T.; Mbuvha, R.; Marwala, T. Bayesian inference of local government audit outcomes. *PLoS ONE* **2021**, *16*, e0261245. [CrossRef]
9. Khan, A.T.; Cao, X.; Li, S.; Katsikis, V.N.; Brajevic, I.; Stanimirovic, P.S. Fraud detection in publicly traded u.s firms using beetle antennae search: A machine learning approach. *Expert Syst. Appl.* **2022**, *191*, 116148. [CrossRef]
10. Jiang, Y.; Jones, S. Corporate distress prediction in China: A machine learning approach. *Account. Financ.* **2018**, *58*, 1063–1109. [CrossRef]
11. Abbasi, A.; Albrecht, C.; Vance, A.; Hansen, J. MetaFraud: A meta-learning framework for detecting financial fraud. *MIS Q.* **2012**, *36*, 1293–1327. [CrossRef]
12. Hamal, S.; Senvar, O. Comparing performances and effectiveness of machine learning classifiers in detecting financial accounting fraud for Turkish SMEs. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 769–782. [CrossRef]
13. Bertomeu, J.; Cheynel, E.; Floyd, E.; Pan, W. Using machine learning to detect misstatements. *Rev. Account. Stud.* **2020**, *26*, 468–519. [CrossRef]
14. Bao, Y.; Ke, B.; Li, B.; Yu, Y.J.; Zhang, J. Detecting accounting fraud in publicly traded U.S. firms using a machine learning approach. *J. Account. Res.* **2020**, *58*, 199–235. [CrossRef]
15. Zhang, X. Application of data mining and machine learning in management accounting information system. *J. Appl. Sci. Eng.* **2021**, *24*, 813–820. ._24(5).0018. [CrossRef]
16. Song, X.P.; Hu, Z.H.; Du, J.G.; Sheng, Z.H. Application of machine learning methods to risk assessment of financial statement fraud: Evidence from China. *J. Forecast.* **2014**, *33*, 611–626. [CrossRef]
17. Papík, M.; Papíková, L. Detecting accounting fraud in companies reporting under US GAAP through data mining. *Int. J. Account. Inf. Syst.* **2022**, *45*, 100559. [CrossRef]
18. Chen, Y.; Zhang, S. Accounting information disclosure and financial crisis beforehand warning based on the artificial neural network. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1–11. [CrossRef]
19. Li, Q. Parallel bookkeeping path of accounting in government accounting system based on deep neural network. *J. Electr. Comput. Eng.* **2022**, *2022*, 1–10. [CrossRef]
20. Liu, L. Evaluation method of financial accounting quality in colleges and universities based on dynamic neuron model. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–11. [CrossRef]
21. Cecchini, M.; Aytug, H.; Koehler, G.J.; Pathak, P. Detecting management fraud in public companies. *Manag. Sci.* **2010**, *56*, 1146–1160. [CrossRef]
22. Kuzey, C.; Uyar, A.; Delen, D. An investigation of the factors influencing cost system functionality using decision trees, support vector machines and logistic regression. *Int. J. Account. Inf. Manag.* **2019**, *27*, 27–55. [CrossRef]
23. de Laat, P.B. Algorithmic decision-making based on machine learning from big data: Can transparency restore accountability? *Philos. Technol.* **2017**, *31*, 525–541. [CrossRef] [PubMed]
24. Bakumenko, A.; Elragal, A. Detecting anomalies in financial data using machine learning algorithms. *Systems* **2022**, *10*, 130. [CrossRef]
25. Zou, J.; Fu, X.; Yang, J.; Gong, C. Measuring bank systemic risk in china: A network model analysis. *Systems* **2022**, *10*, 14. [CrossRef]
26. Nonaka, T.H. Estudo comparativo dos manuais de prestação de contas do governo federal. Bachelor Dissertation, Universidade de Brasília, Brasília, Brazil, 2013. Available online: http://bdm.unb.br/handle/10483/12574 (accessed on 1 August 2022).
27. Pereira, J.R.T.; Filho, J.B.C. Rejeições de prestação de contas de governos municipais: O que está acontecendo? *Contabilidade Gestão e Governança* **2012**, *15*, 33–43. Available online: https://www.revistacgg.org/index.php/contabil/article/view/393 (accessed on 1 August 2022).
28. Lima, M.B. Organizações não governamentais (ONGs): Um estudo sobre a transparência na elaboração da prestação de contas e dos relatórios financeiros emitidos nas organizações não governamentais do DF. Bachelor Dissertation, Universidade de Brasília, Brasília, DF, Brazil, 2011. [CrossRef]
29. e Barros, F.H.G.; Neto, M.S. Inserindo a dimensão de resultados nas prestações de contas. *Revista do Tribunal de Contas da União* **2010**, *119*, 65–70. Available online: https://revista.tcu.gov.br/ojs/index.php/RTCU/article/view/201/194 (accessed on 3 December 2022).
30. Tomaskova, H.; Kopecky, M. Specialization of business process model and notation applications in medicine—A review. *Data* **2020**, *5*, 99. [CrossRef]
31. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2021.

32. Moutinho, J.d.A.; Rabechini Junior, R. Adherence between project management and the management system of agreements and transfer contracts (SICONV). *Syst. Manag.* **2017**, *12*, 83–97. [CrossRef]
33. Borchers, H.W. *Pracma: Practical Numerical Math Functions*, R Package Version 2.3.8; 2022. Available online: https://cran.r-project.org/web/packages/pracma/index.html (accessed on 1 August 2022).
34. Abdi, H.; Williams, L.J. Principal component analysis. *WIREs Comput. Stat.* **2010**, *2*, 433–459. [CrossRef]
35. Kassambara, A.; Mundt, F. *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*, R Package Version 1.0.7; 2020. Available online: https://cran.r-project.org/web/packages/factoextra/readme/README.html (accessed on 1 August 2022).
36. Hartmann, K.; Krois, J. *E-Learning Project SOGA: Statistics and Geospatial Data Analysis*; Department of Earth Sciences, Freie Universitaet Berlin: Berlin, Germany, 2018. Available online: https://www.geo.fu-berlin.de/en/v/soga/index.html (accessed on 12 September 2022).
37. van der Maaten, L.; Hinton, G. Visualizing Data Using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
38. Krijthe, J.H. *Rtsne: T-Distributed Stochastic Neighbor Embedding Using Barnes-Hut Implementation*, R Package Version 0.16; 2015. Available online: https://cran.r-project.org/web/packages/Rtsne/index.html (accessed on 1 August 2022).
39. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *Appl. Stat.* **1979**, *28*, 100. . 2346830. [CrossRef]
40. Lang, M.; Binder, M.; Richter, J.; Schratz, P.; Pfisterer, F.; Coors, S.; Au, Q.; Casalicchio, G.; Kotthoff, L.; Bischl, B. mlr3: A modern object-oriented machine learning framework in R. *J. Open Source Softw.* **2019**, *4*, 1903. [CrossRef]
41. Sonabend, R.; Schratz, P.; Fischer, S. *mlr3extralearners: Extra Learners for mlr3*, R Package Version 0.5.48; 2022. Available online: https://github.com/mlr-org/mlr3extralearners (accessed on 1 August 2022).
42. Lang, M. *mlr3measures: Performance Measures for 'mlr3'*, R Package Version 0.5.0; 2022. Available online: https://cran.r-project.org/web/packages/mlr3measures/index.html (accessed on 1 August 2022).
43. Peng, W.; Ye, Z.S.; Chen, N. Bayesian deep-learning-based health prognostics toward prognostics uncertainty. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2283–2293. [CrossRef]
44. Zhang, L.; Xu, A.; An, L.; Li, M. Bayesian inference of system reliability for multicomponent stress-strength model under Marshall-Olkin Weibull distribution. *Systems* **2022**, *10*, 196. [CrossRef]
45. Bandyopadhyay, S.; Maulik, U. An evolutionary technique based on k-means algorithm for optimal clustering in R^N. *Inf. Sci.* **2002**, *146*, 221–237. [CrossRef]
46. Ikotun, A.M.; Almutari, M.S.; Ezugwu, A.E. K-means-based nature-inspired metaheuristic algorithms for automatic data clustering problems: Recent advances and future directions. *Appl. Sci.* **2021**, *11*, 11246. [CrossRef]
47. Ikotun, A.M.; Ezugwu, A.E. Boosting k-means clustering with symbiotic organisms search for automatic clustering problems. *PLoS ONE* **2022**, *17*, 1–33. [CrossRef]
48. Polikar, R. Ensemble based systems in decision making. *IEEE Circuits Syst. Mag.* **2006**, *6*, 21–45. . 2006.1688199. [CrossRef]
49. Pandya, R.; Pandya, J.; Dholakiya, K.P.; Amreli, I. C5.0 algorithm to improved decision tree with feature selection and reduced error pruning. *Int. J. Comput. Appl.* **2015**, *117*, 975–8887.
50. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. *J. Big Data* **2020**, *7*. . S40537-020-00369-8. [CrossRef]
51. Maloney, K.O.; Weller, D.E.; Russell, M.J.; Hothorn, T. Classifying the biological condition of small streams: An example using benthic macroinvertebrates. *J. N. Am. Benthol. Soc.* **2009**, *28*, 869–884. [CrossRef]
52. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **2010**, *33*, 1–22. [CrossRef] [PubMed]
53. Duin, R.P.W.; De, D.; And, R.; Tax, D.M.J. Featureless pattern classification. *Kybernetika* **1998**, *34*, 399–404. Available online: https://www.kybernetika.cz/content/1998/4/399/paper.pdf (accessed on 3 December 2022).
54. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurorobotics* **2013**, *7*, 21. . 2013.00021/BIBTEX. [CrossRef] [PubMed]
55. Shrivastav, L.K.; Jha, S.K. A gradient boosting machine learning approach in modeling the impact of temperature and humidity on the transmission rate of COVID-19 in India. *Appl. Intell.* **2021**, *51*, 2727–2739. [CrossRef] [PubMed]
56. Kalmegh, S.R. Effective classification of Indian News using Lazy classifier IB1And IBk from weka. *Int. J. Inf. Comput. Sci.* **2019**, *6*, 160–168.
57. Gupta, A.; Mohammad, A.; Syed, A.; Halgamuge, M.N. A comparative study of classification algorithms using data mining: crime and accidents in denver city the USA. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*. [CrossRef]
58. Tarun, I.M.; Gerardo, B.D.; Tanguilig III, B.T. Generating licensure examination performance models using PART and JRip classifiers: A data mining application in education. *Int. J. Comput. Commun. Eng.* **2014**, *3*, 202–207. . V3.320. [CrossRef]
59. Zhang, Z. Introduction to machine learning: K-nearest neighbors. *Ann. Transl. Med.* **2016**, *4*, 218. [CrossRef]
60. Calil, B.C.; Da Cunha, D.V.; Vieira, M.F.; De Oliveira Andrade, A.; Furtado, D.A.; Bellomo Junior, D.P.; Pereira, A.A. Identification of arthropathy and myopathy of the temporomandibular syndrome by biomechanical facial features. *Biomed. Eng. Online* **2020**, *19*. [CrossRef]

61. Bhardwaj, A.; Gupta, A.; Jain, P.; Rani, A.; Yadav, J. Classification of human emotions from EEG signals using SVM and LDA classifiers. In Proceedings of the 2nd International Conference on Signal Processing and Integrated Networks, SPIN 2015, Noida, India, 19–20 February 2015; pp. 180–185. [CrossRef]
62. Cavalheiro, G.L.; Almeida, M.F.S.; Pereira, A.A.; Andrade, A.O. Study of age-related changes in postural control during quiet standing through linear discriminant analysis. *Biomed. Eng. Online* **2009**, *8*, 35. [CrossRef] [PubMed]
63. Al-Zubaidi, A.; Rabee, F.; Al-Sulttani, A.H.; Al-Zubaidi, E.A. Classification of large-scale datasets of Landsat-8 satellite image based on LIBLINEAR library. *Al-Salam J. Eng. Technol.* **2022**, *1*, 9–17. [CrossRef]
64. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.
65. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 3149–3157. [CrossRef]
66. Lee, S.; Jun, C.H. Fast incremental learning of logistic model tree using least angle regression. *Expert Syst. Appl.* **2018**, *97*, 137–145. [CrossRef]
67. Park, Y. A comparison of neural net classifiers and linear tree classifiers: Their similarities and differences. *Pattern Recognit.* **1994**, *27*, 1493–1503. [CrossRef]
68. Behera, S.S.; Chaudhuri, S.B.; Chattopadhyay, S. A comparative study on neural net classifier optimizations. *Int. J. Adv. Eng. Technol.* **2012**, *179*, 179–187. Available online: https://www.ijaet.org/media/0006/20I10-IJAET0907113-A-COMPARATIVE-STUDY.pdf (accessed on 3 December 2022).
69. Behera, S.S.; Chattopadhyay, S. A comparative study of back propagation and simulated annealing algorithms for neural net classifier optimization. *Procedia Eng.* **2012**, *38*, 448–455. [CrossRef]
70. Jamjoom, M. The pertinent single-attribute-based classifier for small datasets classification. *Int. J. Electr. Comput. Eng. (IJECE)* **2020**, *10*, 3227–3234. [CrossRef]
71. Iyer, K.B.P.; Pavithra, K.; Nivetha, D.; Kumudhavarshini, K. Predictive analytics in diabetes using oner classification algorithm. *IJCA Proc. Int. Conf. Commun. Comput. Inf. Technol.* **2018**, 14–19. Available online: https://research.ijcaonline.org/icccmit2017/number1/icccmit201718.pdf (accessed on 3 December 2022).
72. Alam, F.; Pachauri, S. Comparative study of j48, Naive Bayes and One-R classification technique for credit card fraud detection using WEKA. *Adv. Comput. Sci. Technol.* **2017**, *10*, 1731–1743. Available online: https://www.ripublication.com/acst17/acstv10n6_19.pdf (accessed on 3 December 2022).
73. Frank, E.; Witten, I.H. Generating accurate rule sets without global optimization. In Proceedings of the Fifteenth International Conference on Machine Learning, Madison, WI, USA, 24–27 July 1998; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1998; pp. 144–151. [CrossRef]
74. Makalesi, A.; Kaya, Y.; Tekin, R. Comparison of discretization methods for classifier decision trees and decision rules on medical data sets. *Eur. J. Sci. Technol.* **2022**, 275–281. [CrossRef]
75. Nasa, C.; Cse Deptt, S.A.P. Evaluation of different classification techniques for WEB data. *Int. J. Comput. Appl.* **2012**, *52*, 975–8887. [CrossRef]
76. Porwik, P.; Doroz, R.; Orczyk, T. The k-NN classifier and self-adaptive Hotelling data reduction technique in handwritten signatures recognition. *Pattern Anal. Appl.* **2015**, *18*, 983–1001. [CrossRef]
77. Caie, P.D.; Dimitriou, N.; Arandjelović, O. Chapter 8 - Precision medicine in digital pathology via image analysis and machine learning. In *Artificial Intelligence and Deep Learning in Pathology*; Cohen, S., Ed.; Elsevier: Amsterdam, The Netherlands, 2021; pp. 149–173. [CrossRef]
78. Amar, D.; Izraeli, S.; Shamir, R. Utilizing somatic mutation data from numerous studies for cancer research: Proof of concept and applications. *Oncogene* **2017**, *36*, 3375–3383. [CrossRef] [PubMed]
79. Loh, W.Y. Fifty years of classification and regression trees. *Int. Stat. Rev.* **2014**, *82*, 329–348. [CrossRef]
80. Carmona, P.; Dwekat, A.; Mardawi, Z. No more black boxes! Explaining the predictions of a machine learning XGBoost classifier algorithm in business failure. *Res. Int. Bus. Financ.* **2022**, *61*, 101649. [CrossRef]
81. Bentéjac, C.; Csörgő, A.; Martínez-Mu noz, G. A comparative analysis of XGBoost. *Artif. Intell. Rev.* **2019**, *54*, 1937–1967.